XenServer™

Develop for XenServer®

cloud™ SOFTWARE GROUP

XenServer™ Copyright © 1997–2026. Citrix Systems, Inc. All Rights Reserved.

Contents

Develop for XenServer® 69
- Other development tools 69
- Develop drivers and supplemental packs 69

XenServer® Software Development Kit 70
- Getting Started 70
  - System Requirements and Preparation 70
  - Downloading 70
  - SDK contents 71
  - Supported languages 71
  - Command line interface (CLI) 71
- Getting Started with PowerShell 72
  - RPC protocol 73
  - Platform supported 73
  - Library 73
  - Dependencies 73
  - Installation 73
  - Getting help 76
  - Building and debugging the source code 77
  - Examples 77
  - Using the XenServer® PowerShell module 77
    - Getting Started 77
    - XenServer PowerShell module overview 78
    - Complete walk-throughs 81
  - Update your XenServer® hosts by using the XenServer PowerShell module 82
    - Overview 82
    - Configure updates for your pool 82

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Obtain updates for your pool 86
View available updates for your pool 88
Understand the guidance categories and update tasks 89
Apply updates to your pool 94
Copy or live migrate a VM across pools 97
Configure vGPU and GPU pass-through 101
PowerShell examples 107
Getting Started with C # 108
RPC protocol 108
Platform supported 108
Library 108
Dependencies 108
Installation 108
Building the source code 109
Examples 109
C# examples 109
Getting Started with Python 109
Alternative installation 109
RPC protocol 109
Platform supported 109
Library 110
Dependencies 110
Examples 110
Python examples 110
Getting Started with Java 110
RPC protocol 110
Platform supported 110
Library 111

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Dependencies 111
Examples 111
Java examples 111
Getting Started with C 111
RPC protocol 112
Platform supported 112
Library 112
Dependencies 112
Building the source code 112
Examples 112
C examples 112
Getting Started with Go 112
RPC protocol 113
Platform supported 113
Installation 113
Examples 114
Go examples 114
First steps with the API 114
Authentication: acquiring a session reference 114
Acquiring a list of templates to base a new VM installation on 115
Installing the VM based on a template 115
Taking the VM through a start/suspend/resume/stop cycle 116
Logging out 116
Install and start example: summary 117
Object Model Overview 117
Object identifiers 117
VM 118
Host 118

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDI 118
SR 119
Network 119
VBD 119
VIF 120
PIF 120
PBD 120
Working with VIFs and VBDs 122
Creating disks and attaching them to VMs 122
Creating a new blank disk image 122
Attaching the disk image to a VM 123
Hotplugging the VBD 124
Creating and attaching Network Devices to VMs 124
Host configuration for networking and storage 125
Host storage configuration: PBDs 125
Host networking configuration: PIFs 126
Importing and Exporting VMs 126
Where to look next 127
Using the API 127
Anatomy of a typical application 128
Choosing a low-level transport 128
Authentication and session handling 129
Finding references to useful objects 130
Invoking synchronous operations on objects 131
Using Tasks to manage asynchronous operations 131
Subscribing to and listening for events 132
Complete application examples 134
Simultaneously migrating VMs using live migration 134

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Cloning a VM using the xe CLI 138
Using HTTP to interact with XenServer® 141
Importing and exporting VMs as XVA packages 141
VM Export 141
VM Import 142
VDI Import 144
Xen® Virtual Appliance (XVA) VM Import Format 147
Getting XenServer Operational Metrics 152
Metrics available 152
The data store for operational metrics 152
Data Granularity 152
Downloading RRDs 153
Analysing RRDs 155
Example RRDs 156
XenServer® Management API extensions 166
VM console forwarding 166
Console access 166
Retrieving VNC consoles using the API 167
Disabling VNC forwarding for Linux VM 168
Adding xenstore entries to VMs 169
Security enhancements 170
Advanced settings for network interfaces 170
ethtool settings 171
Miscellaneous settings 172
Internationalization for SR names 172
Hiding objects from XenCenter 173
Other extensions 173
Pool 173

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Host 174
VM 175
SR 176
VDI 177
VBD 177
Network 177
VM_guest_metrics 177
Task 178
Management API reference 179
API overview 179
XenServer Management API Deprecation Policy 180
XenServer 9 Management API 180
Types 181
RPCs associated with fields 182
RPCs associated with classes 183
Wire Protocol for Remote API Calls 183
XML-RPC Protocol 183
JSON-RPC Protocol 186
Errors 192
Note on References vs UUIDs 197
Making RPC Calls 197
Example interactive session 199
VM Lifecycle 204
VM boot parameters 205
Classes 206
Relationships Between Classes 210
Types 214
Primitives 214

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Higher-order types 215
Enumeration types 215
Class: auth 242
Fields for class: auth 242
RPCs associated with class: auth 242
Class: blob 244
Fields for class: blob 244
RPCs associated with class: blob 245
Class: Bond 254
Fields for class: Bond 255
RPCs associated with class: Bond 256
Class: Certificate 267
Fields for class: Certificate 268
RPCs associated with class: Certificate 269
Class: Cluster 276
Fields for class: Cluster 277
RPCs associated with class: Cluster 279
Class: Clusterhost 297
Fields for class: Clusterhost 297
RPCs associated with class: Clusterhost 300
Class: console 311
Fields for class: console 311
RPCs associated with class: console 312
Class: crashdump 319
Fields for class: crashdump 320
RPCs associated with class: crashdump 320
Class: datasource 327
Fields for class: datasource 327

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: datasource 328
Class: DRtask 328
Fields for class: DRtask 328
RPCs associated with class: DRtask 329
Class: Drivervariant 333
Fields for class: Drivervariant 333
RPCs associated with class: Drivervariant 334
Class: event 341
Fields for class: event 341
RPCs associated with class: event 341
Class: Feature 346
Fields for class: Feature 346
RPCs associated with class: Feature 346
Class: GPUgroup 353
Fields for class: GPUgroup 353
RPCs associated with class: GPUgroup 354
Class: host 368
Fields for class: host 368
RPCs associated with class: host 377
Class: hostcpu 492
Fields for class: hostcpu 492
RPCs associated with class: hostcpu 493
Class: hostcrashdump 505
Fields for class: hostcrashdump 506
RPCs associated with class: hostcrashdump 506
Class: Hostdriver 514
Fields for class: Hostdriver 514
RPCs associated with class: Hostdriver 515

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: hostmetrics 525
Fields for class: hostmetrics 525
RPCs associated with class: hostmetrics 525
Class: hostpatch 533
Fields for class: hostpatch 533
RPCs associated with class: hostpatch 534
Class: LVHD 546
Fields for class: LVHD 546
RPCs associated with class: LVHD 546
Class: message 549
Fields for class: message 549
RPCs associated with class: message 550
Class: network 556
Fields for class: network 556
RPCs associated with class: network 558
Class: networksriov 579
Fields for class: networksriov 579
RPCs associated with class: networksriov 579
Class: Observer 586
Fields for class: Observer 587
RPCs associated with class: Observer 588
Class: PBD 600
Fields for class: PBD 600
RPCs associated with class: PBD 601
Class: PCI 611
Fields for class: PCI 611
RPCs associated with class: PCI 612
Class: PGPU 627

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: PGPU 627
RPCs associated with class: PGPU 628
Class: PIF 643
Fields for class: PIF 643
RPCs associated with class: PIF 647
Class: PIFmetrics 681
Fields for class: PIFmetrics 681
RPCs associated with class: PIFmetrics 682
Class: pool 694
Fields for class: pool 694
RPCs associated with class: pool 703
Class: poolpatch 814
Fields for class: poolpatch 815
RPCs associated with class: poolpatch 816
Class: poolupdate 831
Fields for class: poolupdate 831
RPCs associated with class: poolupdate 832
Class: proberesult 846
Fields for class: proberesult 846
RPCs associated with class: proberesult 847
Class: PUSB 847
Fields for class: PUSB 847
RPCs associated with class: PUSB 849
Class: PVScachestorage 861
Fields for class: PVScachestorage 862
RPCs associated with class: PVScachestorage 862
Class: PVSproxy 869
Fields for class: PVSproxy 869

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: PVSproxy 869
Class: PVSserver 875
Fields for class: PVSserver 876
RPCs associated with class: PVSserver 876
Class: PVSsite 882
Fields for class: PVSsite 883
RPCs associated with class: PVSsite 883
Class: Repository 893
Fields for class: Repository 893
RPCs associated with class: Repository 895
Class: role 908
Fields for class: role 909
RPCs associated with class: role 909
Class: SDNcontroller 917
Fields for class: SDNcontroller 917
RPCs associated with class: SDNcontroller 917
Class: secret 923
Fields for class: secret 923
RPCs associated with class: secret 923
Class: session 931
Fields for class: session 931
RPCs associated with class: session 933
Class: SM 949
Fields for class: SM 949
RPCs associated with class: SM 950
Class: SR 963
Fields for class: SR 964
RPCs associated with class: SR 966

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: srstat 1002
Fields for class: srstat 1002
RPCs associated with class: srstat 1003
Class: subject 1003
Fields for class: subject 1003
RPCs associated with class: subject 1003
Class: task 1011
Fields for class: task 1011
RPCs associated with class: task 1013
Class: tunnel 1031
Fields for class: tunnel 1031
RPCs associated with class: tunnel 1032
Class: USBgroup 1043
Fields for class: USBgroup 1043
RPCs associated with class: USBgroup 1044
Class: user 1054
Fields for class: user 1054
RPCs associated with class: user 1054
Class: VBD 1062
Fields for class: VBD 1062
RPCs associated with class: VBD 1064
Class: VBDmetrics 1090
Fields for class: VBDmetrics 1090
RPCs associated with class: VBDmetrics 1091
Class: VDI 1098
Fields for class: VDI 1098
RPCs associated with class: VDI 1102
Class: vdinbdserverinfo 1147

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: vdinbdserverinfo 1147

RPCs associated with class: vdinbdserverinfo 1148

Class: VGPU 1148

Fields for class: VGPU 1148

RPCs associated with class: VGPU 1149

Class: VGPUtype 1162

Fields for class: VGPUtype 1162

RPCs associated with class: VGPUtype 1164

Class: VIF 1175

Fields for class: VIF 1175

RPCs associated with class: VIF 1178

Class: VIFmetrics 1209

Fields for class: VIFmetrics 1209

RPCs associated with class: VIFmetrics 1210

Class: VLAN 1217

Fields for class: VLAN 1217

RPCs associated with class: VLAN 1218

Class: VM 1226

Fields for class: VM 1226

RPCs associated with class: VM 1237

Class: VMappliance 1375

Fields for class: VMappliance 1375

RPCs associated with class: VMappliance 1376

Class: VMgroup 1389

Fields for class: VMgroup 1389

RPCs associated with class: VMgroup 1390

Class: VMguestmetrics 1397

Fields for class: VMguestmetrics 1397

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VMguestmetrics 1400
Class: VMmetrics 1413
Fields for class: VMmetrics 1413
RPCs associated with class: VMmetrics 1415
Class: VMPP 1430
Fields for class: VMPP 1430
RPCs associated with class: VMPP 1432
Class: VMSS 1464
Fields for class: VMSS 1464
RPCs associated with class: VMSS 1465
Class: VTPM 1480
Fields for class: VTPM 1481
RPCs associated with class: VTPM 1482
Class: VUSB 1490
Fields for class: VUSB 1490
RPCs associated with class: VUSB 1491
Error Codes 1500
ACTIVATIONWHILENOTFREE 1500
ADDRESSVIOLATESLOCKINGCONSTRAINT 1500
APPLYGUIDANCEFAILED 1501
APPLYLIVEPATCHFAILED 1501
APPLYUPDATESFAILED 1501
APPLYUPDATESINPROGRESS 1501
AUTHALREADYENABLED 1501
AUTHDISABLEFAILED 1502
AUTHDISABLEFAILEDPERMISSIONDENIED 1502
AUTHDISABLEFAILEDWRONGCREDENTIALS 1502
AUTHENABLEFAILED 1502

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHENABLEFAILEDDOMAINLOOKUPFAILED 1502
AUTHENABLEFAILEDINVALIDACCOUNT 1503
AUTHENABLEFAILEDINVALIDOU 1503
AUTHENABLEFAILEDINVALIDTRUSTEDCERTS 1503
AUTHENABLEFAILEDNOTRUSTEDCERTS 1503
AUTHENABLEFAILEDPERMISSIONDENIED 1503
AUTHENABLEFAILEDUNAVAILABLE 1504
AUTHENABLEFAILEDWRONGCREDENTIALS 1504
AUTHINVALIDTRUSTEDCERTS 1504
AUTHISDISABLED 1504
AUTHNOTRUSTEDCERTS 1504
AUTHSERVICEERROR 1505
AUTHSETUPTLSCONNECTION 1505
AUTHSETLDAPSFAILED 1505
AUTHUNKNOWNTYPE 1505
BACKUPSCRIPTFAILED 1505
BALLOONINGTIMEOUTBEFOREMIGRATION 1506
BLOCKEDREPOURL 1506
BOOTLOADERFAILED 1506
BOOTDEVICEPASSTHROUGHDISALLOWED 1506
BRIDGENAMEEXISTS 1506
BRIDGENOTAVAILABLE 1507
BUNDLEREPOSITORYALREADYEXISTS 1507
BUNDLEREPONOTENABLED 1507
BUNDLESYNCFAILED 1507
BUNDLEUNPACKFAILED 1507
CANNOTADDTUNNELTOBONDSLAVE 1508
CANNOTADDTUNNELTOSRIOVLOGICAL 1508

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTADDTUNNELTOVLANONSRIOVLOGICAL 1508
CANNOTADDVLANTOBONDSLAVE 1508
CANNOTCHANGEPIFPROPERTIES 1508
CANNOTCONTACTHOST 1509
CANNOTCREATESTATEFILE 1509
CANNOTDESTROYDISASTERRECOVERYTASK 1509
CANNOTDESTROYSYSTEMNETWORK 1509
CANNOTENABLEREDOLOG 1509
CANNOTEVACUATEHOST 1510
CANNOTFETCHPATCH 1510
CANNOTFINDOEMBACKUPPARTITION 1510
CANNOTFINDPATCH 1510
CANNOTFINDSTATEPARTITION 1510
CANNOTFINDUPDATE 1511
CANNOTFORGETSRIOVLOGICAL 1511
CANNOTPLUGBONDSLAVE 1511
CANNOTPLUGVIF 1511
CANNOTRESETCONTROLDOMAIN 1511
CANNOTRESTARTDEVICEMODEL 1512
CANNOTPERIODICSYNCUPDATES 1512
CANNOTSYNCUPDATES 1512
CERTIFICATEALREADYEXISTS 1512
CERTIFICATECORRUPT 1512
CERTIFICATEDOESNOTEXIST 1513
CERTIFICATELACKSPURPOSE 1513
CERTIFICATELIBRARYCORRUPT 1513
CERTIFICATENAMEINVALID 1513
CHANGEPASSWORDREJECTED 1513

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CLUSTEREDSRDEGRADED 1514
CLUSTERINGDISABLED 1514
CLUSTERINGENABLED 1514
CLUSTERALREADYEXISTS 1514
CLUSTERCREATEINPROGRESS 1514
CLUSTERDOESNOTHAVEONENODE 1515
CLUSTERFORCEDESTROYFAILED 1515
CLUSTERHOSTISLAST 1515
CLUSTERHOSTNOTJOINED 1515
CLUSTERSTACKINUSE 1515
CONFIGUREREPOSITORIESINPROGRESS 1516
COULDNOTFINDNETWORKINTERFACEWITHSPECIFIEDDEVICENAMEANDMACADDRESS 1516
COULDNOTIMPORTDATABASE 1516
COULDNOTUPDATEIGMPSNOOPINGEVERYWHERE 1516
CPUFEATUREMASKINGNOTSUPPORTED 1516
CRLALREADYEXISTS 1517
CRLCORRUPT 1517
CRLDOESNOTEXIST 1517
CRLNAMEINVALID 1517
DBUNIQUENESSCONSTRAINTVIOLATION 1517
DEFAULTSRNOTFOUND 1518
DEVICEALREADYATTACHED 1518
DEVICEALREADYDETACHED 1518
DEVICEALREADYEXISTS 1518
DEVICEATTACHTIMEOUT 1518
DEVICEDETACHREJECTED 1519
DEVICEDETACHTIMEOUT 1519
DEVICENOTATTACHED 1519

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DISABLESSHFAILED 1519

DISABLESSHPARTIALLYFAILED 1519

DISKVBDMUSTBEREADWRITEFORHVM 1520

DOMAINBUILDERERROR 1520

DOMAINEXISTS 1520

DUPLICATEMACSEED 1520

DUPLICATEPIFDEVICENAME 1520

DUPLICATEVM 1521

ENABLESSHFAILED 1521

ENABLESSHPARTIALLYFAILED 1521

EVENTSLOST 1521

EVENTFROMTOKENPARSEFAILURE 1521

EVENTSUBSCRIPTIONPARSEFAILURE 1522

FAILEDTOSTARTEMULATOR 1522

FEATUREREQUIRESHVM 1522

FEATURERESTRICTED 1522

FIELDTYPEERROR 1522

GETHOSTUPDATESFAILED 1523

GETUPDATESFAILED 1523

GPUGROUPCONTAINSNOPGPUS 1523

GPUGROUPCONTAINSPGPU 1523

GPUGROUPCONTAINSVGPU 1523

HANDLEINVALID 1524

HAABORTNEWMASTER 1524

HACANNOTCHANGEBONDSTATUSOFMGMTIFACE 1524

HACONSTRAINTVIOLATIONNETWORKNOTSHARED 1524

HACONSTRAINTVIOLATIONSRNOTSHARED 1525

HADISABLEINPROGRESS 1525

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HAENABLEINPROGRESS 1525

HAFAILEDTOFORMLIVESET 1525

HAHEARTBEATDAEMONSTARTUPFAILED 1525

HAHOSTCANNOTACCESSSTATEFILE 1526

HAHOSTCANNOTSEEPEERS 1526

HAHOSTISARMED 1526

HAISENABLED 1526

HALOSTSTATEFILE 1526

HANOTENABLED 1527

HANOTINSTALLED 1527

HANOPLAN 1527

HAOPERATIONWOULDBREAKFAILOVERPLAN 1527

HAPOOLISENABLEDBUTHOSTISDISABLED 1527

HASHOULDBEFENCED 1528

HATOOFEWHOSTS 1528

HOSTSNOTCOMPATIBLE 1528

HOSTSNOTHOMOGENEOUS 1528

HOSTBROKEN 1528

HOSTCANNOTATTACHNETWORK 1529

HOSTCANNOTDESTROYSELF 1529

HOSTCANNOTREADMETRICS 1529

HOSTCDDRIVEEMPTY 1529

HOSTDISABLED 1529

HOSTDISABLEDINDEFINITELY 1530

HOSTDISABLEDUNTILREBOOT 1530

HOSTDRIVERNOHARDWARE 1530

HOSTEVACUATEINPROGRESS 1530

HOSTEVACUATIONISREQUIRED 1530

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTHASNOMANAGEMENTIP 1531
HOSTHASRESIDENTVMS 1531
HOSTINEMERGENCYMODE 1531
HOSTINUSE 1531
HOSTISLIVE 1531
HOSTISSLAVE 1532
HOSTITSOWNSLAVE 1532
HOSTMASTERCANNOTTALKBACK 1532
HOSTNAMEINVALID 1532
HOSTNOTDISABLED 1532
HOSTNOTENOUGHFREEMEMORY 1533
HOSTNOTENOUGHPCPUS 1533
HOSTNOTLIVE 1533
HOSTOFFLINE 1533
HOSTPENDINGMANDATORYGUIDANCENOTEMPTY 1533
HOSTPOWERONMODEDISABLED 1534
HOSTSTILLBOOTING 1534
HOSTUNKNOWNTOMASTER 1534
HOSTXAPIVERSIONHIGHERTHANCOORDINATOR 1534
ILLEGALVBDDEVICE 1534
IMPORTERROR 1535
IMPORTERRORATTACHEDDISKSNOTFOUND 1535
IMPORTERRORCANNOTHANDLECHUNKED 1535
IMPORTERRORFAILEDTOFINDOBJECT 1535
IMPORTERRORPREMATUREEOF 1535
IMPORTERRORSOMECHECKSUMSFAILED 1536
IMPORTERRORUNEXPECTEDFILE 1536
IMPORTINCOMPATIBLEVERSION 1536

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INCOMPATIBLECLUSTERSTACKACTIVE 1536

INCOMPATIBLEPIFPROPERTIES 1536

INCOMPATIBLESTATEFILESR 1537

INTERFACEHASNOIP 1537

INTERNALERROR 1537

INVALIDBASEURL 1537

INVALIDCIDRADDRESSSPECIFIED 1537

INVALIDCLUSTERSTACK 1538

INVALIDDEVICE 1538

INVALIDEDITION 1538

INVALIDFEATURESTRING 1538

INVALIDGPGKEYPATH 1538

INVALIDIPADDRESSSPECIFIED 1539

INVALIDNTPCONFIG 1539

INVALIDPATCH 1539

INVALIDPATCHWITHLOG 1539

INVALIDREPOMDXML 1539

INVALIDREPOSITORYDOMAINALLOWLIST 1540

INVALIDREPOSITORYPROXYCREDENTIAL 1540

INVALIDREPOSITORYPROXYURL 1540

INVALIDUPDATE 1540

INVALIDUPDATEINFOXML 1540

INVALIDUPDATESYNCDAY 1541

INVALIDVALUE 1541

ISTUNNELACCESSPIF 1541

JOININGHOSTCANNOTBEMASTEROFOTHERHOSTS 1541

JOININGHOSTCANNOTCONTAINSHAREDSRS 1541

JOININGHOSTCANNOTHAVERUNNINGORSUSPENDEDVMS 1542

© 1997–2026 Citrix Systems, Inc. All rights reserved.

JOININGHOSTCANNOTHAVERUNNINGVMS 1542
JOININGHOSTCANNOTHAVEVMSWITHCURRENTOPERATIONS 1542
JOININGHOSTCONNECTIONFAILED 1542
JOININGHOSTSERVICEFAILED 1542
LICENCERESTRICTION 1543
LICENSECANNOTDOWNGRADEWHILEINPOOL 1543
LICENSECHECKOUTERROR 1543
LICENSEDOESNOTSUPPORTPOOLING 1543
LICENSEDOESNOTSUPPORTXHA 1543
LICENSEEXPIRED 1544
LICENSEFILEDEPRECATED 1544
LICENSEHOSTPOOLMISMATCH 1544
LICENSEPROCESSINGERROR 1544
LOCATIONNOTUNIQUE 1544
MACDOESNOTEXIST 1545
MACINVALID 1545
MACSTILLEXISTS 1545
MAPDUPLICATEKEY 1545
MEMORYCONSTRAINTVIOLATION 1545
MEMORYCONSTRAINTVIOLATIONMAXPIN 1546
MEMORYCONSTRAINTVIOLATIONORDER 1546
MESSAGEDEPRECATED 1546
MESSAGEMETHODUNKNOWN 1546
MESSAGEPARAMETERCOUNTMISMATCH 1546
MESSAGEREMOVED 1547
MIRRORFAILED 1547
MISSINGCONNECTIONDETAILS 1547
MULTIPLEUPDATEREPOSITORIESENABLED 1547

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKALREADYCONNECTED 1547
NETWORKCONTAINSPIF 1548
NETWORKCONTAINSVIF 1548
NETWORKHASINCOMPATIBLESRIOVPIFS 1548
NETWORKHASINCOMPATIBLEVLANONSRIOVPIFS 1548
NETWORKINCOMPATIBLEPURPOSES 1548
NETWORKINCOMPATIBLEWITHBOND 1549
NETWORKINCOMPATIBLEWITHSRIOV 1549
NETWORKINCOMPATIBLEWITHTUNNEL 1549
NETWORKINCOMPATIBLEWITHVLANONBRIDGE 1549
NETWORKINCOMPATIBLEWITHVLANONSRIOV 1549
NETWORKSRIOVALREADYENABLED 1550
NETWORKSRIOVDISABLEFAILED 1550
NETWORKSRIOVENABLEFAILED 1550
NETWORKSRIOVINSUFFICIENTCAPACITY 1550
NETWORKUNMANAGED 1550
NOTALLOWEDONOEMEDITION 1551
NOTALLOWEDWHENNTPISENABLED 1551
NOTIMPLEMENTED 1551
NOTINEMERGENCYMODE 1551
NOTSUPPORTEDDURINGUPGRADE 1551
NOTSYSTEMDOMAIN 1552
NOTTRUSTEDCERTIFICATE 1552
NOCLUSTERHOSTSREACHABLE 1552
NOCOMPATIBLECLUSTERHOST 1552
NOHOSTSAVAILABLE 1552
NOMOREREDOLOGSALLOWED 1553
NOREPOSITORIESCONFIGURED 1553

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOREPOSITORYENABLED 1553

NVIDIASRIOVMISCONFIGURED 1553

NVIDIATOOLSERROR 1553

OBJECTNOLONGEREXISTS 1554

ONLYALLOWEDONOEMEDITION 1554

OPENVSWITCHNOTACTIVE 1554

OPERATIONBLOCKED 1554

OPERATIONNOTALLOWED 1554

OPERATIONPARTIALLYFAILED 1555

OTHEROPERATIONINPROGRESS 1555

OUTOFSPACE 1555

PATCHALREADYAPPLIED 1555

PATCHALREADYEXISTS 1555

PATCHAPPLYFAILED 1556

PATCHAPPLYFAILEDBACKUPFILESEXIST 1556

PATCHISAPPLIED 1556

PATCHPRECHECKFAILEDISOMOUNTED 1556

PATCHPRECHECKFAILEDOUTOFSPACE 1556

PATCHPRECHECKFAILEDPREREQUISITEMISSING 1557

PATCHPRECHECKFAILEDUNKNOWNERROR 1557

PATCHPRECHECKFAILEDVMRUNNING 1557

PATCHPRECHECKFAILEDWRONGSERVERBUILD 1557

PATCHPRECHECKFAILEDWRONGSERVERVERSION 1557

PBDEXISTS 1558

PERMISSIONDENIED 1558

PGPUINSUFFICIENTCAPACITYFORVGPU 1558

PGPUINUSEBYVM 1558

PGPUNOTCOMPATIBLEWITHGPUGROUP 1558

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFALLOWSUNPLUG 1559

PIFALREADYBONDED 1559

PIFBONDMORETHANONEIP 1559

PIFBONDNEEDSMOREMEMBERS 1559

PIFCANNOTBONDCROSSHOST 1559

PIFCONFIGURATIONERROR 1560

PIFDEVICENOTFOUND 1560

PIFDOESNOTALLOWUNPLUG 1560

PIFHASFCOESRINUSE 1560

PIFHASNONETWORKCONFIGURATION 1560

PIFHASNOV6NETWORKCONFIGURATION 1561

PIFINCOMPATIBLEPRIMARYADDRESSTYPE 1561

PIFISMANAGEMENTINTERFACE 1561

PIFISNOTPHYSICAL 1561

PIFISNOTSRIOVCAPABLE 1561

PIFISPHYSICAL 1562

PIFISSRIOVLOGICAL 1562

PIFISVLAN 1562

PIFNOTATTACHEDTOHOST 1562

PIFNOTPRESENT 1562

PIFSRIOVSTILLEXISTS 1563

PIFTUNNELSTILLEXISTS 1563

PIFUNMANAGED 1563

PIFVLANEXISTS 1563

PIFVLANSTILLEXISTS 1563

POOLAUTHALREADYENABLED 1564

POOLAUTHDISABLEFAILED 1564

POOLAUTHDISABLEFAILEDINVALIDACCOUNT 1564

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLAUTHDISABLEFAILEDPERMISSIONDENIED 1564

POOLAUTHDISABLEFAILEDWRONGCREDENTIALS 1564

POOLAUTHENABLEFAILED 1565

POOLAUTHENABLEFAILEDDOMAINLOOKUPFAILED 1565

POOLAUTHENABLEFAILEDDUPLICATEHOSTNAME 1565

POOLAUTHENABLEFAILEDINVALIDACCOUNT 1565

POOLAUTHENABLEFAILEDINVALIDOU 1565

POOLAUTHENABLEFAILEDINVALIDTRUSTEDCERTS 1566

POOLAUTHENABLEFAILEDNOSUPPORTENCRYPTTYPE 1566

POOLAUTHENABLEFAILEDNOTRUSTEDCERTS 1566

POOLAUTHENABLEFAILEDPERMISSIONDENIED 1566

POOLAUTHENABLEFAILEDSETUPTLSCONNECTION 1566

POOLAUTHENABLEFAILEDUNAVAILABLE 1567

POOLAUTHENABLEFAILEDWRONGCREDENTIALS 1567

POOLJOININGEXTERNALAUTHMISMATCH 1567

POOLJOININGHOSTCACERTIFICATESCONFLICT 1567

POOLJOININGHOSTHASBONDS 1567

POOLJOININGHOSTHASNETWORKSRIOVS 1568

POOLJOININGHOSTHASNONMANAGEMENTVLANS 1568

POOLJOININGHOSTHASTUNNELS 1568

POOLJOININGHOSTMANAGEMENTVLANDOESNOTMATCH 1568

POOLJOININGHOSTMUSTHAVEONLYONEIPONCLUSTERINGNETWORK 1568

POOLJOININGHOSTMUSTHAVEPHYSICALMANAGEMENTNIC 1569

POOLJOININGHOSTMUSTHAVESAMEAPIVERSION 1569

POOLJOININGHOSTMUSTHAVESAMEDBSCHEMA 1569

POOLJOININGHOSTMUSTHAVESAMEPRODUCTVERSION 1569

POOLJOININGHOSTMUSTONLYHAVEPHYSICALPIFS 1569

POOLJOININGHOSTTLSVERIFICATIONMISMATCH 1570

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGHOSTTRUSTEDCERTIFICATESCONFLICT 1570

POOLJOININGMASTERCERTIFICATENOTINPOOLBUNDLE 1570

POOLJOININGPOOLBUNDLEEMPTYAFTERIMPORT 1570

POOLJOININGPOOLCANNOTENABLECLUSTERINGONVLANNETWORK 1570

POOLJOININGSMFEATURESINCOMPATIBLE 1571

PROVISIONFAILEDOUTOFSPACE 1571

PROVISIONONLYALLOWEDONTEMPLATE 1571

PUSBVDICONFLICT 1571

PVSCACHESTORAGEALREADYPRESENT 1571

PVSCACHESTORAGEISINUSE 1572

PVSPROXYALREADYPRESENT 1572

PVSPROXYPRESENTONHIGHERVIFDEVICE 1572

PVSSERVERADDRESSINUSE 1572

PVSSITECONTAINSRUNNINGPROXIES 1572

PVSSITECONTAINSSERVERS 1573

PVSVIFMUSTBEFIRSTDEVICE 1573

RBACPERMISSIONDENIED 1573

REDOLOGISENABLED 1573

REPOSITORYALREADYEXISTS 1573

REPOSITORYCLEANUPFAILED 1574

REPOSITORYISINUSE 1574

REPOSYNCFAILED 1574

REPOSHOULDBESINGLEONEENABLED 1574

REQUIREDPIFISUNPLUGGED 1574

RESTOREINCOMPATIBLEVERSION 1575

RESTORESCRIPTFAILED 1575

RESTORETARGETMGMTIFNOTINBACKUP 1575

RESTORETARGETMISSINGDEVICE 1575

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ROLEALREADYEXISTS 1575

ROLENOTFOUND 1576

SERVERCERTIFICATECHAININVALID 1576

SERVERCERTIFICATEEXPIRED 1576

SERVERCERTIFICATEINVALID 1576

SERVERCERTIFICATEKEYALGORITHMNOTSUPPORTED 1576

SERVERCERTIFICATEKEYINVALID 1577

SERVERCERTIFICATEKEYMISMATCH 1577

SERVERCERTIFICATEKEYRSALENGTHNOTSUPPORTED 1577

SERVERCERTIFICATEKEYRSAMULTINOTSUPPORTED 1577

SERVERCERTIFICATENOTVALIDYET 1577

SERVERCERTIFICATESIGNATURENOTSUPPORTED 1578

SESSIONAUTHENTICATIONFAILED 1578

SESSIONAUTHORIZATIONFAILED 1578

SESSIONINVALID 1578

SESSIONNOTREGISTERED 1578

SETCONSOLETIMEOUTPARTIALLYFAILED 1579

SETSSHAUTOMODEPARTIALLYFAILED 1579

SETSSHTIMEOUTPARTIALLYFAILED 1579

SLAVEREQUIRESMANAGEMENTINTERFACE 1579

SMPLUGINCOMMUNICATIONFAILURE 1579

SRATTACHFAILED 1580

SRBACKENDFAILURE 1580

SRDEVICEINUSE 1580

SRDOESNOTSUPPORTMIGRATION 1580

SRFULL 1580

SRHASMULTIPLEPBDS 1581

SRHASNOPBDS 1581

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SRHASPBD 1581

SRINDESTRUCTIBLE 1581

SRISCACHESR 1581

SRNOTATTACHED 1582

SRNOTEMPTY 1582

SRNOTSHARABLE 1582

SROPERATIONNOTSUPPORTED 1582

SRREQUIRESUPGRADE 1582

SRSOURCESPACEINSUFFICIENT 1583

SRSUSPENDSPACEINSUFFICIENT 1583

SRUNHEALTHY 1583

SRUNKNOWNDRIVER 1583

SRUUIDEXISTS 1583

SRVDILOCKINGFAILED 1584

SSLVERIFYERROR 1584

SUBJECTALREADYEXISTS 1584

SUBJECTCANNOTBERESOLVED 1584

SUSPENDIMAGENOTACCESSIBLE 1584

SYNCBUNDLEINPROGRESS 1585

SYNCUPDATESINPROGRESS 1585

SYSPREP 1585

SYSTEMSTATUSMUSTUSETARONOEM 1585

SYSTEMSTATUSRETRIEVALFAILED 1585

TASKCANCELLED 1586

TELEMETRYNEXTCOLLECTIONTOOLATE 1586

TLSCONNECTIONFAILED 1586

TLSVERIFICATIONNOTENABLEDINPOOL 1586

TOOBUSY 1586

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TOOMANYGROUPS 1587

TOOMANYPENDINGTASKS 1587

TOOMANYSTORAGEMIGRATES 1587

TOOMANYVUSBS 1587

TRANSPORTPIFNOTCONFIGURED 1587

TRUSTEDCERTIFICATEALREADYEXISTS 1588

TRUSTEDCERTIFICATEEXPIRED 1588

TRUSTEDCERTIFICATEINVALID 1588

TRUSTEDCERTIFICATENOTVALIDYET 1588

UNIMPLEMENTEDINSMBACKEND 1588

UNKNOWNBOOTLOADER 1589

UPDATEINFOHASHMISMATCH 1589

UPDATEALREADYAPPLIED 1589

UPDATEALREADYAPPLIEDINPOOL 1589

UPDATEALREADYEXISTS 1589

UPDATEAPPLYFAILED 1590

UPDATEISAPPLIED 1590

UPDATEPOOLAPPLYFAILED 1590

UPDATEPRECHECKFAILEDCONFLICTPRESENT 1590

UPDATEPRECHECKFAILEDGPGKEYNOTIMPORTED 1590

UPDATEPRECHECKFAILEDOUTOFSPACE 1591

UPDATEPRECHECKFAILEDPREREQUISITEMISSING 1591

UPDATEPRECHECKFAILEDUNKNOWNERROR 1591

UPDATEPRECHECKFAILEDWRONGSERVERVERSION 1591

UPDATESYNCINGREMOTEPOOLCOORDINATORCONNECTIONFAILED 1591

UPDATESYNCINGREMOTEPOOLCOORDINATORSERVICEFAILED 1592

USBALREADYATTACHED 1592

USBGROUPCONFLICT 1592

© 1997–2026 Citrix Systems, Inc. All rights reserved.

USBGROUPCONTAINSNOPUSBS 1592

USBGROUPCONTAINSPUSB 1592

USBGROUPCONTAINSVUSB 1593

USERISNOTLOCALSUPERUSER 1593

UUIDINVALID 1593

V6DFAILURE 1593

VALUENOTSUPPORTED 1593

VBDCDSMUSTBEREADONLY 1594

VBDISEMPTY 1594

VBDNOTEMPTY 1594

VBDNOTREMOVABLEMEDIA 1594

VBDNOTUNPLUGGABLE 1594

VBDTRAYLOCKED 1595

VCPUMAXNOTCORESPERSOCKETMULTIPLE 1595

VDICBTENABLED 1595

VDICONTAINSMETADATAOFTHISPOOL 1595

VDICOPYFAILED 1595

VDIHASRRDS 1596

VDIINCOMPATIBLETYPE 1596

VDIINUSE 1596

VDIISAPHYSICALDEVICE 1596

VDIISENCRYPTED 1596

VDIISNOTISO 1597

VDILOCATIONMISSING 1597

VDIMISSING 1597

VDINEEDSVMFORMIGRATE 1597

VDINOTAVAILABLE 1597

VDINOTINMAP 1598

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDINOTMANAGED 1598

VDINOTSPARSE 1598

VDINOCBTMETADATA 1598

VDIONBOOTMODEINCOMPATIBLEWITHOPERATION 1598

VDIREADONLY 1599

VDITOOLARGE 1599

VDITOOSMALL 1599

VGPUDESTINATIONINCOMPATIBLE 1599

VGPUGUESTDRIVERLIMIT 1599

VGPUSUSPENSIONNOTSUPPORTED 1600

VGPUTYPENOTCOMPATIBLE 1600

VGPUTYPENOTCOMPATIBLEWITHRUNNINGTYPE 1600

VGPUTYPENOTENABLED 1600

VGPUTYPENOTSUPPORTED 1600

VGPUTYPENOLONGERSUPPORTED 1601

VIFINUSE 1601

VIFNOTINMAP 1601

VLANINUSE 1601

VLANTAGINVALID 1601

VMPPARCHIVEMOREFREQUENTTHANBACKUP 1602

VMPPHASVM 1602

VMSSHASVM 1602

VMSFAILEDTOCOOPERATE 1602

VMASSIGNEDTOPROTECTIONPOLICY 1602

VMASSIGNEDTOSNAPSHOTSCHEDULE 1603

VMATTACHEDTOMORETHANONEVDIWITHTIMEOFFSETMARKEDASRESETONBOOT 1603

VMBADPOWERSTATE 1603

VMBIOSSTRINGSALREADYSET 1603

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMCALLPLUGINRATELIMIT 1603

VMCANNOTDELETEDEFAULTTEMPLATE 1604

VMCHECKPOINTRESUMEFAILED 1604

VMCHECKPOINTSUSPENDFAILED 1604

VMCRASHED 1604

VMDUPLICATEVBDDEVICE 1604

VMFAILEDSHUTDOWNACKNOWLEDGMENT 1605

VMFAILEDSUSPENDACKNOWLEDGMENT 1605

VMHALTED 1605

VMHASCHECKPOINT 1605

VMHASNOSUSPENDVDI 1605

VMHASPCIATTACHED 1606

VMHASSRIOVVIF 1606

VMHASTOOMANYSNAPSHOTS 1606

VMHASVGPU 1606

VMHASVUSBS 1606

VMHOSTINCOMPATIBLEVERSION 1607

VMHOSTINCOMPATIBLEVERSIONMIGRATE 1607

VMHOSTINCOMPATIBLEVIRTUALHARDWAREPLATFORMVERSION 1607

VMHVMREQUIRED 1607

VMINCOMPATIBLEWITHTHISHOST 1607

VMISIMMOBILE 1608

VMISPARTOFANAPPLIANCE 1608

VMISPROTECTED 1608

VMISTEMPLATE 1608

VMISUSINGNESTEDVIRT 1608

VMLACKSFEATURE 1609

VMMEMORYSIZETOOLOW 1609

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMMIGRATECONTACTREMOTESERVICEFAILED 1609
VMMIGRATEFAILED 1609
VMMISSINGPVDRIVERS 1609
VMNONSUSPENDABLE 1610
VMNOTRESIDENTHERE 1610
VMNOCRASHDUMPSR 1610
VMNOEMPTYCDVBD 1610
VMNOSUSPENDSR 1610
VMNOVCPUS 1611
VMOLDPVDRIVERS 1611
VMPCIBUSFULL 1611
VMPVDRIVERSINUSE 1611
VMREBOOTED 1611
VMREQUIRESGPU 1612
VMREQUIRESIOMMU 1612
VMREQUIRESNETWORK 1612
VMREQUIRESSR 1612
VMREQUIRESVDI 1612
VMREQUIRESVGPU 1613
VMREQUIRESVUSB 1613
VMREVERTFAILED 1613
VMSHUTDOWNTIMEOUT 1613
VMSNAPSHOTWITHQUIESCEFAILED 1613
VMSNAPSHOTWITHQUIESCENOTSUPPORTED 1614
VMSNAPSHOTWITHQUIESCEPLUGINDEOSNOTRESPOND 1614
VMSNAPSHOTWITHQUIESCETIMEOUT 1614
VMSUSPENDTIMEOUT 1614
VMTOOMANYVCPUS 1614

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMTOIMPORTISNOTNEWERVERSION 1615

VMUNSAFEBOOT 1615

VTPMMAXAMOUNTREACHED 1615

WLBAUTHENTICATIONFAILED 1615

WLBCONNECTIONREFUSED 1615

WLBCONNECTIONRESET 1616

WLBDISABLED 1616

WLBINTERNALERROR 1616

WLBMALFORMEDREQUEST 1616

WLBMALFORMEDRESPONSE 1616

WLBNOTINITIALIZED 1617

WLBTIMEOUT 1617

WLBUNKNOWNHOST 1617

WLBURLINVALID 1617

WLBXENSERVERAUTHENTICATIONFAILED 1617

WLBXENSERVERCONNECTIONREFUSED 1618

WLBXENSERVERMALFORMEDRESPONSE 1618

WLBXENSERVERTIMEOUT 1618

WLBXENSERVERUNKNOWNHOST 1618

XAPIHOOKFAILED 1618

XENAPIMISSINGPLUGIN 1619

XENAPIPLUGINFAILURE 1619

XENINCOMPATIBLE 1619

XENVSSREQERRORADDINGVOLUMETOSNAPSETFAILED 1619

XENVSSREQERRORCREATINGSNAPSHOT 1619

XENVSSREQERRORCREATINGSNAPSHOTXMLSTRING 1620

XENVSSREQERRORINITFAILED 1620

XENVSSREQERRORNOVOLUMESSUPPORTED 1620

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XENVSSREQERRORPREPARINGWRITERS 1620
XENVSSREQERRORPROVNOTLOADED 1620
XENVSSREQERRORSTARTSNAPSHOTSETFAILED 1621
XMLRPCUNMARSHALFAILURE 1621
XenServer 8.4 Management API 1621
Types 1622
RPCs associated with fields 1623
RPCs associated with classes 1623
Wire Protocol for Remote API Calls 1624
XML-RPC Protocol 1624
JSON-RPC Protocol 1627
Errors 1633
Note on References vs UUIDs 1638
Making RPC Calls 1638
Example interactive session 1640
VM Lifecycle 1645
VM boot parameters 1646
Classes 1647
Relationships Between Classes 1651
Types 1655
Primitives 1655
Higher-order types 1656
Enumeration types 1656
Class: auth 1683
Fields for class: auth 1683
RPCs associated with class: auth 1683
Class: blob 1685
Fields for class: blob 1685

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: blob 1686
Class: Bond 1695
Fields for class: Bond 1696
RPCs associated with class: Bond 1697
Class: Certificate 1708
Fields for class: Certificate 1709
RPCs associated with class: Certificate 1710
Class: Cluster 1717
Fields for class: Cluster 1718
RPCs associated with class: Cluster 1720
Class: Clusterhost 1738
Fields for class: Clusterhost 1738
RPCs associated with class: Clusterhost 1741
Class: console 1752
Fields for class: console 1752
RPCs associated with class: console 1753
Class: crashdump 1760
Fields for class: crashdump 1761
RPCs associated with class: crashdump 1761
Class: datasource 1768
Fields for class: datasource 1768
RPCs associated with class: datasource 1769
Class: DRtask 1769
Fields for class: DRtask 1769
RPCs associated with class: DRtask 1770
Class: Drivervariant 1774
Fields for class: Drivervariant 1774
RPCs associated with class: Drivervariant 1775

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: event 1782
Fields for class: event 1782
RPCs associated with class: event 1782
Class: Feature 1787
Fields for class: Feature 1787
RPCs associated with class: Feature 1787
Class: GPUgroup 1794
Fields for class: GPUgroup 1794
RPCs associated with class: GPUgroup 1795
Class: host 1809
Fields for class: host 1809
RPCs associated with class: host 1818
Class: hostcpu 1931
Fields for class: hostcpu 1932
RPCs associated with class: hostcpu 1933
Class: hostcrashdump 1945
Fields for class: hostcrashdump 1945
RPCs associated with class: hostcrashdump 1946
Class: Hostdriver 1954
Fields for class: Hostdriver 1954
RPCs associated with class: Hostdriver 1955
Class: hostmetrics 1964
Fields for class: hostmetrics 1965
RPCs associated with class: hostmetrics 1965
Class: hostpatch 1973
Fields for class: hostpatch 1973
RPCs associated with class: hostpatch 1974
Class: LVHD 1986

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: LVHD 1986
RPCs associated with class: LVHD 1986
Class: message 1989
Fields for class: message 1989
RPCs associated with class: message 1989
Class: network 1995
Fields for class: network 1995
RPCs associated with class: network 1997
Class: networksriov 2018
Fields for class: networksriov 2018
RPCs associated with class: networksriov 2018
Class: Observer 2025
Fields for class: Observer 2026
RPCs associated with class: Observer 2027
Class: PBD 2039
Fields for class: PBD 2039
RPCs associated with class: PBD 2040
Class: PCI 2050
Fields for class: PCI 2050
RPCs associated with class: PCI 2051
Class: PGPU 2066
Fields for class: PGPU 2066
RPCs associated with class: PGPU 2067
Class: PIF 2082
Fields for class: PIF 2082
RPCs associated with class: PIF 2086
Class: PIFmetrics 2120
Fields for class: PIFmetrics 2120

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: PIFmetrics 2121
Class: pool 2133
Fields for class: pool 2133
RPCs associated with class: pool 2142
Class: poolpatch 2251
Fields for class: poolpatch 2252
RPCs associated with class: poolpatch 2253
Class: poolupdate 2268
Fields for class: poolupdate 2268
RPCs associated with class: poolupdate 2269
Class: proberesult 2283
Fields for class: proberesult 2283
RPCs associated with class: proberesult 2284
Class: PUSB 2284
Fields for class: PUSB 2284
RPCs associated with class: PUSB 2286
Class: PVScachestorage 2298
Fields for class: PVScachestorage 2299
RPCs associated with class: PVScachestorage 2299
Class: PVSproxy 2306
Fields for class: PVSproxy 2306
RPCs associated with class: PVSproxy 2306
Class: PVSserver 2312
Fields for class: PVSserver 2313
RPCs associated with class: PVSserver 2313
Class: PVSsite 2319
Fields for class: PVSsite 2320
RPCs associated with class: PVSsite 2320

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: Repository 2330
Fields for class: Repository 2330
RPCs associated with class: Repository 2332
Class: role 2345
Fields for class: role 2346
RPCs associated with class: role 2346
Class: SDNcontroller 2354
Fields for class: SDNcontroller 2354
RPCs associated with class: SDNcontroller 2354
Class: secret 2360
Fields for class: secret 2360
RPCs associated with class: secret 2360
Class: session 2368
Fields for class: session 2368
RPCs associated with class: session 2370
Class: SM 2386
Fields for class: SM 2386
RPCs associated with class: SM 2387
Class: SR 2400
Fields for class: SR 2401
RPCs associated with class: SR 2403
Class: srstat 2439
Fields for class: srstat 2439
RPCs associated with class: srstat 2440
Class: subject 2440
Fields for class: subject 2440
RPCs associated with class: subject 2440
Class: task 2448

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: task 2448
RPCs associated with class: task 2450
Class: tunnel 2468
Fields for class: tunnel 2468
RPCs associated with class: tunnel 2469
Class: USBgroup 2480
Fields for class: USBgroup 2480
RPCs associated with class: USBgroup 2481
Class: user 2491
Fields for class: user 2491
RPCs associated with class: user 2491
Class: VBD 2499
Fields for class: VBD 2499
RPCs associated with class: VBD 2501
Class: VBDmetrics 2527
Fields for class: VBDmetrics 2527
RPCs associated with class: VBDmetrics 2528
Class: VDI 2535
Fields for class: VDI 2535
RPCs associated with class: VDI 2539
Class: vdinbdserverinfo 2584
Fields for class: vdinbdserverinfo 2584
RPCs associated with class: vdinbdserverinfo 2585
Class: VGPU 2585
Fields for class: VGPU 2585
RPCs associated with class: VGPU 2586
Class: VGPUtype 2599
Fields for class: VGPUtype 2599

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VGPUtype 2601
Class: VIF 2612
Fields for class: VIF 2612
RPCs associated with class: VIF 2615
Class: VIFmetrics 2646
Fields for class: VIFmetrics 2646
RPCs associated with class: VIFmetrics 2647
Class: VLAN 2654
Fields for class: VLAN 2654
RPCs associated with class: VLAN 2655
Class: VM 2663
Fields for class: VM 2663
RPCs associated with class: VM 2674
Class: VMappliance 2812
Fields for class: VMappliance 2812
RPCs associated with class: VMappliance 2813
Class: VMgroup 2826
Fields for class: VMgroup 2826
RPCs associated with class: VMgroup 2827
Class: VMguestmetrics 2834
Fields for class: VMguestmetrics 2834
RPCs associated with class: VMguestmetrics 2837
Class: VMmetrics 2850
Fields for class: VMmetrics 2850
RPCs associated with class: VMmetrics 2852
Class: VMPP 2865
Fields for class: VMPP 2865
RPCs associated with class: VMPP 2867

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: VMSS 2899
Fields for class: VMSS 2900
RPCs associated with class: VMSS 2901
Class: VTPM 2916
Fields for class: VTPM 2916
RPCs associated with class: VTPM 2917
Class: VUSB 2925
Fields for class: VUSB 2925
RPCs associated with class: VUSB 2926
Error Codes 2936
ACTIVATIONWHILENOTFREE 2936
ADDRESSVIOLATESLOCKINGCONSTRAINT 2936
APPLYGUIDANCEFAILED 2936
APPLYLIVEPATCHFAILED 2937
APPLYUPDATESFAILED 2937
APPLYUPDATESINPROGRESS 2937
AUTHALREADYENABLED 2937
AUTHDISABLEFAILED 2937
AUTHDISABLEFAILEDPERMISSIONDENIED 2938
AUTHDISABLEFAILEDWRONGCREDENTIALS 2938
AUTHENABLEFAILED 2938
AUTHENABLEFAILEDDOMAINLOOKUPFAILED 2938
AUTHENABLEFAILEDINVALIDACCOUNT 2938
AUTHENABLEFAILEDINVALIDOU 2939
AUTHENABLEFAILEDPERMISSIONDENIED 2939
AUTHENABLEFAILEDUNAVAILABLE 2939
AUTHENABLEFAILEDWRONGCREDENTIALS 2939
AUTHISDISABLED 2939

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHSERVICEERROR 2940
AUTHUNKNOWNTYPE 2940
BACKUPSCRIPTFAILED 2940
BALLOONINGTIMEOUTBEFOREMIGRATION 2940
BLOCKEDREPOURL 2940
BOOTLOADERFAILED 2941
BOOTDEVICEPASSTHROUGHDISALLOWED 2941
BRIDGENAMEEXISTS 2941
BRIDGENOTAVAILABLE 2941
BUNDLEREPOSITORYALREADYEXISTS 2941
BUNDLEREPONOTENABLED 2942
BUNDLESYNCFAILED 2942
BUNDLEUNPACKFAILED 2942
CANNOTADDTUNNELTOBONDSLAVE 2942
CANNOTADDTUNNELTOSRIOVLOGICAL 2942
CANNOTADDTUNNELTOVLANONSRIOVLOGICAL 2943
CANNOTADDVLANTOBONDSLAVE 2943
CANNOTCHANGEPIFPROPERTIES 2943
CANNOTCONTACTHOST 2943
CANNOTCREATESTATEFILE 2943
CANNOTDESTROYDISASTERRECOVERYTASK 2944
CANNOTDESTROYSYSTEMNETWORK 2944
CANNOTENABLEREDOLOG 2944
CANNOTEVACUATEHOST 2944
CANNOTFETCHPATCH 2944
CANNOTFINDOEMBACKUPPARTITION 2945
CANNOTFINDPATCH 2945
CANNOTFINDSTATEPARTITION 2945

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTFINDUPDATE 2945
CANNOTFORGETSRIOVLOGICAL 2945
CANNOTPLUGBONDSLAVE 2946
CANNOTPLUGVIF 2946
CANNOTRESETCONTROLDOMAIN 2946
CANNOTRESTARTDEVICEMODEL 2946
CANNOTPERIODICSYNCUPDATES 2946
CANNOTSYNCUPDATES 2947
CERTIFICATEALREADYEXISTS 2947
CERTIFICATECORRUPT 2947
CERTIFICATEDOESNOTEXIST 2947
CERTIFICATELACKSPURPOSE 2947
CERTIFICATELIBRARYCORRUPT 2948
CERTIFICATENAMEINVALID 2948
CHANGEPASSWORDREJECTED 2948
CLUSTEREDSRDEGRADED 2948
CLUSTERINGDISABLED 2948
CLUSTERINGENABLED 2949
CLUSTERALREADYEXISTS 2949
CLUSTERCREATEINPROGRESS 2949
CLUSTERDOESNOTHAVEONENODE 2949
CLUSTERFORCEDESTROYFAILED 2949
CLUSTERHOSTISLAST 2950
CLUSTERHOSTNOTJOINED 2950
CLUSTERSTACKINUSE 2950
CONFIGUREREPOSITORIESINPROGRESS 2950
COULDNOTFINDNETWORKINTERFACEWITHSPECIFIEDDEVICENAMEANDMACADDRESS 2950
COULDNOTIMPORTDATABASE 2951

© 1997–2026 Citrix Systems, Inc. All rights reserved.

COULDNOTUPDATEIGMPSNOOPINGEVERYWHERE 2951
CPUFEATUREMASKINGNOTSUPPORTED 2951
CRLALREADYEXISTS 2951
CRLCORRUPT 2951
CRLDOESNOTEXIST 2952
CRLNAMEINVALID 2952
DBUNIQUENESSCONSTRAINTVIOLATION 2952
DEFAULTSRNOTFOUND 2952
DEVICEALREADYATTACHED 2952
DEVICEALREADYDETACHED 2953
DEVICEALREADYEXISTS 2953
DEVICEATTACHTIMEOUT 2953
DEVICEDETACHREJECTED 2953
DEVICEDETACHTIMEOUT 2953
DEVICENOTATTACHED 2954
DISABLESSHFAILED 2954
DISABLESSHPARTIALLYFAILED 2954
DISKVBDMUSTBEREADWRITEFORHVM 2954
DOMAINBUILDERERROR 2954
DOMAINEXISTS 2955
DUPLICATEMACSEED 2955
DUPLICATEPIFDEVICENAME 2955
DUPLICATEVM 2955
ENABLESSHFAILED 2955
ENABLESSHPARTIALLYFAILED 2956
EVENTSLOST 2956
EVENTFROMTOKENPARSEFAILURE 2956
EVENTSUBSCRIPTIONPARSEFAILURE 2956

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FAILEDTOSTARTEMULATOR 2956

FEATUREREQUIRESHVM 2957

FEATURERESTRICTED 2957

FIELDTYPEERROR 2957

GETHOSTUPDATESFAILED 2957

GETUPDATESFAILED 2957

GPUGROUPCONTAINSNOPGPUS 2958

GPUGROUPCONTAINSPGPU 2958

GPUGROUPCONTAINSVGPU 2958

HANDLEINVALID 2958

HAABORTNEWMASTER 2958

HACANNOTCHANGEBONDSTATUSOFMGMTIFACE 2959

HACONSTRAINTVIOLATIONNETWORKNOTSHARED 2959

HACONSTRAINTVIOLATIONSRNOTSHARED 2959

HADISABLEINPROGRESS 2959

HAENABLEINPROGRESS 2959

HAFAILEDTOFORMLIVESET 2960

HAHEARTBEATDAEMONSTARTUPFAILED 2960

HAHOSTCANNOTACCESSSTATEFILE 2960

HAHOSTCANNOTSEEPEERS 2960

HAHOSTISARMED 2960

HAISENABLED 2961

HALOSTSTATEFILE 2961

HANOTENABLED 2961

HANOTINSTALLED 2961

HANOPLAN 2961

HAOPERATIONWOULDBREAKFAILOVERPLAN 2962

HAPOOLISENABLEDBUTHOSTISDISABLED 2962

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HASHOULDBEFENCED 2962

HATOOFEWHOSTS 2962

HOSTSNOTCOMPATIBLE 2962

HOSTSNOTHOMOGENEOUS 2963

HOSTBROKEN 2963

HOSTCANNOTATTACHNETWORK 2963

HOSTCANNOTDESTROYSELF 2963

HOSTCANNOTREADMETRICS 2963

HOSTCDDRIVEEMPTY 2964

HOSTDISABLED 2964

HOSTDISABLEDINDEFINITELY 2964

HOSTDISABLEDUNTILREBOOT 2964

HOSTDRIVERNOHARDWARE 2964

HOSTEVACUATEINPROGRESS 2965

HOSTEVACUATIONISREQUIRED 2965

HOSTHASNOMANAGEMENTIP 2965

HOSTHASRESIDENTVMS 2965

HOSTINEMERGENCYMODE 2965

HOSTINUSE 2966

HOSTISLIVE 2966

HOSTISSLAVE 2966

HOSTITSOWNSLAVE 2966

HOSTMASTERCANNOTTALKBACK 2966

HOSTNAMEINVALID 2967

HOSTNOTDISABLED 2967

HOSTNOTENOUGHFREEMEMORY 2967

HOSTNOTENOUGHPCPUS 2967

HOSTNOTLIVE 2967

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTOFFLINE 2968

HOSTPENDINGMANDATORYGUIDANCENOTEMPTY 2968

HOSTPOWERONMODEDISABLED 2968

HOSTSTILLBOOTING 2968

HOSTUNKNOWNTOMASTER 2968

HOSTXAPIVERSIONHIGHERTHANCOORDINATOR 2969

ILLEGALVBDDEVICE 2969

IMPORTERROR 2969

IMPORTERRORATTACHEDDISKSNOTFOUND 2969

IMPORTERRORCANNOTHANDLECHUNKED 2969

IMPORTERRORFAILEDTOFINDOBJECT 2970

IMPORTERRORPREMATUREEOF 2970

IMPORTERRORSOMECHECKSUMSFAILED 2970

IMPORTERRORUNEXPECTEDFILE 2970

IMPORTINCOMPATIBLEVERSION 2970

INCOMPATIBLECLUSTERSTACKACTIVE 2971

INCOMPATIBLEPIFPROPERTIES 2971

INCOMPATIBLESTATEFILESR 2971

INTERFACEHASNOIP 2971

INTERNALERROR 2971

INVALIDBASEURL 2972

INVALIDCIDRADDRESSSPECIFIED 2972

INVALIDCLUSTERSTACK 2972

INVALIDDEVICE 2972

INVALIDEDITION 2972

INVALIDFEATURESTRING 2973

INVALIDGPGKEYPATH 2973

INVALIDIPADDRESSSPECIFIED 2973

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDNTPCONFIG 2973

INVALIDPATCH 2973

INVALIDPATCHWITHLOG 2974

INVALIDREPOMDXML 2974

INVALIDREPOSITORYDOMAINALLOWLIST 2974

INVALIDREPOSITORYPROXYCREDENTIAL 2974

INVALIDREPOSITORYPROXYURL 2974

INVALIDUPDATE 2975

INVALIDUPDATEINFOXML 2975

INVALIDUPDATESYNCDAY 2975

INVALIDVALUE 2975

ISTUNNELACCESSPIF 2975

JOININGHOSTCANNOTBEMASTEROFOTHERHOSTS 2976

JOININGHOSTCANNOTCONTAINSHAREDSRS 2976

JOININGHOSTCANNOTHAVERUNNINGORSUSPENDEDVMS 2976

JOININGHOSTCANNOTHAVERUNNINGVMS 2976

JOININGHOSTCANNOTHAVEVMSWITHCURRENTOPERATIONS 2976

JOININGHOSTCONNECTIONFAILED 2977

JOININGHOSTSERVICEFAILED 2977

LICENCERESTRICTION 2977

LICENSECANNOTDOWNGRADEWHILEINPOOL 2977

LICENSECHECKOUTERROR 2977

LICENSEDOESNOTSUPPORTPOOLING 2978

LICENSEDOESNOTSUPPORTXHA 2978

LICENSEEXPIRED 2978

LICENSEFILEDEPRECATED 2978

LICENSEHOSTPOOLMISMATCH 2978

LICENSEPROCESSINGERROR 2979

© 1997–2026 Citrix Systems, Inc. All rights reserved.

LOCATIONNOTUNIQUE 2979
MACDOESNOTEXIST 2979
MACINVALID 2979
MACSTILLEXISTS 2979
MAPDUPLICATEKEY 2980
MEMORYCONSTRAINTVIOLATION 2980
MEMORYCONSTRAINTVIOLATIONMAXPIN 2980
MEMORYCONSTRAINTVIOLATIONORDER 2980
MESSAGEDEPRECATED 2980
MESSAGEMETHODUNKNOWN 2981
MESSAGEPARAMETERCOUNTMISMATCH 2981
MESSAGEREMOVED 2981
MIRRORFAILED 2981
MISSINGCONNECTIONDETAILS 2981
MULTIPLEUPDATEREPOSITORIESENABLED 2982
NETWORKALREADYCONNECTED 2982
NETWORKCONTAINSPIF 2982
NETWORKCONTAINSVIF 2982
NETWORKHASINCOMPATIBLESRIOVPIFS 2982
NETWORKHASINCOMPATIBLEVLANONSRIOVPIFS 2983
NETWORKINCOMPATIBLEPURPOSES 2983
NETWORKINCOMPATIBLEWITHBOND 2983
NETWORKINCOMPATIBLEWITHSRIOV 2983
NETWORKINCOMPATIBLEWITHTUNNEL 2983
NETWORKINCOMPATIBLEWITHVLANONBRIDGE 2984
NETWORKINCOMPATIBLEWITHVLANONSRIOV 2984
NETWORKSRIOVALREADYENABLED 2984
NETWORKSRIOVDISABLEFAILED 2984

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKSRIOVENABLEFAILED 2984

NETWORKSRIOVINSUFFICIENTCAPACITY 2985

NETWORKUNMANAGED 2985

NOTALLOWEDONOEMEDITION 2985

NOTALLOWEDWHENNTPISENABLED 2985

NOTIMPLEMENTED 2985

NOTINEMERGENCYMODE 2986

NOTSUPPORTEDDURINGUPGRADE 2986

NOTSYSTEMDOMAIN 2986

NOTTRUSTEDCERTIFICATE 2986

NOCLUSTERHOSTSREACHABLE 2986

NOCOMPATIBLECLUSTERHOST 2987

NOHOSTSAVAILABLE 2987

NOMOREREDOLOGSALLOWED 2987

NOREPOSITORIESCONFIGURED 2987

NOREPOSITORYENABLED 2987

NVIDIASRIOVMISCONFIGURED 2988

NVIDIATOOLSERROR 2988

OBJECTNOLONGEREXISTS 2988

ONLYALLOWEDONOEMEDITION 2988

OPENVSWITCHNOTACTIVE 2988

OPERATIONBLOCKED 2989

OPERATIONNOTALLOWED 2989

OPERATIONPARTIALLYFAILED 2989

OTHEROPERATIONINPROGRESS 2989

OUTOFSPACE 2989

PATCHALREADYAPPLIED 2990

PATCHALREADYEXISTS 2990

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PATCHAPPLYFAILED 2990

PATCHAPPLYFAILEDBACKUPFILESEXIST 2990

PATCHISAPPLIED 2990

PATCHPRECHECKFAILEDISOMOUNTED 2991

PATCHPRECHECKFAILEDOUTOFSPACE 2991

PATCHPRECHECKFAILEDPREREQUISITEMISSING 2991

PATCHPRECHECKFAILEDUNKNOWNERROR 2991

PATCHPRECHECKFAILEDVMRUNNING 2991

PATCHPRECHECKFAILEDWRONGSERVERBUILD 2992

PATCHPRECHECKFAILEDWRONGSERVERVERSION 2992

PBDEXISTS 2992

PERMISSIONDENIED 2992

PGPUINSUFFICIENTCAPACITYFORVGPU 2992

PGPUINUSEBYVM 2993

PGPUNOTCOMPATIBLEWITHGPUGROUP 2993

PIFALLOWSUNPLUG 2993

PIFALREADYBONDED 2993

PIFBONDMORETHANONEIP 2993

PIFBONDNEEDSMOREMEMBERS 2994

PIFCANNOTBONDCROSSHOST 2994

PIFCONFIGURATIONERROR 2994

PIFDEVICENOTFOUND 2994

PIFDOESNOTALLOWUNPLUG 2994

PIFHASFCOESRINUSE 2995

PIFHASNONETWORKCONFIGURATION 2995

PIFHASNOV6NETWORKCONFIGURATION 2995

PIFINCOMPATIBLEPRIMARYADDRESSTYPE 2995

PIFISMANAGEMENTINTERFACE 2995

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFISNOTPHYSICAL 2996

PIFISNOTSRIOVCAPABLE 2996

PIFISPHYSICAL 2996

PIFISSRIOVLOGICAL 2996

PIFISVLAN 2996

PIFNOTATTACHEDTOHOST 2997

PIFNOTPRESENT 2997

PIFSRIOVSTILLEXISTS 2997

PIFTUNNELSTILLEXISTS 2997

PIFUNMANAGED 2997

PIFVLANEXISTS 2998

PIFVLANSTILLEXISTS 2998

POOLAUTHALREADYENABLED 2998

POOLAUTHDISABLEFAILED 2998

POOLAUTHDISABLEFAILEDINVALIDACCOUNT 2998

POOLAUTHDISABLEFAILEDPERMISSIONDENIED 2999

POOLAUTHDISABLEFAILEDWRONGCREDENTIALS 2999

POOLAUTHENABLEFAILED 2999

POOLAUTHENABLEFAILEDDOMAINLOOKUPFAILED 2999

POOLAUTHENABLEFAILEDDUPLICATEHOSTNAME 2999

POOLAUTHENABLEFAILEDINVALIDACCOUNT 3000

POOLAUTHENABLEFAILEDINVALIDOU 3000

POOLAUTHENABLEFAILEDPERMISSIONDENIED 3000

POOLAUTHENABLEFAILEDUNAVAILABLE 3000

POOLAUTHENABLEFAILEDWRONGCREDENTIALS 3000

POOLJOININGEXTERNALAUTHMISMATCH 3001

POOLJOININGHOSTCACERTIFICATESCONFLICT 3001

POOLJOININGHOSTHASBONDS 3001

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGHOSTHASNETWORKSRIOVS	3001
POOLJOININGHOSTHASNONMANAGEMENTVLANS	3001
POOLJOININGHOSTHASTUNNELS	3002
POOLJOININGHOSTMANAGEMENTVLANDOESNOTMATCH	3002
POOLJOININGHOSTMUSTHAVEONLYONEIPONCLUSTERINGNETWORK	3002
POOLJOININGHOSTMUSTHAVEPHYSICALMANAGEMENTNIC	3002
POOLJOININGHOSTMUSTHAVESAMEAPIVERSION	3002
POOLJOININGHOSTMUSTHAVESAMEDBSCHEMA	3003
POOLJOININGHOSTMUSTHAVESAMEPRODUCTVERSION	3003
POOLJOININGHOSTMUSTONLYHAVEPHYSICALPIFS	3003
POOLJOININGHOSTTLSVERIFICATIONMISMATCH	3003
POOLJOININGHOSTTRUSTEDCERTIFICATESCONFLICT	3003
POOLJOININGMASTERCERTIFICATENOTINPOOLBUNDLE	3004
POOLJOININGPOOLBUNDLEEMPTYAFTERIMPORT	3004
POOLJOININGPOOLCANNOTENABLECLUSTERINGONVLANNETWORK	3004
POOLJOININGSMFEATURESINCOMPATIBLE	3004
PROVISIONFAILEDOUTOFSPACE	3004
PROVISIONONLYALLOWEDONTEMPLATE	3005
PUSBVDICONFLICT	3005
PVSCACHESTORAGEALREADYPRESENT	3005
PVSCACHESTORAGEISINUSE	3005
PVSPROXYALREADYPRESENT	3005
PVSPROXYPRESENTONHIGHERVIFDEVICE	3006
PVSSERVERADDRESSINUSE	3006
PVSSITECONTAINSRUNNINGPROXIES	3006
PVSSITECONTAINSSERVERS	3006
PVSVIFMUSTBEFIRSTDEVICE	3006
RBACPERMISSIONDENIED	3007

© 1997–2026 Citrix Systems, Inc. All rights reserved.

REDOLOGISENABLED	3007
REPOSITORYALREADYEXISTS	3007
REPOSITORYCLEANUPFAILED	3007
REPOSITORYISINUSE	3007
REPOSYNCFAILED	3008
REPOSHOULDBESINGLEONEENABLED	3008
REQUIREDPIFISUNPLUGGED	3008
RESTOREINCOMPATIBLEVERSION	3008
RESTORESCRIPTFAILED	3008
RESTORETARGETMGMTIFNOTINBACKUP	3009
RESTORETARGETMISSINGDEVICE	3009
ROLEALREADYEXISTS	3009
ROLENOTFOUND	3009
SERVERCERTIFICATECHAININVALID	3009
SERVERCERTIFICATEEXPIRED	3010
SERVERCERTIFICATEINVALID	3010
SERVERCERTIFICATEKEYALGORITHMNOTSUPPORTED	3010
SERVERCERTIFICATEKEYINVALID	3010
SERVERCERTIFICATEKEYMISMATCH	3010
SERVERCERTIFICATEKEYRSALENGTHNOTSUPPORTED	3011
SERVERCERTIFICATEKEYRSAMULTINOTSUPPORTED	3011
SERVERCERTIFICATENOTVALIDYET	3011
SERVERCERTIFICATESIGNATURENOTSUPPORTED	3011
SESSIONAUTHENTICATIONFAILED	3011
SESSIONAUTHORIZATIONFAILED	3012
SESSIONINVALID	3012
SESSIONNOTREGISTERED	3012
SETCONSOLETIMEOUTPARTIALLYFAILED	3012

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SETSSHAUTOMODEPARTIALLYFAILED 3012

SETSSHTIMEOUTPARTIALLYFAILED 3013

SLAVEREQUIRESMANAGEMENTINTERFACE 3013

SMPLUGINCOMMUNICATIONFAILURE 3013

SRATTACHFAILED 3013

SRBACKENDFAILURE 3013

SRDEVICEINUSE 3014

SRDOESNOTSUPPORTMIGRATION 3014

SRFULL 3014

SRHASMULTIPLEPBDS 3014

SRHASNOPBDS 3014

SRHASPBD 3015

SRINDESTRUCTIBLE 3015

SRISCACHESR 3015

SRNOTATTACHED 3015

SRNOTEMPTY 3015

SRNOTSHARABLE 3016

SROPERATIONNOTSUPPORTED 3016

SRREQUIRESUPGRADE 3016

SRSOURCESPACEINSUFFICIENT 3016

SRSUSPENDSPACEINSUFFICIENT 3016

SRUNHEALTHY 3017

SRUNKNOWNDRIVER 3017

SRUUIDEXISTS 3017

SRVDILOCKINGFAILED 3017

SSLVERIFYERROR 3017

SUBJECTALREADYEXISTS 3018

SUBJECTCANNOTBERESOLVED 3018

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SUSPENDIMAGENOTACCESSIBLE 3018
SYNCBUNDLEINPROGRESS 3018
SYNCUPDATESINPROGRESS 3018
SYSPREP 3019
SYSTEMSTATUSMUSTUSETARONOEM 3019
SYSTEMSTATUSRETRIEVALFAILED 3019
TASKCANCELLED 3019
TELEMETRYNEXTCOLLECTIONTOOLATE 3019
TLSCONNECTIONFAILED 3020
TLSVERIFICATIONNOTENABLEDINPOOL 3020
TOOBUSY 3020
TOOMANYGROUPS 3020
TOOMANYPENDINGTASKS 3020
TOOMANYSTORAGEMIGRATES 3021
TOOMANYVUSBS 3021
TRANSPORTPIFNOTCONFIGURED 3021
TRUSTEDCERTIFICATEALREADYEXISTS 3021
TRUSTEDCERTIFICATEEXPIRED 3021
TRUSTEDCERTIFICATEINVALID 3022
TRUSTEDCERTIFICATENOTVALIDYET 3022
UNIMPLEMENTEDINSMBACKEND 3022
UNKNOWNBOOTLOADER 3022
UPDATEINFOHASHMISMATCH 3022
UPDATEALREADYAPPLIED 3023
UPDATEALREADYAPPLIEDINPOOL 3023
UPDATEALREADYEXISTS 3023
UPDATEAPPLYFAILED 3023
UPDATEISAPPLIED 3023

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATEPOOLAPPLYFAILED 3024

UPDATEPRECHECKFAILEDCONFLICTPRESENT 3024

UPDATEPRECHECKFAILEDGPGKEYNOTIMPORTED 3024

UPDATEPRECHECKFAILEDOUTOFSPACE 3024

UPDATEPRECHECKFAILEDPREREQUISITEMISSING 3024

UPDATEPRECHECKFAILEDUNKNOWNERROR 3025

UPDATEPRECHECKFAILEDWRONGSERVERVERSION 3025

UPDATESYNCINGREMOTEPOOLCOORDINATORCONNECTIONFAILED 3025

UPDATESYNCINGREMOTEPOOLCOORDINATORSERVICEFAILED 3025

USBALREADYATTACHED 3025

USBGROUPCONFLICT 3026

USBGROUPCONTAINSNOPUSBS 3026

USBGROUPCONTAINSPUSB 3026

USBGROUPCONTAINSVUSB 3026

USERISNOTLOCALSUPERUSER 3026

UUIDINVALID 3027

V6DFAILURE 3027

VALUENOTSUPPORTED 3027

VBDCDSMUSTBEREADONLY 3027

VBDISEMPTY 3027

VBDNOTEMPTY 3028

VBDNOTREMOVABLEMEDIA 3028

VBDNOTUNPLUGGABLE 3028

VBDTRAYLOCKED 3028

VCPUMAXNOTCORESPERSOCKETMULTIPLE 3028

VDICBTENABLED 3029

VDICONTAINSMETADATAOFTHISPOOL 3029

VDICOPYFAILED 3029

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDIHASRRDS 3029

VDIINCOMPATIBLETYPE 3029

VDIINUSE 3030

VDIISAPHYSICALDEVICE 3030

VDIISENCRYPTED 3030

VDIISNOTISO 3030

VDILOCATIONMISSING 3030

VDIMISSING 3031

VDINEEDSVMFORMIGRATE 3031

VDINOTAVAILABLE 3031

VDINOTINMAP 3031

VDINOTMANAGED 3031

VDINOTSPARSE 3032

VDINOCBTMETADATA 3032

VDIONBOOTMODEINCOMPATIBLEWITHOPERATION 3032

VDIREADONLY 3032

VDITOOLARGE 3032

VDITOOSMALL 3033

VGPUDESTINATIONINCOMPATIBLE 3033

VGPUGUESTDRIVERLIMIT 3033

VGPUSUSPENSIONNOTSUPPORTED 3033

VGPUTYPENOTCOMPATIBLE 3033

VGPUTYPENOTCOMPATIBLEWITHRUNNINGTYPE 3034

VGPUTYPENOTENABLED 3034

VGPUTYPENOTSUPPORTED 3034

VGPUTYPENOLONGERSUPPORTED 3034

VIFINUSE 3034

VIFNOTINMAP 3035

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VLANINUSE 3035

VLANTAGINVALID 3035

VMPPARCHIVEMOREFREQUENTTHANBACKUP 3035

VMPPHASVM 3035

VMSSHASVM 3036

VMSFAILEDTOCOOPERATE 3036

VMASSIGNEDTOPROTECTIONPOLICY 3036

VMASSIGNEDTOSNAPSHOTSCHEDULE 3036

VMATTACHEDTOMORETHANONEVDIWITHTIMEOFFSETMARKEDASRESETONBOOT 3036

VMBADPOWERSTATE 3037

VMBIOSSTRINGSALREADYSET 3037

VMCALLPLUGINRATELIMIT 3037

VMCANNOTDELETEDEFAULTTEMPLATE 3037

VMCHECKPOINTRESUMEFAILED 3037

VMCHECKPOINTSUSPENDFAILED 3038

VMCRASHED 3038

VMDUPLICATEVBDDEVICE 3038

VMFAILEDSHUTDOWNACKNOWLEDGMENT 3038

VMFAILEDSUSPENDACKNOWLEDGMENT 3038

VMHALTED 3039

VMHASCHECKPOINT 3039

VMHASNOSUSPENDVDI 3039

VMHASPCIATTACHED 3039

VMHASSRIOVVIF 3039

VMHASTOOMANYSNAPSHOTS 3040

VMHASVGPU 3040

VMHASVUSBS 3040

VMHOSTINCOMPATIBLEVERSION 3040

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMHOSTINCOMPATIBLEVERSIONMIGRATE 3040

VMHOSTINCOMPATIBLEVIRTUALHARDWAREPLATFORMVERSION 3041

VMHVMREQUIRED 3041

VMINCOMPATIBLEWITHTHISHOST 3041

VMISIMMOBILE 3041

VMISPARTOFANAPPLIANCE 3041

VMISPROTECTED 3042

VMISTEMPLATE 3042

VMISUSINGNESTEDVIRT 3042

VMLACKSFEATURE 3042

VMMEMORYSIZETOOLOW 3042

VMMIGRATECONTACTREMOTESERVICEFAILED 3043

VMMIGRATEFAILED 3043

VMMISSINGPVDRIVERS 3043

VMNONSUSPENDABLE 3043

VMNOTRESIDENTHERE 3043

VMNOCRASHDUMPSR 3044

VMNOEMPTYCDVBD 3044

VMNOSUSPENDSR 3044

VMNOVCPUS 3044

VMOLDPVDRIVERS 3044

VMPCIBUSFULL 3045

VMPVDRIVERSINUSE 3045

VMREBOOTED 3045

VMREQUIRESGPU 3045

VMREQUIRESIOMMU 3045

VMREQUIRESNETWORK 3046

VMREQUIRESSR 3046

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMREQUIRESVDI 3046

VMREQUIRESVGPU 3046

VMREQUIRESVUSB 3046

VMREVERTFAILED 3047

VMSHUTDOWNTIMEOUT 3047

VMSNAPSHOTWITHQUIESCEFAILED 3047

VMSNAPSHOTWITHQUIESCENOTSUPPORTED 3047

VMSNAPSHOTWITHQUIESCEPLUGINDEOSNOTRESPOND 3047

VMSNAPSHOTWITHQUIESCETIMEOUT 3048

VMSUSPENDTIMEOUT 3048

VMTOOMANYVCPUS 3048

VMTOIMPORTISNOTNEWERVERSION 3048

VMUNSAFEBOOT 3048

VTPMMAXAMOUNTREACHED 3049

WLBAUTHENTICATIONFAILED 3049

WLBCONNECTIONREFUSED 3049

WLBCONNECTIONRESET 3049

WLBDISABLED 3049

WLBINTERNALERROR 3050

WLBMALFORMEDREQUEST 3050

WLBMALFORMEDRESPONSE 3050

WLBNOTINITIALIZED 3050

WLBTIMEOUT 3050

WLBUNKNOWNHOST 3051

WLBURLINVALID 3051

WLBXENSERVERAUTHENTICATIONFAILED 3051

WLBXENSERVERCONNECTIONREFUSED 3051

WLBXENSERVERMALFORMEDRESPONSE 3051

© 1997–2026 Citrix Systems, Inc. All rights reserved.

WLBXENSERVERTIMEOUT 3052
WLBXENSERVERUNKNOWNHOST 3052
XAPIHOOKFAILED 3052
XENAPIMISSINGPLUGIN 3052
XENAPIPLUGINFAILURE 3052
XENINCOMPATIBLE 3053
XENVSSREQERRORADDINGVOLUMETOSNAPSETFAILED 3053
XENVSSREQERRORCREATINGSNAPSHOT 3053
XENVSSREQERRORCREATINGSNAPSHOTXMLSTRING 3053
XENVSSREQERRORINITFAILED 3053
XENVSSREQERRORNOVOLUMESSUPPORTED 3054
XENVSSREQERRORPREPARINGWRITERS 3054
XENVSSREQERRORPROVNOTLOADED 3054
XENVSSREQERRORSTARTSNAPSHOTSETFAILED 3054
XMLRPCUNMARSHALFAILURE 3054
XenServer® Changed Block Tracking Guide 3055
How changed block tracking works 3055
Benefits of changed block tracking 3055
Getting started using changed block tracking 3056
Prerequisites 3057
Procedure 3057
Setting up changed block tracking 3057
Taking incremental backups 3060
Restoring a VDI from exported changed block data 3062
Enabling NBD connections on XenServer® 3065
Enabling an NBD connection for a network (`FORCEDTLS` mode) 3066
Examples 3066

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enabling an insecure NBD connection for a network (`NOTLS` mode) 3066
Examples 3067
Disabling NBD connections for a network 3067
Examples 3067
Using changed block tracking with a virtual disk image 3068
Incremental backup sets 3068
Enabling changed block tracking for a VDI 3068
Examples 3069
Errors 3069
Disabling changed block tracking for a VDI 3070
Examples 3070
Errors 3071
Checking whether changed block tracking is enabled 3071
Examples 3071
Deleting VDI snapshot data and retaining the snapshot metadata 3071
Examples 3072
Errors 3072
Checking the type of a VDI or VDI snapshot 3073
Examples 3073
Getting the list of blocks that changed between VDIs 3074
Examples 3074
Errors 3075
Export changed blocks over a network block device connection 3075
Getting NBD connection information for a VDI 3076
Examples 3077
Errors 3077
Exporting the changed blocks using an NBD client 3078
Verifying TLS certificates for NBD connections 3078

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Coalescing changed blocks onto a base VDI 3079
Examples 3081
Troubleshoot Changed Block Tracking 3082
Common error scenarios 3082
You can't enable changed block tracking on a VDI 3082
You can't data destroy a VDI 3082
You can't list changed blocks between two VDI snapshots 3083
Notable error conditions 3083
Troubleshooting the NBD server 3084
Common NBD connection issues 3084
Appendices 3085
Constraints 3085
Third-party solutions 3086
Additional Resources 3086
fix 3087

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Develop for XenServer®

XenServer provides an SDK that enables you to develop or script clients that work with XenServer in the following languages: PowerShell, Python, C#, C, Go, and Java.

To get started developing for XenServer, see our SDK guide.

If you already know what you are doing and want to look up a class or method, see our API reference for XenServer 8.4 or XenServer 9.

Other development tools

To develop an incremental backup solution, see Changed Block Tracking.

Develop drivers and supplemental packs

To develop drivers, supplemental packs, or other software that requires changes to XenServer, contact your Citrix Ready® Partner contact for further information. If you are not yet a Citrix Ready Partner, learn more at Citrix Ready Partners.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XenServer® Software Development Kit

Welcome to the developer's guide for XenServer. Here you will find the information you need in order to understand and use the Software Development Kit (SDK) that XenServer provides. This information will provide you with some of the architectural background and thinking that underpins the APIs, the tools that have been provided, and how to quickly get off the ground.

Getting Started

XenServer includes a Remote Procedure Call (RPC) based API providing programmatic access to the extensive set of XenServer management features and tools. You can call the XenServer Management API from a remote system or from local to the XenServer host.

It's possible to write applications that use the XenServer Management API directly through raw RPC calls. However, the task of developing third-party applications is greatly simplified by using a language binding. These language bindings expose the individual API calls as first-class functions in the target language. The XenServer SDK provides language bindings for the C, C#, Go, Java, Python, and PowerShell programming languages.

System Requirements and Preparation

The first step towards using the SDK is to install XenServer. XenServer is available for download at `<https://www.xenserver.com/downloads>`. For detailed instructions on how to set up your development host, see the Install XenServer.

When the installation is complete, note the *host IP address* and the *host password*.

Downloading

The SDK is packaged as a ZIP file and is available as a free download from `<https://www.xenserver.com/downloads>`.

The Python module is also available as a package on PyPi. See section SDK Languages - Python for details.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SDK contents

The extracted contents of the SDK ZIP file are in the `XenServer-SDK` directory. Where necessary, subdirectories have their own individual README files.

NOTE:

The contents of the SDK ZIP consist of library binaries and their source code. Previous releases of the SDK ZIP included a number of examples for each of the SDK languages to help you get started with the SDK. These examples have now been removed from the SDK ZIP and are available at XenServer SDK usage examples on GitHub.

The examples provided aren't the same across all the SDK languages. If you intend to use one language, it's advisable to browse the sample code available in the others as well.
The top level of the `XenServer-SDK` directory includes the XenServer Management API Reference document. This document provides an overview of the API types and classes.

The API supports two wire formats, one based upon XML-RPC and one based upon JSON-RPC (v1.0 and v2.0 are both recognised). For more information on the API semantics and the wire protocol of the RPC messages, see section XenServer Management API. The format supported by each of the SDK languages is specified in the following paragraphs.

Supported languages

The XenServer SDK provides support for the following languages:

- PowerShell
- C #
- Python
- Java
- C (deprecated)
- Go

Command line interface (CLI)

Besides using raw RPC or one of the supplied SDK languages, third-party software developers can integrate with XenServer hosts by using the xe command line interface `xe`. The xe CLI is installed by default on XenServer hosts. A stand-alone remote CLI is also available for Linux. On Windows, the `xe.exe` CLI executable is installed along with XenCenter.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Platform supported:

- Linux
- Windows

Library:

- None

Binary:

- xe on Linux
- xe.exe on Windows

Dependencies:

- None

The CLI allows almost every API call to be directly invoked from a script or other program, silently taking care of the required session management. The xe CLI syntax and capabilities are described in detail in the Command line interface documentation.

> **NOTE:**
>
> When running the CLI from a XenServer host console, tab completion of both command names and arguments is available.

Getting Started with PowerShell

The `XenServer-SDK` directory contains the following folders that are relevant to PowerShell users:

- `XenServerPowerShell`
  - `PowerShell_7`: The XenServer SDK for PowerShell 7.
    - `XenServerPSModule`: The XenServer PowerShell module for PowerShell 7.
    - `src`: C# source code for the XenServer PowerShell cmdlets.
  - `PowerShell_51`: The XenServer SDK for PowerShell 5.1.
    - `XenServerPSModule`: The XenServer PowerShell module for PowerShell 5.1.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

    - `src`: C# source code for the XenServer PowerShell cmdlets.

RPC protocol

The PowerShell SDK supports the JSON-RPC v2.0 protocol.

Platform supported

XenServer® SDK for PowerShell 7:

- Windows
- Linux

This library requires .NET 8.0 and PowerShell 7.4 or greater.

XenServer SDK for PowerShell 5.1:

- Windows

This library requires .NET Framework 4.5 and PowerShell 5.1.

Library

- `XenServerPSModule`

NOTE:

This module is generally, but not fully, backwards compatible. To interact with hosts running older versions of XenServer or Citrix Hypervisor™, it is advisable to use the module of the same version as the host.

Dependencies

- Newtonsoft JSON.NET by James Newton-King.
- XenServer.NET, the C# SDK for XenServer.

Installation

First, extract the contents of the SDK ZIP file.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTE:

Some web browsers may mark the SDK ZIP file as "blocked" during the download. To import the module successfully you will need to unblock the SDK ZIP file before extracting its contents. To unblock the SDK ZIP file, right-click on it and launch the **Properties** dialog. Click the **Unblock** button, then the **Apply** or **OK** button.

XenServer SDK for PowerShell 7:

1. Navigate to the extracted `XenServer-SDK\XenServerPowerShell\PowerShell_7` directory and copy the whole folder `XenServerPSModule` into your PowerShell modules directory.

   - On Windows this will normally be `$env:UserProfile\Documents\PowerShell\Modules` for per-user configuration, or `$env:ProgramFiles\PowerShell\7\Modules` for system-wide configuration.
   - On Linux this will be `~/.local/share/powershell/Modules` for per-user configuration, or `/usr/local/share/powershell/Modules` for system-wide configuration.

   For more information see PowerShell's documentation on module paths:

   ```
   Get-Help about_PSModulePath
   ```

2. Open a PowerShell 7 prompt as administrator.

   To do this, open the Windows **Start** menu by clicking the **Start** icon, find the item **PowerShell 7**, right-click it and select **Run as administrator**.

3. Determine the current execution policy:

   ```
   Get-ExecutionPolicy
   ```

   If the current policy is `Restricted`, you need to set it to `RemoteSigned`:

   ```
   Set-ExecutionPolicy RemoteSigned
   ```

   You should understand the security implications of this change. If you are unsure, see PowerShell's documentation on execution policies:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Get-Help about_Execution_Policies
```

If the current policy is `AllSigned`, you will be able to use the XenServer PowerShell module, but it will be inconvenient because this policy requires even scripts that you write on the local computer to be signed. You may want to change it to `RemoteSigned`, as above.

If the current policy is `RemoteSigned`, `ByPass`, or `Unrestricted` there is nothing to do.

4. Exit the privileged instance of PowerShell.
5. Open a PowerShell 7 prompt as a regular user (click **Start** > **PowerShell 7**) and import the XenServer PowerShell module:

```
Import-Module XenServerPSModule
```

6. If you wish to load specific environment settings when the XenServer PowerShell module is loaded, create the file `XenServerProfile.ps1` and put it in the folder containing your `$PROFILE` file for per-user configuration, or in `$PSHOME` for system-wide configuration.
   - On Windows these will normally be `$env:UserProfile\Documents\PowerShell` for per-user configuration, or `$env:ProgramFiles\PowerShell\7` for system-wide configuration.
   - On Linux these will be `~/.config/powershell` for per-user configuration, or `/opt/microsoft/powershell/7` for system-wide configuration.

XenServer SDK for PowerShell 5.1:

1. Navigate to the extracted `XenServer-SDK\XenServerPowerShell\PowerShell_51` directory and copy the whole folder `XenServerPSModule` into your PowerShell modules directory.
   - On Windows this will normally be `$env:UserProfile\Documents\WindowsPowerShell\Modules` for per-user configuration, or `$env:windir\system32\WindowsPowerShell\v1.0\Modules` for system-wide configuration.

   For more information see PowerShell's documentation on module paths:

```
PS> Get-Help about_PSModulePath
```

2. Open a PowerShell 5.1 prompt as administrator.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

To do this, open the Windows **Start** menu by clicking the **Start** icon, find the item **PowerShell 5.1**, right click it and select **Run as administrator**.

3. Determine the current execution policy:

```
PS> Get-ExecutionPolicy
```

If the current policy is `Restricted`, you need to set it to `RemoteSigned`:

```
PS> Set-ExecutionPolicy RemoteSigned
```

You should understand the security implications of this change. If you are unsure, see PowerShell's documentation on execution policies:

```
PS> Get-Help about_Execution_Policies
```

If the current policy is `AllSigned`, you will be able to use the XenServer PowerShell module, but it will be inconvenient because this policy requires even scripts that you write on the local computer to be signed. You may want to change it to `RemoteSigned`, as above.

If the current policy is `RemoteSigned`, `ByPass`, or `Unrestricted` there is nothing to do.

4. Exit the privileged instance of PowerShell.
5. Open a PowerShell 5.1 prompt as a regular user (click **Start > PowerShell**) and import the XenServer PowerShell Module:

```
PS> Import-Module XenServerPSModule
```

6. If you wish to load specific environment settings when the XenServer PowerShell module is loaded, create the file `XenServerProfile.ps1` and put it in the folder containing your `$PROFILE` file for per-user configuration, or in `$PSHOME` for system-wide configuration.

   - On Windows these will normally be `$env:UserProfile\Documents\WindowsPowerShell` for per-user configuration, or `$env:windir\system32\WindowsPowerShell\v1.0` for system-wide configuration.

Getting help

For an overview of the XenServer PowerShell module, type:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Get-Help about_XenServer
```

You can obtain a list of all available cmdlets by running:

```
Get-Command -Module XenServerPSModule
```

For help with a specific command use:

```
Get-Help [CommandName]
```

See Using the XenServer PowerShell module for an overview of the module cmdlets and their structure, and a number of walk-throughs on how to perform certain specialized tasks.

Building and debugging the source code

Open the project `XenServerPowerShell.csproj` in Visual Studio 2022. You should now be ready to build the source code.

If in Debug mode, clicking **Start** will launch a PowerShell 7 prompt as an external process and import the compiled `XenServerPowerShell.dll` as a module (without, however, processing the scripts, types, and formats shipped within the XenServer PowerShell module). You should now be ready to debug the cmdlets.

Examples

Examples on the usage of the XenServer PowerShell module can be found at XenServer PowerShell Module usage examples on GitHub.

Using the XenServer® PowerShell module

This article describes how to use the XenServer PowerShell module to manage XenServer resource pools, starting with an overview of the module cmdlets and their structure.

Getting Started

For information about the system requirements for the PowerShell module and how to download and install it, see Getting Started - PowerShell.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XenServer PowerShell module overview

XenServer sessions

The first cmdlet you will need is `Connect-XenServer` to open a session to a server:

```
Connect-XenServer -Url https://<servername> -UserName user -Password pwd
```

It is possible to connect to multiple servers using the same credentials as well as open more than one sessions to the same server. All open sessions can be listed using the cmdlet `Get-XenSession`. If more than one sessions are to be used, you can set the most frequently used session as the default one as follows:

```
Connect-XenServer -Server srv -UserName usr -Password pwd
-SetDefaultSession
```

or

```
$XenServer_Default_Session = Get-XenSession -Server srv
```

All XenAPI calls are made in the context of a login session, so all cmdlets accept the parameter `-SessionOpaqueRef` which allows you to specify which of the open XenServer sessions to use:

```
Verb-Noun [-SessionOpaqueRef [<String>]]
```

This parameter is not necessary when only one open session exists or when a default session has been specified.

Once you have finished interacting with a server, it is good practice to log out using the cmdlet `Disconnect-XenServer`:

```
Get-XenSession | Disconnect-XenServer
```

Managing XenAPI objects

The cmdlets fall into the following categories:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

1. CLASS GETTERS

These retrieve a XenAPI object and have names such as `Get-XenT`, where `T` is an exposed XenAPI class. The object to get can be specified by `-Ref` or, for those that have a uuid or name, `-Name` or `-Uuid`. If no parameters are specified, all objects of this type are returned. Example:

```
Get-XenHost -Name "Demo Host"
```

2. CONSTRUCTORS

These create a new XenAPI object and have names such as `New-XenT`, where `T` is an exposed XenAPI class. Example:

```
$vm = Get-XenVM -Name "Demo VM"

New-XenVBD -VM $vm -VDI $null -Userdevice 3 -Bootable $false -Mode RO `

    -Type CD -Unpluggable $true -Empty $true -OtherConfig @{} `

    -QosAlgorithmType "" -QosAlgorithmParams @{}
```

3. CLASS REMOVERS

These destroy a XenAPI object and have names such as `Remove-XenT`, where `T` is an exposed XenAPI class. To specify the object to remove use the parameter `-T`, where `T` is the exposed XenAPI class, or `-Ref` or, for those objects that have a uuid or name, `-UUID` or `-Name`. Example:

```
Get-XenSR -Name "Demo SR" | Remove-XenSR
```

4. PROPERTY SETTERS

These set a field of a XenAPI object and have names such as `Set-XenT`, where `T` is an exposed XenAPI class. To specify the object use the parameter `-T`, where `T` is the exposed XenAPI class, or `-Ref` or, for those objects that have a uuid or name, `-UUID` or `-Name`. The field to set can be specified as an accordingly named parameter. Note that more than one fields at a time can be set in a synchronous call. Example:

```
Get-XenVM -Name "Demo VM" |`

    Set-XenVM -NameLabel "New name" -NameDescription "New description"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

5. PROPERTY ADDERS

These add an element to a field of a XenAPI object and have names such as `Add-XenT`, where `T` is an exposed XenAPI class. To specify the object use the parameter `-T`, where `T` is the exposed XenAPI class, or `-Ref` or, for those objects that have a uuid or name, `-UUID` or `-Name`. The field to which the element will be added can be specified as an accordingly named parameter. Note that elements can be added to more than one fields at a time in a synchronous call. Example:

```
Add-XenHost -Name "Demo Host" -Tags "Tag1"
```

6. PROPERTY REMOVERS

These remove an element from a field of a XenAPI object and have names such as `Remove-XenTProperty`, where `T` is an exposed XenAPI class. To specify the object use the parameter `-T`, where `T` is the exposed XenAPI class, or `-Ref` or, for those objects that have a uuid or name, `-UUID` or `-Name`. The field from which the element will be removed can be specified as an accordingly named parameter. Note that elements can be removed from more than one fields at a time in a synchronous call. Example:

```
Remove-XenHostProperty -Name "myHost" -Tags "tag1" -OtherConfig "myKey"
```

7. PROPERTY GETTERS

These retrieve the value of a field of a XenAPI object and have names such as `Get-XenTProperty`, where `T` is an exposed XenAPI class. To specify the object use the parameters `-T`, where `T` is the exposed XenAPI class, or `-Ref`. To specify the field use the enum parameter `-XenProperty`. Example:

```
Get-XenPIFProperty -Ref OpaqueRef:f433bf7b-2b0c-5f53-7018-7d195addb3ca `

    -XenProperty Network
```

8. INVOKERS

These invoke operations on XenAPI objects and have names such as `Invoke-XenT`, where `T` is an exposed XenAPI class. To specify the object use the parameter `-T`, where `T` is the exposed XenAPI class, or `-Ref` or, for those objects that have a uuid or name, `-UUID` or `-Name`. To specify the call to invoke, use the enum parameter `-XenAction`. Example:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Get-XenPBD -Uuid 1871ac51-ce6b-efc3-7fd0-28bc65aa39ff |`

    Invoke-XenPBD -XenAction Unplug
```

Most of the XenAPI calls can be run synchronously or asynchronously. To run a cmdlet asynchronously use the parameter `-Async` where available.

Note that, in the case of the setters, only one field can be set asynchronously at a time, and in the case of the adders and removers, elements can be added or removed asynchronously to only one field at a time.

Also, note that the cmdlets that are not explicit "getters" but return objects, do so only when the standard parameter `-PassThru` is specified. These cmdlets are `Connect-XenServer`, the constructors, adders, setters, certain invokers, the property removers, as well as all the asynchronous calls from any category. In the latter case, the cmdlet returns a `Task` object, the progress of which can be tracked by piping it into the cmdlet `Wait-XenTask`:

```
Invoke-XenVM -Name $vm_name -XenAction Start -Async -PassThru |`

    Wait-XenTask -ShowProgress
```

`Wait-XenTask`, too, can be used with the `-PassThru` parameter and, where applicable, it returns the opaque reference of the object that would be returned if the call were run synchronously.

Finally, as many of the cmdlets handle objects of type `XenRef<T>`, where `T` is an exposed API class, the cmldet `ConvertTo-XenRef` can be used to aid conversion between the two. Example:

```
Get-XenVM -Name "Demo VM" | ConvertTo-XenRef
```

Complete walk-throughs

The following sections include a number of walk-throughs on how to perform certain specialized tasks.

- Update your XenServer hosts
- Copy or live migrate a VM across pools
- Configure vGPU and GPU pass-through

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Update your XenServer® hosts by using the XenServer PowerShell module

Overview

Complete the following steps to update your XenServer pools:

1. Install or upgrade to XenServer 8.4
2. Configure updates for your pool
3. Obtain updates for your pool
4. View available updates for your pool
5. Understand the guidance categories and update tasks
6. Apply updates to your pool

> **NOTE:**
>
> Applying updates is only assured to work when the previous state of the pool is less than six months old. If you haven't applied updates for more than six months, updating to the latest update level may not work.

Configure updates for your pool

Updates from the Content Delivery Network

To receive updates from the Content Delivery Network (CDN), your XenServer hosts require internet access. If your hosts are behind a firewall, ensure that they have access to subdomains of `ops.xenserver.com`. For more information, see Connectivity requirements.

Before being able to apply updates to your XenServer hosts and pools, you must subscribe your hosts or pools to the **Early Access** or **Normal** update channel. These channels control how soon you can access updates that are made available in the CDN. Follow these steps to configure updates:

1. Create and enable the Early Access update channel for your pool:

```
$baseBinUrl = "https://repo.ops.xenserver.com/xs8/base"

$baseSrcUrl = "https://repo-src.ops.xenserver.com/xs8/base"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
$baseRepo = Invoke-XenRepository -Name "dummy" -XenAction Introduce`

    -BinaryUrl $baseBinUrl -SourceUrl $baseSrcUrl -Update $false `

    -NameLabel "base_repo" -NameDescription "Base" -GpgkeyPath ""
-PassThru



$eaBinUrl = "https://repo.ops.xenserver.com/xs8/earlyaccess"

$eaSrcUrl = "https://repo-src.ops.xenserver.com/xs8/earlyaccess"



$updateRepo = Invoke-XenRepository -Name "dummy" -XenAction Introduce`

    -BinaryUrl $eaBinUrl -SourceUrl $eaSrcUrl -Update $true `

    -NameLabel "early_access_repo" -NameDescription "Early Access"
-GpgkeyPath "" -PassThru



Get-XenPool | Add-XenPool -Repository $updateRepo.opaque_ref
```

Alternatively, create and enable the Normal update channel for your pool:

```
$baseBinUrl = "https://repo.ops.xenserver.com/xs8/base"

$baseSrcUrl = "https://repo-src.ops.xenserver.com/xs8/base"



$baseRepo = Invoke-XenRepository -Name "dummy" -XenAction Introduce`

    -BinaryUrl $baseBinUrl -SourceUrl $baseSrcUrl -Update $false `

    -NameLabel "base_repo" -NameDescription "Base" -GpgkeyPath ""
-PassThru
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
$normBinUrl = "https://repo.ops.xenserver.com/xs8/normal"

$normSrcUrl = "https://repo-src.ops.xenserver.com/xs8/normal"



$updateRepo = Invoke-XenRepository -Name "dummy" -XenAction Introduce`

    -BinaryUrl $normBinUrl -SourceUrl $normSrcUrl -Update $true `

    -NameLabel "normal_repo" -NameDescription "Normal" -GpgkeyPath ""
-PassThru



Get-XenPool | Add-XenPool -Repository $updateRepo.opaque_ref
```

NOTE:

If you want to switch from the Early Access to the Normal channel, or vice versa, you need to disable the previous update channel before enabling the new one:

```
Get-XenPool | Remove-XenPoolProperty -Repository $updateRepo.opaque_ref
```

2. Retrieve a list of the currently enabled repositories:

```
Get-XenPool | select -ExpandProperty repositories
```

3. View more details about the currently enabled repositories:

```
Get-XenPool | select -ExpandProperty repositories | Get-XenRepository
```

4. Optionally, you can configure and enable a proxy server to use for communication between your server and the CDN that hosts the repositories:

```
Get-XenPool | Invoke-XenPool -XenAction ConfigureRepositoryProxy -Url
"http://proxy.example.com" `
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    -Username $proxyUser -Password $proxyPassword
```

View the proxy server configuration:

```
Get-XenPool | select repository_proxy_url,repository_proxy_username
```

Disable the proxy server configuration:

```
Get-XenPool | Invoke-XenPool -XenAction DisableRepositoryProxy
```

Offline updates using update bundles

Before being able to apply offline updates, you must configure your host or pool to use the **Offline** update channel. Follow these steps:

1. Create the Offline update channel for your pool:

```
$offlineRepo = Invoke-XenRepository -Name "dummy" -XenAction
IntroduceBundle `

    -NameLabel "offline_repo" -NameDescription "Offline" -PassThru
```

This cmdlet creates a bundle repository that acts as the Offline update channel and stores the update bundle for offline application.

2. Enable the Offline update channel for your pool:

```
Get-XenPool | Add-XenPool -Repository $offlineRepo.opaque_ref
```

This cmdlet enables the bundle repository as the Offline update channel for your pool.

NOTE:

If you want to switch from one of the CDN channels (Early Access or Normal) to the Offline channel, or vice versa, you need to disable the previous channel before enabling the new one:

```
Get-XenPool | Remove-XenPoolProperty -Repository $repoToDisable.opaque_ref
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Obtain updates for your pool

Synchronize updates for your pool from CDN

Enable your pool to automatically synchronize with the update channel by configuring a synchronization schedule. You can schedule a synchronization to take place daily or weekly on a certain day of the week. Synchronizing your pool with the update channel downloads all available updates to the pool coordinator and you can then apply them to your pool.

1. Set your pool to synchronize daily:

```
$pool = Get-XenPool

Invoke-XenPool -Pool $pool -XenAction ConfigureUpdateSync
-UpdateSyncFrequency daily

Set-XenPool -Pool $pool -UpdateSyncEnabled $true
```

   Alternatively, set your pool to synchronize weekly:

```
$pool = Get-XenPool

Invoke-XenPool -Pool $pool -XenAction ConfigureUpdateSync
-UpdateSyncFrequency weekly -UpdateSyncDay 0 # (0 is Sunday, 1 is
Monday, etc.)

Set-XenPool -Pool $pool -UpdateSyncEnabled $true
```

2. View your synchronization configuration:

```
Get-XenPool | select
update_sync_frequency,update_sync_day,update_sync_enabled
```

3. Get the timestamp of your pool's last successful synchronization with the update channel:

```
Get-XenPool | select last_update_sync
```

Alternatively, you can manually synchronize your XenServer pool with the update channel:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
$updateChecksum = Invoke-XenPool -Pool $pool -XenAction SyncUpdates
-PassThru
```

`$updateChecksum` is a unique identifier that indicates the level of the updates installed. It changes whenever new updates are made available in the public CDN and is later used when applying updates to your pool to ensure that you are always applying the latest available updates. `$updateChecksum` can also provide useful information if you need to contact Technical Support.

If you have configured a synchronization schedule, you can retrieve the update checksum at any time by querying the update channel used for the synchronization:

```
Get-XenPool | select -ExpandProperty repositories | Get-XenRepository |
select -ExpandProperty hash
```

NOTE:

After synchronizing, apply the updates to your pool as soon as possible to benefit from the latest updates.

If you designate a new pool coordinator after synchronizing but before applying updates to the hosts in the pool, you must synchronize again with the new pool coordinator before you can update the pool.
Do not synchronize your XenServer pool while the pool is in the process of being updated.

Upload an update bundle to your pool

Upload an update bundle file:

```
$coordinator = Get-XenPool | select -ExpandProperty master | Get-XenHost

$task = New-XenTask -PassThru -Label "MyBundleUploadTask"

Send-XenBundle -XenHost $coordinator.address -Path $xsbundleFile -TaskRef
$task.opaque_ref

$task | Wait-XenTask -ShowProgress
```

You can view the checksum of the uploaded bundle at any time by querying the Offline channel using the following cmdlet:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Get-XenPool | select -ExpandProperty repositories | Get-XenRepository |
select -ExpandProperty hash
```

View available updates for your pool

Before installing updates, view the available updates for your pool and review the update tasks required. For more information about the different update tasks that might be required for an update, see Understand the guidance categories and update tasks.

Quick check for available updates for a particular host

Check if there are any available updates for a particular host:

```
Get-XenHost -Uuid $hostUuid | select latest_synced_updates_applied
```

This cmdlet returns `yes` if there are updates available for a particular host and `no` if there are none available. See View available updates and required update tasks for all hosts for more information on how to view details on the available updates.

View available updates and required update tasks for all hosts

Check for available updates for all hosts by sending a GET request to the pool coordinator's HTTP(S) endpoint `/updates`. This is achieved using the following cmdlets:

```
$jsonPath = $env:TEMP + [System.IO.Path]::GetRandomFileName()

$coordinator = Get-XenPool | select -ExpandProperty master | Get-XenHost

$task = New-XenTask -PassThru -Label "MyReceiveTask"

Receive-XenUpdates -XenHost $coordinator.address -Path $jsonPath -TaskRef
$task.opaque_ref

$task | Wait-XenTask -ShowProgress

$jsonOutput = Get-Content -Raw -Path $jsonPath | ConvertFrom-Json
```

The output returned is in JSON format and contains the following objects:

- `hosts`: Lists the available updates for individual hosts.
- `updates`: Lists the details of the available updates.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `hash`: The `$updateChecksum` (used to ensure that you are always applying the latest available updates).

The `hosts` and `updates` objects also contain the `guidance` object which consists of the following keys:

- `mandatory`
- `recommended`
- `full`
- `livepatch`

These keys refer to the different guidance categories for update tasks. They list the update tasks required for your hosts and VMs. For more information about the different tasks that might be required for an update, see Understand the guidance categories and update tasks.

Understand the guidance categories and update tasks

Some tasks (such as evacuating or rebooting your hosts) might be required before and after applying updates to your pool. Sometimes, there are no update tasks required.

Guidance categories

XenServer tries to minimize the disruption to your VMs that these tasks might cause by categorizing the tasks into **Mandatory**, **Recommended**, **Full-effectiveness**, and **Live patch**. These categorizations enable you to judge whether an update task that might cause downtime or minor disruption for your hosts or VMs is necessary for your environment and risk profile.

Updates can have tasks listed in more than one of these categories. For example, an update might require that you restart the host to get the full-effectiveness of the update, but recommend restarting the toolstack to get most of the benefit of the update with less potential disruption to the pool.

During the update process, you can choose to carry out one of the following three levels of tasks:

1. Mandatory
2. Mandatory + Recommended
3. Mandatory + Recommended + Full-effectiveness

© 1997–2026 Citrix Systems, Inc. All rights reserved.

MANDATORY

Mandatory tasks *must* be performed after an update otherwise the system might fail at runtime. These actions are required to enable critical fixes and ensure that your environment is secure and stable. You cannot opt out of mandatory tasks.

RECOMMENDED

Recommended tasks are the tasks that we recommend performing to get the benefit of the majority of features and fixes delivered in the updates. If you choose not to perform these tasks now, they are listed in the pending update tasks for the applicable pool, host, or VM.

Why perform the recommended tasks:

- These tasks are the ones that ensure a secure, stable XenServer environment.

Why opt out of the recommended tasks:

- After reviewing the detailed information for the updates, you judge that the risk of not fully applying these updates now is acceptable.
- The recommended tasks cause unwanted disruption to your VMs now.

If you have opted to defer the recommended tasks, plan to complete these tasks during a suitable maintenance window to maintain the stability of your environment.

FULL-EFFECTIVENESS

Full effectiveness tasks are required to get the benefit of the related update. The updates that have full-effectiveness tasks associated with them are usually relevant only to users on certain hardware or using specific features.

Review the update information to understand whether these tasks are required for your environment. If you choose not to perform these tasks now, they are listed in the pending tasks for the applicable pool, host, or VM.

Why perform the full-effectiveness tasks:

- The updates that have full-effectiveness tasks are relevant to your hardware, environment, or configuration.

Why opt out of the full-effectiveness tasks:

- The updates that have full-effectiveness guidance are not relevant to your hardware, environment, or configuration.
- The full-effectiveness tasks cause unwanted disruption to your VMs now.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- You do not need the benefits of these updates right now.

If the full-effectiveness tasks apply to your environment, but you have opted to defer them, plan to complete these tasks during a suitable maintenance window to maintain the stability of your environment.

LIVE PATCHES

Updates to certain components can include a live patch. A live patch enables updates to be applied to core components without requiring a host reboot.

Whether a live patch can be applied to your hosts depends on the version of the component that was installed when the hosts were last rebooted. If an update can be applied as a live patch to your hosts, the live patch guidance replaces the recommended guidance.

EXAMPLE:

You have two pools. Pool A is updated to a recent level. Pool B has not been updated for some time. We release a new update that has the recommended update task "Restart host" and the live patch update task "Restart toolstack".

In pool A, the live patch can be applied to these more up-to-date hosts. The recommended guidance shows "Restart toolstack". The less disruptive task from the live patch guidance overrides the recommended guidance.
In pool B, the live patch cannot be applied to the hosts as they are at an older level. The recommended guidance shows "Restart host". The recommended guidance remains applicable. The live patch guidance is irrelevant in this case.
Sometimes only some of the fixes in an update are enabled when the update is applied as a live patch. In this situation, you might still see a recommended reboot for an update even though the live patch applies. Review the update details to understand whether you need all fixes in the update or just those fixes enabled by the live patch. You can then use this information to choose whether to perform the tasks. For more information, see View available updates for your pool.

Update tasks

One or more of the following tasks might be required when applying an update. Any type of update task can be listed in any guidance category.

UPDATE TASKS FOR YOUR HOSTS

You only ever carry this task out *before* applying updates and you sometimes carry it out as part of the 'Reboot host' task:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATE TASK	CMDLET TO CARRY OUT TASK	DESCRIPTION
Evacuate host	`Invoke-XenHost -XenAction Evacuate`	Migrate all VMs off the XenServer host or shut them down. While this task is in progress, the XenServer pool is operating at reduced capacity as one host is temporarily unavailable to run VMs.

The following tasks require actions on the updated host:

UPDATE TASK	CMDLET TO CARRY OUT TASK	DESCRIPTION
Reboot host	`Invoke-XenHost -XenAction Reboot`	The XenServer host must be restarted. Any VMs are migrated off the host and the host is restarted. While this task is in progress, the XenServer pool is operating at reduced capacity as one host is temporarily unavailable to run VMs.
Reboot host on Xen live patch failure	`Invoke-XenHost -XenAction Reboot`	Applying a Xen live patch failed. The XenServer host must be restarted to get the update to take effect. Any VMs are migrated off the host and the host is restarted. While this task is in progress, the XenServer pool is operating at reduced capacity as one host is temporarily unavailable to run VMs.
Reboot host on kernel live patch failure	`Invoke-XenHost -XenAction Reboot`	Applying a dom0 kernel live patch failed. The XenServer host must be restarted to get the update to take effect. Any

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATE TASK	CMDLET TO CARRY OUT TASK	DESCRIPTION
		VMs are migrated off the host and the host is restarted. While this task is in progress, the XenServer pool is operating at reduced capacity as one host is temporarily unavailable to run VMs.
Restart toolstack	`Invoke-XenHost -XenAction RestartAgent`	The toolstack on the host must be restarted. Restart the toolstack on the updated host instead of the pool coordinator. When the toolstack is restarted on the pool coordinator, the connection to the pool is lost but when the toolstack is restarted on other pool members, there is no visible effect.

View the update tasks required for your hosts

See View available updates for your pool for information on how to review update tasks required for your hosts prior to applying updates to your pool.

After applying updates, view the mandatory, recommended, and full-effectiveness tasks required for your hosts by using the following cmdlet:

```
Get-XenHost -Uuid $hostUuid | `

    select
pending_guidances,pending_guidances_recommended,pending_guidances_full
```

UPDATE TASKS FOR YOUR VMS

Some updates provide new features for your VMs. These updates might require the following tasks on your VMs:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATE TASK	CMDLET TO CARRY OUT TASK	DESCRIPTION
Restart VM	`Invoke-XenVM -XenAction CleanReboot`	The VM must be restarted. While the VM is restarting, the VM is unavailable to the end user.
Restart device model	`Invoke-XenVM -XenAction RestartDeviceModels`	The device model for VMs on the updated host must be restarted. While the device model is restarting, you can't stop, start, or migrate the VM. The end user of the VM might see a slight pause and resume in their session. For the restart device model action to be supported on a Windows VM, the VM must have the XenServer VM Tools for Windows installed.

View the update tasks required for your VMs

See View available updates for your pool for information on how to review update tasks required for your VMs prior to applying available updates to your pool.

After applying updates, view the mandatory, recommended, and full-effectiveness tasks required for your VMs by using the following cmdlet:

```
Get-XenVM -Uuid $vmUuid | `

    select
pending_guidances,pending_guidances_recommended,pending_guidances_full
```

Apply updates to your pool

Before you start

- Ensure that all the hosts in your pool are online before carrying out the pool update.
- Ensure that there are no pending mandatory update tasks on any host or VM. Any mandatory update tasks pending from previous updates *must* be carried out before starting a new pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

update. For more information, see View pending update tasks required for your hosts and View the pending update tasks required for your VMs.

- Disable High Availability (HA) if it is enabled:

```
Get-XenPool | Invoke-XenPool -XenAction DisableHa
```

- Disable Workload Balancing (WLB) if it is enabled:

```
Get-XenPool | Set-XenPool -WlbEnabled $false
```

Install updates

To perform an update to your pool, you must apply updates on every host in the pool, starting with the pool coordinator first. Follow these steps, starting with the pool coordinator:

1. Disable the host:

```
Invoke-XenHost -Ref $hostRef -XenAction Disable
```

2. If one of the update tasks required for the update is 'Evacuate host' or 'Reboot host', evacuate the host:

```
Invoke-XenHost -Ref $hostRef -XenAction Evacuate
```

   If you can't migrate any VMs to other hosts during the host evacuation, shut down or suspend the VMs.

3. Apply updates to the host:

```
Invoke-XenHost -Ref $hostRef -XenAction ApplyUpdates `

    -Hash $updateChecksum -Async -PassThru | Wait-XenTask
```

View and carry out the required update tasks

1. Get a list of the host update tasks required. For more information, see Update tasks for your host.

   Carry out the host's update tasks in the list in the following order:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

a. Restart toolstack (can be skipped if there is a 'Reboot host' to be carried out)

b. Reboot host

2. For every running VM on the host, get a list of the VM update tasks required. For more information, see Update tasks for your VM.

   Carry out the VM post-update tasks in the list in the following order:

   a. Restart device model (can be skipped if there is a 'Restart VM' to be carried out)

   b. Restart VM

3. Enable the host if it is still in a disabled state:

```
Invoke-XenHost -Ref $hostRef -XenAction Enable
```

4. For every VM which you migrated to another host using `Invoke-XenHost -XenAction Evacuate` before the host update, get a list of the update tasks. For more information, see Update tasks for your VM.

   If 'Restart VM' is in the list of update tasks, shut down the VM and start it on the current updated host. Otherwise, migrate the VM back to the current updated host.

5. Resume or start the VMs that you shut down or suspended before you applied updates.

6. View the host update status:

```
Get-XenHost | select
last_software_update,latest_synced_updates_applied,last_update_hash
```

Repeat the steps above to update every host in your pool.

After updating your hosts

After updating each host in your pool, carry out any remaining update tasks.

1. For every VM in your pool, get a list of the update tasks. For more information, see Update tasks for your VM.

   If 'Restart VM' is in the list of update tasks, carry it out.

2. Enable HA if you disabled it before applying updates:

```
Get-XenPool | Invoke-XenPool -XenAction EnableHa
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

3. Enable WLB if you disabled it before applying updates:

```
Get-XenPool | Set-XenPool -WlbEnabled $true
```

4. If you chose to carry out only the mandatory update tasks required for a pool update, the update tasks that have not been carried out are appended to the list of pending update tasks required for your hosts. To view this list and carry out these tasks, see Update tasks for your host.

Copy or live migrate a VM across pools

The following tutorial shows how to use the XenServer® PS module to migrate a running VM or copy a halted VM from a pool or standalone server, hereafter referred to as the source server, to a different pool or standalone server, hereafter referred to as the destination server.

This example also displays how to work with multiple sessions and use implicitly the global variable `$XenServer_Default_Session`.

We're starting by connecting to the source server. The session created with the source server will be more frequently used, so it makes sense to set it as the default session:

```
Connect-XenServer -Url $sourceServerUrl -UserName $sourceServerUsername
-Password $sourceServerPwd -SetDefaultSession
```

Let's select the VM we are going to copy or live migrate from the source pool:

```
$theVM = Get-XenVM -Name $nameOfSourceVmToCopy
```

Note that in the above call we did not need to specify the `-Session` parameter since we have set the source session as the default one.

Then we need to connect to the destination pool and obtain its coordinator. Note that we will need to specify the `-Session` parameter in all calls made to the destination pool since this session is not the default one.

```
$destSession = Connect-XenServer -Url $destServerUrl -UserName
$destServerUsername -Password $destServerPwd -PassThru
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
$destCoordinator = Get-XenPool -Session $destSession.opaque_ref | `

    select -ExpandProperty master | `

    Get-XenHost -Session $destSession.opaque_ref
```

The coordinator of the destination pool has to be prepared to receive the VM that will be live migrated or copied. For this, we will also need to obtain the details of the network on the destination pool through which migration traffic will be received.

```
$destTransferNetwork = Get-XenNetwork -Name $destTransferNetworkName
-Session $destSession.opaque_ref



$destMap = Invoke-XenHost -XenAction MigrateReceive -XenHost
$destCoordinator -Network $destTransferNetwork -Session
$destSession.opaque_ref -options @null -PassThru
```

The next step is to map the VM's VIFs to networks on the destination pool. Let's consider a case where the destination pool has as many networks as the VIFs on the VM we want to live migrate or copy, so we can create a one-to-one mapping of VIFs to networks as follows:

```
$targetNetworkRefs = Get-XenNetwork -SessionOpaqueRef
$destSession.opaque_ref | ConvertTo-XenRef



$vmVifRefs = $theVM.VIFs | Get-XenVIF | ConvertTo-XenRef



$vifmap = @{}



for ($i = 0; $i -lt $vmVifRefs.Count; $i++) {

    $vifmap.Add($vmVifRefs[$i], $targetNetworkRefs[$i])
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
}
```

Similarly, we proceed to map the VM's disks to storage repositories on the destination pool. Let's consider a simple case where we want to place all VM disks on the same storage repository with name `$destSrName`.

```
$targetSrRef = Get-XenSR -name $destSrName -SessionOpaqueRef
$destSession.opaque_ref | ConvertTo-XenRef



$vdiRefs = $theVM.VBDs | Get-XenVBD | `

    where { $_.type -eq [XenAPI.vbd_type]::Disk} | `

    select -ExpandProperty VDI



$vdimap = @{}



foreach ($vdiRef in $vdiRefs) {

    $vdimap.Add($vdiRef, $targetSrRef)

}
```

Now we are ready to live migrate or copy the VM to the destination pool. Both operations are performed using the same cmdlet. However, we need to specify different options in each case. The call is run on the source pool session. Due to the long duration of the operation, it is recommended to run it asynchronously.

To live migrate a running VM:

```
$theTask = Invoke-XenVM -VM $theVM -XenAction MigrateSend -Live $true `

    -Dest $destMap -VifMap $vifmap -VdiMap $vdimap `
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    -Options @{} -Verbose -Async -PassThru
```

Or, to copy a halted VM:

```
$theTask = Invoke-XenVM -VM $theVM -XenAction MigrateSend -Live $false `
    -Dest $destMap -VifMap $vifmap -VdiMap $vdimap `
    -Options @{"copy"="true"} -Verbose -Async -PassThru
```

In either case, we can monitor the task progress:

```
Wait-XenTask -Task $theTask -ShowProgress
```

Once the task is finished, we can query it to obtain the new VM on the destination pool:

```
$theTask = Get-XenTask -uuid $theTask.uuid



if ($theTask.status -eq [XenAPI.task_status_type]::success) {
    $doc = [xml]$theTask.result
    $newVmRef = $doc.value
    Get-XenVM -Ref $newVmRef -SessionOpaqueRef $destSession.opaque_ref
}
else {
    Write-Host "Operation failed: " $theTask.error_info
}
```

Finally, we need to disconnect from the source and the destination pools:

```
Get-XenSession | Disconnect-XenServer
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Configure vGPU and GPU pass-through

The following example is a tutorial of a typical vGPU configuration use case using the XenServer® PowerShell SDK cmdlets. The example output provided is from a server with both a NVIDIA Grid K1 and K2 card installed.

We start by connecting to the server:

```
Connect-XenServer -Server server -UserName username -Password password
```

The first thing we probably want to check is which GPU groups exist (these have been created automatically once the graphics hardware is installed):

```
PS> Get-XenGPUGroup | select name_label, GPU_types, allocation_algorithm



name_label                                          GPU_types
allocation_algorithm

----------                                          ---------
--------------------

Group of NVIDIA Corporation GK104GL [GRID K2] GPUs   {10de/11bf}
 depth_first

Group of NVIDIA Corporation GK107GL [GRID K1] GPUs   {10de/0ff2}
 breadth_first
```

The `allocation_algorithm` is the placement policy for assigning VMs to GPUs in order to achieve either maximum density by placing as many VMs as possible on the same GPU (`depth_first`), or maximum performance by placing VMs on as many GPUs as possible (`breadth_first`). For example, the following command sets the `allocation_algorithm` to achieve maximum performance:

```
PS> Get-XenGPUGroup | Set-XenGPUGroup -AllocationAlgorithm breadth_first

PS> Get-XenGPUGroup | select name_label, GPU_types, allocation_algorithm
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
name_label                                              GPU_types
 allocation_algorithm

----------                                              ---------
 --------------------

Group of NVIDIA Corporation GK104GL [GRID K2] GPUs   {10de/11bf}
breadth_first

Group of NVIDIA Corporation GK107GL [GRID K1] GPUs   {10de/0ff2}
breadth_first
```

Now we can list some information about the vGPU types. The vGPU types are pre-sets which can be used to create different kinds of vGPUs.

```
PS> Get-XenVGPUType | ft vendor_name, model_name, framebuffer_size,
max_heads, `

    max_resolution_x, max_resolution_y



vendor_name           model_name   framebuffer_size    max_heads
max_resolution_x   max_resolution_y

-----------           ----------   ----------------    ---------
----------------   ----------------

NVIDIA Corporation    GRID K100           268435456            2
    1920               1200

NVIDIA Corporation    GRID K140Q         1006632960            2
    2560               1600

NVIDIA Corporation    GRID K240Q         1006632960            2
    2560               1600

NVIDIA Corporation    GRID K260Q         2013265920            4
    2560               1600

NVIDIA Corporation    GRID K200           268435456            2
    1920               1200
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
                    passthrough                 0           0
       0                0
```

The vGPU type `passthrough` is supported for all PCI display devices, and can be used to create pass-through vGPUs.

We can see for which PCI display devices a vGPU type is supported as follows:

```
PS> Get-XenVGPUType | Where {$_.model_name -eq "GRID K100"} |`

    Get-XenVGPUTypeProperty -XenProperty SupportedOnPGPUs |`

    Get-XenPGPU | Get-XenPCI -Ref {$_.PCI} | select device_name, pci_id



device_name          pci_id

-----------          ------

GK107GL [GRID K1]    0000:0a:00.0

GK107GL [GRID K1]    0000:07:00.0

GK107GL [GRID K1]    0000:08:00.0

GK107GL [GRID K1]    0000:09:00.0
```

A vGPU type will show up as supported or enabled in a GPU group if it is supported or enabled respectively on at least one of the pGPUs in the group. We can query the GPU groups to find out which vGPU types are supported or enabled. For example:

```
PS> Get-XenGPUGroup | Where {$_.name_label -match "GRID K2"} |`

    Get-XenGPUGroupProperty -XenProperty EnabledVGPUTypes |`

    Get-XenVGPUType | select model_name



model_name
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
----------

GRID K200

passthrough

GRID K260Q

GRID K240Q
```

We may want to disallow a certain vGPU type on a pGPU. For example, the following commands disable and then re-enable the vGPU type Grid K240Q:

```
PS> $vgpuType = Get-XenVgpuType | where {$_.model_name -eq "GRID K240Q"}

PS> $vgpuType | Get-XenVGPUTypeProperty -XenProperty EnabledOnPGPUs |`

    Get-XenPGPU | select uuid



uuid

----

a913a90d-0be9-aea3-862f-3fbfe1823a3f

8a0d1316-0f09-1ee9-f95b-c778c759ee40



PS> Remove-XenPGPUProperty -Uuid a913a90d-0be9-aea3-862f-3fbfe1823a3f `

    -EnabledVGPUTypes $vgputype.opaque_ref



PS> $vgpuType | Get-XenVGPUTypeProperty -XenProperty EnabledOnPGPUs |get-
xenpgpu | select uuid
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
uuid

----

8a0d1316-0f09-1ee9-f95b-c778c759ee40



PS> Add-XenPGPU -Uuid a913a90d-0be9-aea3-862f-3fbfe1823a3f
-EnabledVGPUTypes $vgputype.opaque_ref

PS> $vgpuType | Get-XenVGPUTypeProperty -XenProperty EnabledOnPGPUs |get-
xenpgpu | select uuid



uuid

----

a913a90d-0be9-aea3-862f-3fbfe1823a3f

8a0d1316-0f09-1ee9-f95b-c778c759ee40
```

We may want to find out how many vGPUs of a certain type can be started on the pGPUs in a group, in addition to the vGPUs which are already running:

```
PS> $gpuGroups = Get-XenGPUGroup

PS> Get-XenVGPUtype | % {Get-XenGPUGroupProperty $gpuGroups[0]
-XenProperty RemainingCapacity -VgpuType $_}



0

0

8

4
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
16

2
```

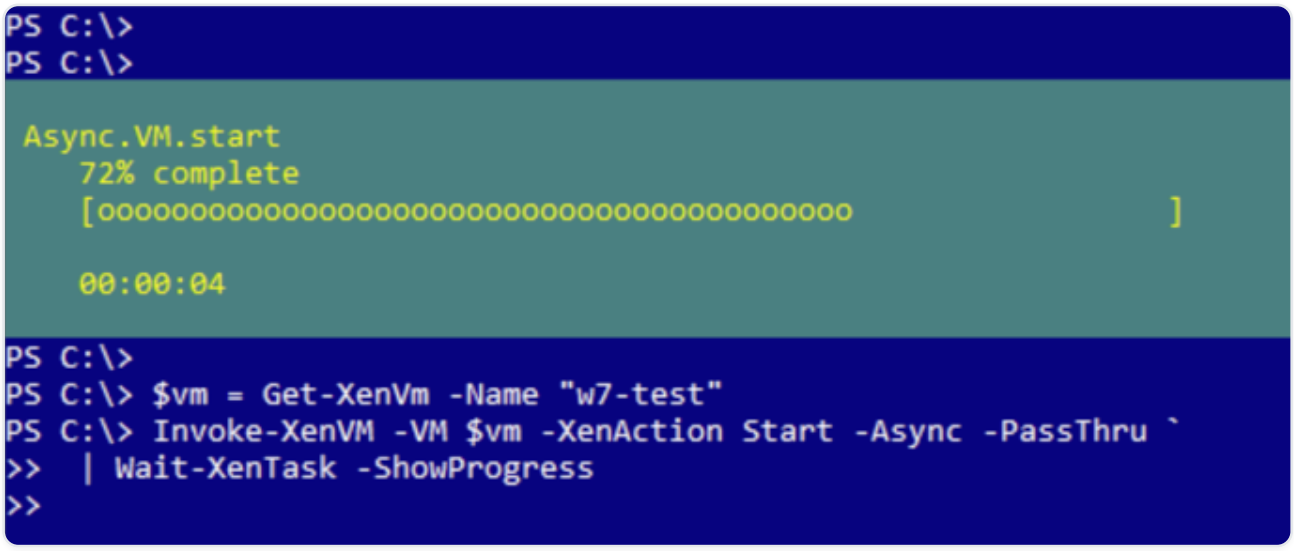
Now suppose we want to assign a vGPU of type Grid K260Q to a VM which has no vGPU yet. This can be done as follows:

```
PS> $vm = Get-XenVm -Name "w7-test"

PS> $vgpuTypes = Get-XenVGPUtype

PS> Get-XenVMProperty -VM $vm -XenProperty VGPUs

PS>

PS> New-XenVGPU -VM $vm -GPUGroup $gpuGroups[0] -Device 0 -Type
$vgpuTypes[3] -PassThru



uuid               : f1122947-3b11-3fd3-0630-779701b37265

VM                 : XenAPI.XenRef`1[XenAPI.VM]

GPU_group          : XenAPI.XenRef`1[XenAPI.GPU_group]

device             : 0

currently_attached : False

other_config       : {}

type               : XenAPI.XenRef`1[XenAPI.VGPU_type]

resident_on        : XenAPI.XenRef`1[XenAPI.PGPU]

opaque_ref         : OpaqueRef:15b2d1a9-8944-2f28-53df-6b8274d4d6fb

Changed            : True
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

At this stage we can boot the VM, install the in-guest NVIDIA drivers, and enable RDP. Then we may want to disable the VNC console as this has a significant performance overhead. To do this we need to shut down the VM, set the flag `vgpu_vnc_enabled` to `false` and then boot the VM.

```
PS> Invoke-XenVM -VM $vm -XenAction Shutdown –Async –PassThru | Wait-
XenTask –ShowProgress



PS> $p = Get-XenVMProperty -VM $vm -XenProperty Platform

PS> $p["vgpu_vnc_enabled"]="false"

PS> Set-XenVM -VM $vm -Platform $p

PS> Invoke-XenVM -VM $vm -XenAction Start –Async –PassThru | Wait-XenTask
–ShowProgress
```



Before finishing, we should remember to disconnect from the server:

```
Get-XenSession | Disconnect-XenServer
```

PowerShell examples

PowerShell examples

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Getting Started with C

The `XenServer-SDK` directory contains the following folders that are relevant to C# programmers:

- `XenServer.NET`: The XenServer SDK for C#.NET.
  - `XenServer.NET.x.y.z.nupkg`: The compiled library shipped as a NuGet package.
  - `src`: XenServer.NET source code shipped as a Microsoft Visual Studio project. Every API object is associated with one C# file. For example, the functions implementing the VM operations are contained within the file `VM.cs`.

RPC protocol

The C# SDK supports the JSON-RPC v2.0 protocol.

Platform supported

- Windows
- Linux

The compiled library targets .NET Framework 4.5, .NET Standard 2.0, and .NET 8.0.

Library

The library is generated as a Dynamic Link Library `XenServer.dll` that C# programs can reference.

The C# SDK is backwards compatible and can be used to interact with hosts running all versions of XenServer or Citrix Hypervisor (from XenServer 7.3 and Citrix Hypervisor 8.0 onwards).

Dependencies

Newtonsoft JSON.NET by James Newton-King.

Installation

To use the NuGet package in your code, add its location as a NuGet source:

```
nuget sources Add -Name "Offline Package" -Source "/full/path/to/package/
directory"
```

Then install the package:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
nuget install XenServer.NET
```

Building the source code

Open the project `XenServer.csproj` in Visual Studio 2022. You should now be ready to build the source code.

Examples

Examples on the usage of the C# SDK can be found at XenServer.NET usage examples on GitHub.

C# examples

C# examples

Getting Started with Python

The `XenServer-SDK` directory contains the following folders that are relevant to Python developers:

- `XenServerPython`: This directory contains the XenServer Python module *XenAPI.py*.

Python support:

- The XenServer 8.4 SDK (version **24.32.0** and later) supports only Python 3.

Alternative installation

The Python module is also available as a package on PyPi. To install the package, enable the virtual environment where it will be used and run

```
pip install XenAPI
```

RPC protocol

The Python module supports the XML-RPC protocol.

Platform supported

- Linux

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Windows

Library

- XenAPI.py

Dependencies

- xmlrpclib

Examples

Examples on the usage of the XenServer Python module can be found at XenAPI.py usage examples on GitHub.

Python examples

Python examples

Getting Started with Java

The `XenServer-SDK` directory contains the following folders that are relevant to Java programmers:

- `XenServerJava`: The XenServer SDK for Java
  - `xen-api-x.y.z.jar`: Java archive file containing the compiled library.
  - `xen-api-x.y.z-javadoc.jar`: Java archive file containing the documentation.
  - `xen-api-x.y.z-sources.jar`: Java archive file containing the source code as a Maven project. Every API object is associated with one Java file. For example the functions implementing the VM operations are contained within the file `VM.java`.

RPC protocol

The Java SDK supports the JSON-RPC v2.0 protocol.

Platform supported

- Linux
- Windows

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Library

- The library is a Java archive file `xenserver.jar` that is linked by Java programs.

> **NOTE:**
>
> This library is generally, but not fully, backwards compatible. To interact with hosts running older versions of XenServer or Citrix Hypervisor™, it is advisable to use the library of the same version as the host.

Dependencies

XenServerJava is dependent upon the JSON library by the Jackson project and the Apache HttpClient by the Apache Software Foundation.

Examples

Examples on the usage of the Java SDK can be found at XenServerJava usage examples on GitHub.

Java examples

Java examples

Getting Started with C

> **NOTE:**
>
> C support in the XenServer SDK is deprecated and will be removed in a future release.

The `XenServer-SDK` directory contains the following folders that are relevant to C programmers:

- `libxenserver`: The XenServer SDK for C.
  - `bin`: The libxenserver compiled library.
  - `src`: The libxenserver source code and a Makefile to build it. Every API object is associated with a header file, which contains declarations for all the object's API functions. For example, the type definitions and functions required to invoke VM operations are all contained in `xen_vm.h`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC protocol

The C SDK supports the XML-RPC protocol.

Platform supported

- Linux
- Windows (under cygwin)

Library

The library is generated as `libxenserver.so` that is linked by C programs.

> **NOTE:**
>
> This library is not backwards compatible. To interact with hosts running older versions of XenServer or Citrix Hypervisor™, it is advisable to use the library of the same version as the host.

Dependencies

The library is dependent upon the XML toolkit from the GNOME project, by Daniel Veillard, et al. It is packaged as `libxml2-devel` on CentOS and `libxml2-dev` on Debian.

Building the source code

To compile the source code type `make` in the `libxenserver/src` directory.

Examples

Examples on the usage of the C SDK can be found at libxenserver usage examples on GitHub.

C examples

C examples

Getting Started with Go

The `XenServer-SDK` directory contains the following folders that are relevant to Go programmers:

- `XenServerGo`: The XenServer SDK for Go.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `src`: XenServerGo shipped as source files which can be used as a local module for other Go projects. Every API object is associated with one Go file. For example, the functions implementing the VM operations are contained within the file vm.go.

RPC protocol

The Go SDK supports the JSON-RPC v2.0 protocol.

Platform supported

- Windows
- Linux

This library requires Go 1.22 or greater.

NOTE:

This module is backwards compatible by using different names for API functions that have changed throughout different XenServer or Citrix Hypervisor versions. To communicate with hosts running different versions of Citrix Hypervisor or XenServer, it is advisable to check the versions supported by each function first. For example, `Host.Evacuate3` can only be used for servers running XAPI version 1.297.0 or above and `Host.Evacuate4` for servers running XAPI version 23.27.0 or above.

Installation

1. Extract the contents of the SDK ZIP file.
2. Navigate to the extracted `XenServer-SDK\XenServerGo` directory and copy the whole folder `src` into your Go project directory.

   To use XenServerGo SDK as a local Go module, include the following line into the `go.mod` file under your Go project:

   ```
   replace xenapi => ./src
   ```

   You can then import the XenServer® module with the following command:

   ```
   import "xenapi"
   ```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Before building your project, run the following Go commands.

```
go get -u all

go mod tidy
```

Examples

Examples on the usage of the XenServer SDK for Go can be found at XenServerGo usage examples on GitHub.

Go examples

Go examples

First steps with the API

Let's start our tour of the API by describing the calls required to create a VM on a XenServer® installation, and take it through a start/suspend/resume/stop cycle. This section does not reference code in any specific language. At this stage we just describe the informal sequence of RPC invocations that do our "install and start" task.

NOTE:

We recommend strongly against using the `VM.create` call, which might be removed or changed in a future version of the API. Read on to learn other ways to make a new VM.

Authentication: acquiring a session reference

The first step is to call `Session.login_with_password(username, password, client_API_version, originator)`. The API is session based, so before you can make other calls you must authenticate with the server. Assuming the user name and password are authenticated correctly, the result of this call is a *session reference*. Subsequent API calls take the session reference as a parameter. In this way, we ensure that only API users who are suitably authorized can perform operations on a XenServer installation. You can continue to use the same session for any number of API calls. When you have finished the session, it is recommended that you call `Session.logout(session)` to clean up (see section Logging out).

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Acquiring a list of templates to base a new VM installation on

The next step is to query the list of "templates" on the host. Templates are specially marked VM objects that specify suitable default parameters for various supported guest types. (If you want to see a quick enumeration of the templates on a XenServer installation for yourself, you can run the `xe template-list` CLI command.) To get a list of templates from the API, find the VM objects on the server that have their `is_a_template` field set to `true`. One way to do find these objects is by calling `VM.get_all_records(session)` where the session parameter is the reference we acquired from our `Session.login_with_password` call earlier. This call queries the server, returning a snapshot (taken at the time of the call) containing all the VM object references and their field values.

(Remember that at this stage we are not concerned with how the returned object references and field values can be manipulated in any particular client language: that detail is dealt with by each language-specific SDK and described concretely in the relevant chapter. For now, assume the existence of an abstract mechanism for reading and manipulating objects and field values returned by API calls.)

Now that we have a snapshot of the VM objects' field values in the memory of our client application, we can iterate through them and find the VMs that have their `is_a_template` value set to `true`. At this stage, let's assume that our example application further iterates through the template objects and remembers the reference corresponding to the one that has its `name_label` set to "Debian Buster 10" (one of the default Linux templates supplied with XenServer).

Installing the VM based on a template

Continuing through our example, we must now install a new VM based on the template we selected. The installation process requires two API calls:

- First we must now invoke the API call `VM.clone(session, t_ref, "my first VM")`. This call tells the server to clone the VM object referenced by `t_ref` to make a new VM object. The return value of this call is the VM reference corresponding to the newly created VM. Let's call this `new_vm_ref`.
- At this stage, the object referred to by `new_vm_ref` is still a template (like the VM object referred to by `t_ref`, from which it was cloned). To make `new_vm_ref` into a VM object, we must call `VM.provision(session, new_vm_ref)`. When this call returns the `new_vm_ref` object will have had its `is_a_template` field set to `false`, indicating that `new_vm_ref` now refers to a regular VM ready for starting.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTE:

The provision operation can take a few minutes, as it is during this call that the template's disk images are created. For the Debian template, the newly created disks are also at this stage populated with a Debian root filesystem.

Taking the VM through a start/suspend/resume/stop cycle

Now that we have an object reference representing our newly installed VM, it is trivial to take it through a few lifecycle operations:

- To start our VM, we can call `VM.start(session, new_vm_ref)`
- After it's running, we can suspend it by calling `VM.suspend(session, new_vm_ref)`,
- We can resume it by calling `VM.resume(session, new_vm_ref)`.
- We can call `VM.shutdown(session, new_vm_ref)` to shut down the VM cleanly.

Logging out

When an application is finished interacting with a XenServer host, it is good practice to call `Session.logout(session)`. This call invalidates the session reference (so it cannot be used in subsequent API calls) and deallocates server-side memory used to store the session object.

Although inactive sessions eventually time out, the server has a hardcoded limit of 500 concurrent sessions for each `username` or `originator`. After this limit has been reached, fresh logins evict the session objects that have been used least recently. The session references of these evicted session objects become invalid. For successful interoperability with other applications, concurrently accessing the server, the best policy is:

- Choose a string that identifies your application and its version.
- Create a single session at start-of-day, using that identifying string for the `originator` parameter to `Session.login_with_password`.
- Use this session throughout the application and then explicitly logout when possible.

NOTE:

Sessions can be used across multiple separate client-server *network connections*.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If a poorly written client leaks sessions or otherwise exceeds the limit, then as long as the client uses an appropriate `originator` argument, it is easily identifiable from the XenServer logs. XenServer destroys the longest-idle sessions of the rogue client only. This behavior might cause problems for that client but not for other clients. If the misbehaving client doesn't specify an `originator`, it is harder to identify and causes the premature destruction of other client sessions that also didn't specify an `originator`.

Install and start example: summary

We have seen how the API can be used to install a VM from a XenServer template and perform various lifecycle operations on it. Note that the number of calls we had to make to affect these operations was small:

- One call to acquire a session: `Session.login_with_password()`
- One call to query the VM (and template) objects present on the XenServer installation: `VM.get_all_records()`. Recall that we used the information returned from this call to select a suitable template to install from.
- Two calls to install a VM from our chosen template: `VM.clone()`, followed by `VM.provision()`.
- One call to start the resultant VM: `VM.start()` (and similarly other single calls to suspend, resume, and shut down accordingly)
- And then one call to log out `Session.logout()`.

Although the API as a whole is complex and fully featured, common tasks (such as VM lifecycle operations) are straightforward, requiring only a few simple API calls. Keep this fact in mind as you study the next section which might, on first reading, appear a little daunting!

Object Model Overview

This section gives a high-level overview of the object model of the API. For a more detailed description of the fields and methods of each class, see the XenServer Management API reference.

We start by giving a brief outline of some of the core classes that make up the API. (Don't worry if these definitions seem abstract in their initial presentation. The textual description in the following sections, and the code-sample walk through in the next section make these concepts concrete.)

Object identifiers

API objects can be identified by reference or by UUID.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

References are opaque types and are understood only by the particular server that generated them. They are not guaranteed to be permanent identifiers for objects. References generated during one session cannot be assumed to be valid for future sessions. References do not generally allow objects to be compared for equality. Two references to the same object are not guaranteed to be textually identical.

UUIDs are intended to be permanent identifiers for objects. Clients may store UUIDs on disk and use them to look up objects in subsequent sessions with the server. Clients may also test equality on objects by comparing UUID strings.

The API provides mechanisms for translating between UUIDs and opaque references. Each class that contains a `uuid` field provides:

- A `get_by_uuid` method that takes a UUID and returns an opaque reference to the server-side object that has that UUID;
- A `get_uuid` method that takes an opaque reference and returns the UUID of the server-side object that is referenced by it.

For a more detailed description of opaque references and UUIDs, see the XenServer Management API reference.

VM

A VM object represents a particular Virtual Machine instance on a XenServer host or resource pool. Example methods include `start`, `suspend`, `pool_migrate`. Example fields include `power_state`, `memory_static_max`, and `name_label`. (In the previous section we saw how the VM class is used to represent both templates and regular VMs).

Host

A host object represents a physical host in a XenServer pool. Example methods include `reboot` and `shutdown`. Example fields include `software_version`, `hostname`, and \[IP\]`address`.

VDI

A VDI object represents a Virtual Disk Image. Virtual Disk Images can be attached to VMs. A block device appears inside the VM through which the bits encapsulated by the attached Virtual Disk Image can be read and written. Example methods of the VDI class include `resize` and `clone`. Example fields include `virtual_size` and `sharable`. When we called `VM.provision` on the VM template in our previous example, some VDI objects were automatically created to represent the newly created disks. These VDIs were attached to the VM object.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SR

An SR object (Storage Repository) aggregates a collection of VDIs. It encapsulates the properties of physical storage on which the VDIs' bits reside. Example fields include:

- `type` which determines the storage-specific driver a XenServer installation uses to read/write the SR's VDIs
- `physical_utilisation`

Example methods include:

- `scan` which invokes the storage-specific driver to acquire a list of the VDIs contained with the SR and the properties of these VDIs
- `create` which initializes a block of physical storage so it is ready to store VDIs

Network

A network object represents a layer-2 network that exists in the environment in which the XenServer host instance lives. Since XenServer does not manage networks directly, network is a lightweight class that models physical and virtual network topology. VM and Host objects that are *attached* to a particular Network object can send network packets to each other. The objects are attached through VIF and PIF instances. For more information, see the following section.

If you are finding this enumeration of classes rather terse, you can skip to the code walk-throughs of the next chapter. There are plenty of useful applications that can be written using only a subset of the classes already described. If you want to continue this description of classes in the abstract, read on.

In addition to the classes listed in the previous section, there are four more that act as *connectors*. These *connectors* specify relationships between VMs, Hosts, Storage, and Networks. The first two of these classes that we consider, VBD and VIF, determine how VMs are attached to virtual disks and network objects respectively.

VBD

A VBD object (Virtual Block Device) represents an attachment between a VM and a VDI. When a VM is booted, its VBD objects are queried to determine which disk images (VDIs) to attach.

Example methods of the VBD class include:

- `plug` which *hot plugs* a disk device into a running VM, making the specified VDI accessible therein

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `unplug` which *hot unplugs* a disk device from a running guest

Example fields include:

- `device` which determines the device name inside the guest under which the specified VDI is made accessible

VIF

A VIF object (Virtual Network Interface) represents an attachment between a VM and a Network object. When a VM is booted, its VIF objects are queried to determine which network devices to create.

Example methods of the VIF class include:

- `plug` which *hot plugs* a network device into a running VM
- `unplug` which *hot unplugs* a network device from a running guest

The second set of "connector classes" that we consider determine how Hosts are attached to Networks and Storage.

PIF

A PIF object (Physical Interface) represents an attachment between a Host and a Network object. If a host is connected to a Network over a PIF, packets from the specified host can be transmitted/ received by the corresponding host. Example fields of the PIF class include:

- `device` which specifies the device name to which the PIF corresponds. For example, *eth0*
- `MAC` which specifies the MAC address of the underlying NIC that a PIF represents

PIFs abstract both physical interfaces and VLANs (the latter distinguished by the existence of a positive integer in the "VLAN" field).

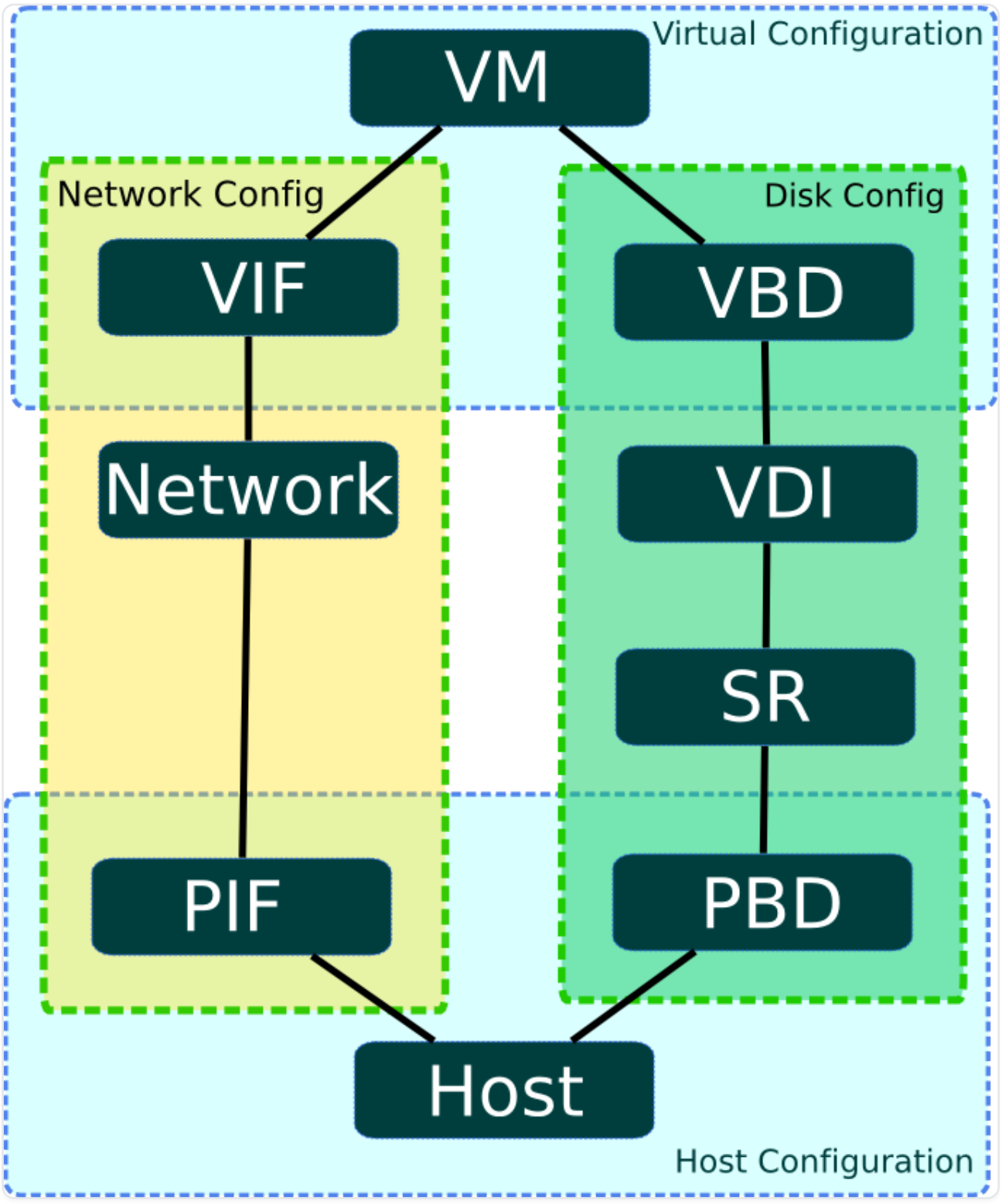
PBD

A PBD object (Physical Block Device) represents an attachment between a Host and an SR object. Fields include:

- `currently-attached` which specifies whether the chunk of storage represented by the specified SR object is available to the host
- `device_config` which specifies storage-driver specific parameters that determine how the low-level storage devices are configured on the specified host. For example, when an SR

© 1997–2026 Citrix Systems, Inc. All rights reserved.

rendered on an NFS filer, device_config can specify the host-name of the filer and the path on the filer in which the SR files live.



This figure presents a graphical overview of the API classes involved in managing VMs, Hosts, Storage, and Networking. From this diagram, the symmetry between storage and network configuration, and also the symmetry between virtual machine and host configuration is plain to see.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Working with VIFs and VBDs

In this section we walk through a few more complex scenarios. These scenarios describe how various tasks involving virtual storage and network devices can be done using the API.

Creating disks and attaching them to VMs

Let's start by considering how to make a new blank disk image and attach it to a running VM. We assume that we already have a running VM, and we know its corresponding API object reference. For example, we might have created this VM using the procedure described in the previous section and had the server return its reference to us. We also assume that we have authenticated with the XenServer installation and have a corresponding `session reference`. Indeed, the rest of this chapter, for the sake of brevity, does not mention sessions altogether.

Creating a new blank disk image

First, instantiate the disk image on physical storage by calling `VDI.create()`. The `VDI.create` call takes a number of parameters, including:

- `name_label` and `name_description`: a human-readable name/description for the disk (for example, for convenient display in the UI). These fields can be left blank if desired.
- `SR`: the object reference of the Storage Repository representing the physical storage in which the VDI's bits are placed.
- `read_only`: setting this field to true indicates that the VDI can *only* be attached to VMs in a read-only fashion. (Attempting to attach a VDI with its `read_only` field set to true in a read/write fashion results in error.)

Invoking the `VDI.create` call causes the XenServer installation to create a blank disk image on physical storage, create an associated VDI object (the datamodel instance that refers to the disk image on physical storage) and return a reference to this newly created VDI object.

The way in which the disk image is represented on physical storage depends on the type of the SR in which the created VDI resides. For example, if the SR is of type "lvm" then the new disk image will be rendered as an LVM volume; if the SR is of type "nfs" then the new disk image will be a sparse VHD file created on an NFS filer. (You can query the SR type through the API using the `SR.get_type()` call.)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTE:

Some SR types might round up the `virtual-size` value to make it divisible by a configured block size.

Attaching the disk image to a VM

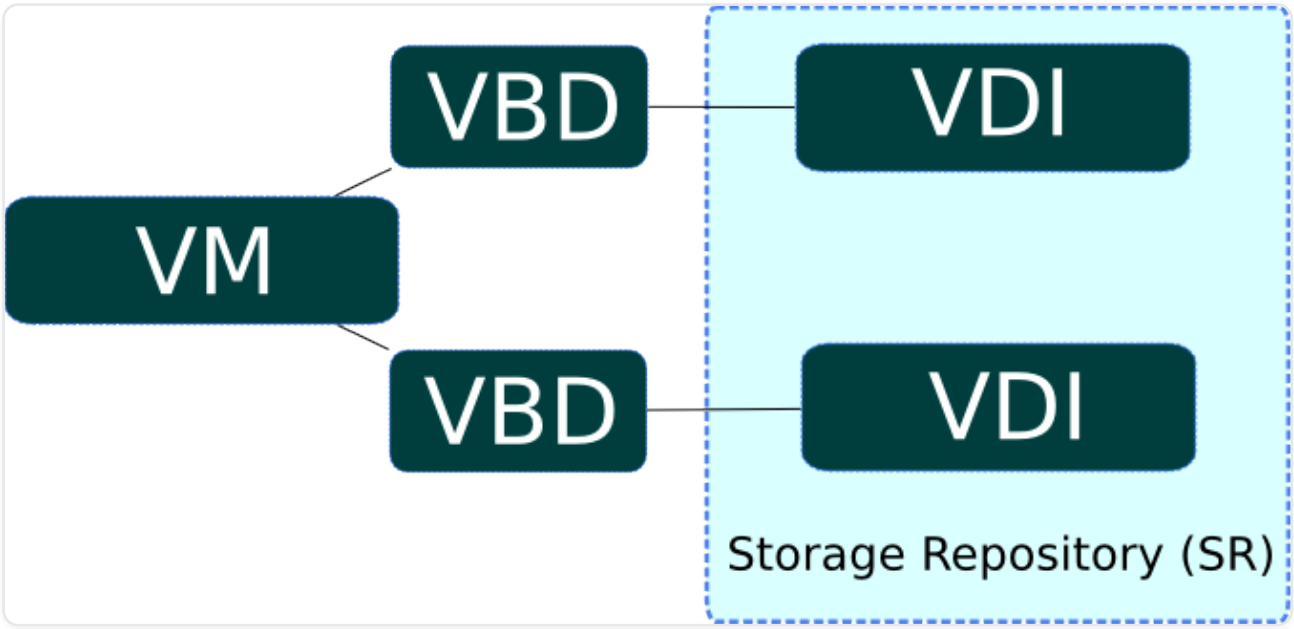
So far we have a running VM (that we assumed the existence of at the start of this example) and a fresh VDI that we just created. Right now, these are both independent objects that exist on the XenServer Host, but there is nothing linking them together. So our next step is to create such a link, associating the VDI with our VM.

The attachment is formed by creating a new "connector" object called a VBD (Virtual Block Device). To create our VBD we invoke the `VBD.create()` call. The `VBD.create()` call takes a number of parameters including:

- `VM` - the object reference of the VM to which the VDI is to be attached
- `VDI` - the object reference of the VDI that is to be attached
- `mode` - specifies whether the VDI is to be attached in a read-only or a read-write fashion
- `userdevice` - specifies the block device inside the guest through which applications running inside the VM will be able to read/write the VDI's bits.
- `type` - specifies whether the VDI is presented inside the VM as a regular disk or as a CD. (Note that this particular field has more meaning for Windows VMs than it does for Linux VMs, but we will not explore this level of detail in this chapter.)

Invoking `VBD.create` makes a VBD object on the XenServer installation and returns its object reference. However, this call in itself does not have any side-effects on the running VM (that is, if you go and look inside the running VM you will see that the block device has not been created). The fact that the VBD object exists but that the block device in the guest is not active, is reflected by the fact that the VBD object's `currently_attached` field is set to false.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



For expository purposes, this figure presents a graphical example that shows the relationship between VMs, VBDs, VDIs and SRs. In this instance a VM object has 2 attached VDIs: there are two VBD objects that form the connections between the VM object and its VDIs; and the VDIs reside within the same SR.

Hotplugging the VBD

If we rebooted the VM at this stage then, after rebooting, the block device corresponding to the VBD would appear: on boot, XenServer queries all VBDs of a VM and actively attaches each of the corresponding VDIs.

Rebooting the VM is all very well, but recall that we wanted to attach a newly created blank disk to a *running* VM. This can be achieved by invoking the `plug` method on the newly created VBD object. When the `plug` call returns successfully, the block device to which the VBD relates will have appeared inside the running VM -- i.e. from the perspective of the running VM, the guest operating system is led to believe that a new disk device has just been *hot plugged*. Mirroring this fact in the managed world of the API, the `currently_attached` field of the VBD is set to true.

Unsurprisingly, the VBD `plug` method has a dual called "`unplug`". Invoking the `unplug` method on a VBD object causes the associated block device to be *hot unplugged* from a running VM, setting the `currently_attached` field of the VBD object to false accordingly.

Creating and attaching Network Devices to VMs

The API calls involved in configuring virtual network interfaces in VMs are similar in many respects to the calls involved in configuring virtual disk devices. For this reason we will not run through a full example of how one can create network interfaces using the API object-model; instead we will use

© 1997–2026 Citrix Systems, Inc. All rights reserved.

this section just to outline briefly the symmetry between virtual *networking device* and virtual *storage device* configuration.

The networking analogue of the VBD class is the VIF class. Just as a VBD is the API representation of a block device inside a VM, a VIF (*Virtual network InterFace*) is the API representation of a network device inside a VM. Whereas VBDs associate VM objects with VDI objects, VIFs associate VM objects with Network objects. Just like VBDs, VIFs have a `currently_attached` field that determines whether or not the network device (inside the guest) associated with the VIF is currently active or not. And as we saw with VBDs, at VM boot-time the VIFs of the VM are queried and a corresponding network device for each created inside the booting VM. Similarly, VIFs also have `unplug` and `unplug` methods for hot plugging/unplugging network devices in/out of running VMs.

Host configuration for networking and storage

We have seen that the VBD and VIF classes are used to manage configuration of block devices and network devices (respectively) inside VMs. To manage host configuration of storage and networking there are two analogous classes: PBD (Physical Block Device) and PIF (Physical \[network\] Interface).

Host storage configuration: PBDs

Let us start by considering the PBD class. A `PBD.create()` call takes a number of parameters including:

PARAMETER	DESCRIPTION
host	physical machine on which the PBD is available
SR	the Storage Repository that the PBD connects to
device_config	a string-to-string map that is provided to the host's SR-backend-driver, containing the low-level parameters required to configure the physical storage device(s) on which the SR is to be realized. The specific contents of the `device_config` field depend on the type of the SR to which the PBD is connected. (Running `xe sm-list` will show a list of possible SR types; the *configuration* field in

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PARAMETER	DESCRIPTION
	this enumeration specifies the `device_config` parameters that each SR type expects.)

For example, imagine we have an SR object *s* of type "nfs" (representing a directory on an NFS filer within which VDIs are stored as VHD files); and let's say that we want a host, *h*, to be able to access *s*. In this case we invoke `PBD.create()` specifying host *h*, SR *s*, and a value for the *device_config* parameter that is the following map:

```
("server", "my_nfs_server.example.com"), ("serverpath", "/scratch/mysrs/sr1")
```

This tells the XenServer host that SR *s* is accessible on host *h*, and further that to access SR *s*, the host needs to mount the directory `/scratch/mysrs/sr1` on the NFS server named `my_nfs_server.example.com`.

Like VBD objects, PBD objects also have a field called `currently_attached`. Storage repositories can be attached and detached from a given host by invoking `PBD.plug` and `PBD.unplug` methods respectively.

Host networking configuration: PIFs

Host network configuration is specified by virtue of PIF objects. If a PIF object connects a network object, *n*, to a host object *h*, then the network corresponding to *n* is bridged onto a physical interface (or a physical interface plus a VLAN tag) specified by the fields of the PIF object.

For example, imagine a PIF object exists connecting host *h* to a network *n*, and that `device` field of the PIF object is set to `eth0`. This means that all packets on network *n* are bridged to the NIC in the host corresponding to host network device `eth0`.

Importing and Exporting VMs

VMs can be exported to a file and later imported to any XenServer host. Because the import and export operations can take some time to complete, they are performed using an asynchronous HTTP(S) interface exposed by XenServer.

XenServer supports a VM input format called XVA. This is a format specific to Xen-based hypervisors and packages a single VM as a single file archive consisting of a metadata descriptor file and disk images.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Detailed information on how to use the HTTP(S) interface to import and export VMs, and a description of the XVA file format can be found in chapter Using HTTP to interact with XenServer.

NOTE:

Single VMs and virtual appliances (vApps) of multiple VMs can be exported to and imported from OVF/OVA packages using XenCenter. For more details visit chapter Importing and Exporting VMs of XenCenter's documentation.

Where to look next

In this chapter we have presented a brief high-level overview of the API and its object-model. The aim here is not to present the detailed semantics of the API, but just to provide enough background for you to start reading the code samples of the next chapter and to find your way around the more detailed XenServer Management API reference.

There are a number of places you can find more information:

- The Command line interface documentation contains an overview of the `xe` CLI. Since a good deal of `xe` commands are a thin veneer over the API, playing with `xe` is a good way to start finding your way around the API object model described in this chapter.
- The code samples in the next chapter provide some concrete instances of API coding in a variety of client languages.
- The XenServer Management API reference provides a more detailed description of the API semantics as well as the wire protocol of the RPC messages.
- There are a few scripts that use the API in the XenServer host dom0 itself under `/opt/xensource/libexec/`.
- There is a number of examples for each of the SDK languages at XenServer SDK usage examples on GitHub. These include ready-to-run programs demonstrating operations like creating a VM and taking it through a start/suspend/resume/stop cycle, monitoring events, creating a shared SR, migrating a VM between servers in a pool, etc.

Using the API

This chapter describes how to use the XenServer® Management API to write real programs to manage XenServer hosts and VMs. The chapter begins with a walk-through of a typical client application and demonstrates how the API can be used to perform common tasks. Example code

© 1997–2026 Citrix Systems, Inc. All rights reserved.

fragments are given in Python syntax but equivalent code in the other programming languages would look very similar. The chapter finishes with walk-throughs of two complete examples.

Anatomy of a typical application

This section describes the structure of a typical application using the XenServer Management API. Most client applications begin by connecting to a XenServer host and authenticating using a username and password. Assuming the authentication succeeds, the server will create a "session" object and return a reference to the client. This reference will be passed as an argument to all future API calls. Once authenticated, the client may search for references to other useful objects (XenServer hosts, VMs, SRs, and so on) and invoke operations on them. Operations may be invoked either synchronously or asynchronously; special task objects represent the state and progress of asynchronous operations. These application elements are all described in detail in the following sections.

Choosing a low-level transport

API calls can be issued over two transports:

- SSL-encrypted TCP on port 443 (https) over an IP network
- plaintext over a local Unix domain socket: `/var/xapi/xapi`

Switching from the XML-RPC to the JSON-RPC backend can be done by adding the suffix `/jsonrpc` to the host URL path.

The SSL-encrypted TCP transport is used for all off-host traffic, while the Unix domain socket can be used from services running directly on the XenServer host itself. In the SSL-encrypted TCP transport, all API calls must be directed at the pool coordinator. If directed at a pool supporter, the calls will fail with the error `HOST_IS_SLAVE`, which includes the IP address of the coordinator as an error parameter.

Because the coordinator host of a pool can change, especially if HA is enabled on a pool, clients must implement the following steps to detect a coordinator host change and connect to the new coordinator as required:

Handling pool coordinator changes

1. Subscribe to updates in the list of hosts servers, and maintain a current list of hosts in the pool
2. If the connection to the pool coordinator fails to respond, attempt to connect to all hosts in the list until one responds

© 1997–2026 Citrix Systems, Inc. All rights reserved.

3. If the first host to respond is not the new coordinator, it returns the `HOST_IS_SLAVE` error message, which contains the identity of the new pool coordinator
4. Connect to the new coordinator

NOTE:

As a special-case, all messages sent through the Unix domain socket are transparently forwarded to the correct node.

Authentication and session handling

The vast majority of API calls take a session reference as their first parameter; failure to supply a valid reference will result in a `SESSION_INVALID` error being returned. Acquire a session reference by supplying a user name and password to the `login_with_password` function.

NOTE:

As a special-case, if this call is run over the local Unix domain socket then the user name and password are ignored and the call always succeeds.

Every session has an associated "last active" timestamp which is updated on every API call. The server software currently has a built-in limit of 500 active sessions and will remove those with the oldest "last active" field if this limit is exceeded for a given `username` or `originator`. In addition all sessions whose "last active" field is older than 24 hours are also removed. Therefore it is important to:

- Specify an appropriate `originator` when logging in; and
- Remember to log out of active sessions to avoid leaking them; and
- Be prepared to log in again to the server if a `SESSION_INVALID` error is caught.

NOTE:

A session reference obtained by a login request to the XML-RPC backend can be used in subsequent requests to the JSON-RPC backend, and vice-versa.

In the following Python fragment a connection is established over the Unix domain socket and a session is created:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
import XenAPI



session = XenAPI.xapi_local()

try:

    session.xenapi.login_with_password("root", "", "2.20", "My Widget
v0.1")

    ...

finally:

    session.xenapi.session.logout()
```

Finding references to useful objects

Once an application has authenticated the next step is to acquire references to objects in order to query their state or invoke operations on them. All objects have a set of "implicit" messages which include the following:

- `get_by_name_label` : return a list of all objects of a particular class with a particular label;
- `get_by_uuid` : return a single object named by its UUID;
- `get_all` : return a set of references to all objects of a particular class; and
- `get_all_records` : return a map of reference to records for each object of a particular class.

For example, to list all hosts:

```
hosts = session.xenapi.host.get_all()
```

To find all VMs with the name "my first VM":

```
vms = session.xenapi.VM.get_by_name_label('my first VM')
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTE:

Object `name_label` fields are not guaranteed to be unique and so the `get_by_name_label` API call returns a set of references rather than a single reference.

In addition to the methods of finding objects described above, most objects also contain references to other objects within fields. For example it is possible to find the set of VMs running on a particular host by calling:

```
vms = session.xenapi.host.get_resident_VMs(host)
```

Invoking synchronous operations on objects

Once object references have been acquired, operations may be invoked on them. For example to start a VM:

```
session.xenapi.VM.start(vm, False, False)
```

All API calls are by default synchronous and will not return until the operation has completed or failed. For example in the case of `VM.start` the call does not return until the VM has started booting.

NOTE:

When the `VM.start` call returns, the VM will be booting. To determine when the booting has finished, wait for the in-guest agent to report internal metrics through the `VM_guest_metrics` object.

Using Tasks to manage asynchronous operations

To simplify managing operations which take quite a long time (for example, `VM.clone` and `VM.copy`) functions are available in two forms: synchronous (the default) and asynchronous. Each asynchronous function returns a reference to a task object which contains information about the in-progress operation including:

- whether it is pending
- whether it is has succeeded or failed
- progress (in the range 0-1)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- the result or error code returned by the operation

An application which wanted to track the progress of a `VM.clone` operation and display a progress bar would have code like the following:

```
vm = session.xenapi.VM.get_by_name_label('my vm')

task = session.xenapi.Async.VM.clone(vm)

while session.xenapi.task.get_status(task) == "pending":

    progress = session.xenapi.task.get_progress(task)

    update_progress_bar(progress)

    time.sleep(1)

session.xenapi.task.destroy(task)
```

NOTE:

A well-behaved client must delete tasks created by asynchronous operations when it has finished reading the result or error. If the number of tasks exceeds a built-in threshold then the server will delete the oldest of the completed tasks.

Subscribing to and listening for events

With the exception of the task and metrics classes, whenever an object is modified the server generates an event. Clients can subscribe to this event stream on a per-class basis and receive updates rather than resorting to frequent polling. Events come in three types:

- `add` - generated when an object has been created;
- `del` - generated immediately before an object is destroyed; and
- `mod` - generated when an object's field has changed.

Events also contain a monotonically increasing ID, the name of the class of object, and a snapshot of the object state equivalent to the result of a `get_record()`.

Clients can receive events by calling `event.from()` with a list of class names or the special string `*` (to receive events for all classes). The output of `event.from()` includes a token, which

© 1997–2026 Citrix Systems, Inc. All rights reserved.

can be passed into a subsequent `event.from()` call to receive only the events that have occurred since the last call. If an empty string is passed, `event.from()` will return all events.

The following Python code fragment demonstrates how to print a summary of events for all classes generated by a system:

```
token = ''

fmt = "%8s %s %20s  %5s  %s %s"



while True:

    try

        output = session.xenapi.event_from("*", token, 30.0)

        events = output['events']

        token = output['token']



        for event in events:

            name = "n/a"

            snapshot = ''

            if "snapshot" in event.keys():

                snapshot = event['snapshot']

                if "name_label" in snapshot.keys():

                    name = snapshot['name_label']

            print fmt % (event['id'], event["ref"], event['class'],
event['operation'], name, repr(snapshot))
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        time.sleep(10)
    finally:
        session.xenapi.session.logout()
```

The full script can be found at `watch-all-events.py` on Github.

Complete application examples

This section describes two complete examples of real programs using the API.

Simultaneously migrating VMs using live migration

This python example (the full script can be found at `permute.py` on Github) demonstrates how to use live migration to move VMs simultaneously between hosts in a Resource Pool. The example makes use of asynchronous API calls and shows how to wait for a set of tasks to complete.

The program begins with some standard boilerplate and imports the API module

```
import sys, time

import XenAPI
```

Next the commandline arguments containing a server URL, user name, password and a number of iterations are parsed. The user name and password are used to establish a session which is passed to the function `main`, which is called multiple times in a loop. Note the use of `try: finally:` to make sure the program logs out of its session at the end.

```
if __name__ == "__main__":
    if len(sys.argv) <> 5:
        print "Usage:"
        print sys.argv[0], " <url> <username> <password> <iterations>"
        sys.exit(1)
    url = sys.argv[1]
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    username = sys.argv[2]
    password = sys.argv[3]
    iterations = int(sys.argv[4])
    # First acquire a valid session by logging in:
    session = XenAPI.Session(url)
    session.xenapi.login_with_password(username, password, "2.3",
                                       "Example migration-demo v0.1")
    try:
        for i in range(iterations):
            main(session, i)
    finally:
        session.xenapi.session.logout()
```

The `main` function examines each running VM in the system, taking care to filter out *control domains* (which are part of the system and not controllable by the user). A list of running VMs and their current hosts is constructed.

```
def main(session, iteration):
    # Find a non-template VM object
    all = session.xenapi.VM.get_all()
    vms = []
    hosts = []
    for vm in all:
        record = session.xenapi.VM.get_record(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        if not(record["is_a_template"]) and \
            not(record["is_control_domain"]) and \
            record["power_state"] == "Running":
            vms.append(vm)
            hosts.append(record["resident_on"])
    print "%d: Found %d suitable running VMs" % (iteration, len(vms))
```

Next the list of hosts is rotated:

```
# use a rotation as a permutation
hosts = [hosts[-1]] + hosts[:(len(hosts)-1)]
```

Each VM is then moved using live migration to the new host under this rotation (that is, a VM running on host at position 2 in the list is moved to the host at position 1 in the list, and so on.) In order to execute each of the movements in parallel, the asynchronous version of the `VM.pool_migrate` is used and a list of task references constructed. Note the `live` flag passed to the `VM.pool_migrate`; this causes the VMs to be moved while they are still running.

```
tasks = []
    for i in range(0, len(vms)):
        vm = vms[i]
        host = hosts[i]
        task = session.xenapi.Async.VM.pool_migrate(vm, host, { "live":
"true" })
        tasks.append(task)
```

The list of tasks is then polled for completion:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
finished = False

    records = {}

    while not(finished):

        finished = True

        for task in tasks:

            record = session.xenapi.task.get_record(task)

            records[task] = record

            if record["status"] == "pending":

                finished = False

        time.sleep(1)
```

Once all tasks have left the *pending* state (i.e. they have successfully completed, failed or been cancelled) the tasks are polled once more to see if they all succeeded:

```
allok = True

    for task in tasks:

        record = records[task]

        if record["status"] <> "success":

            allok = False
```

If any one of the tasks failed then details are printed, an exception is raised and the task objects left around for further inspection. If all tasks succeeded then the task objects are destroyed and the function returns.

```
if not(allok):

        print "One of the tasks didn't succeed at", \
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
            time.strftime("%F:%HT%M:%SZ", time.gmtime())

        idx = 0

        for task in tasks:

            record = records[task]

            vm_name = session.xenapi.VM.get_name_label(vms[idx])

            host_name = session.xenapi.host.get_name_label(hosts[idx])

            print "%s : %12s %s -> %s [ status: %s; result = %s; error =
%s ]" % \

                    (record["uuid"], record["name_label"], vm_name,
host_name,      \

                    record["status"], record["result"],
repr(record["error_info"]))

            idx = idx + 1

        raise "Task failed"

    else:

        for task in tasks:

            session.xenapi.task.destroy(task)
```

Cloning a VM using the xe CLI

This example is a `bash` script which uses the xe CLI to clone a VM taking care to shut it down first if it is powered on.

The example begins with some boilerplate which first checks if the environment variable `XE` has been set: if it has it assumes that it points to the full path of the CLI, else it is assumed that the xe CLI is on the current path. Next the script prompts the user for a server name, user name and password:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
# Allow the path to the 'xe' binary to be overridden by the XE environment
variable

if [ -z "${XE}" ]; then

    XE=xe

fi



if [ ! -e "${HOME}/.xe" ]; then

    read -p "Server name: " SERVER

    read -p "Username: " USERNAME

    read -p "Password: " PASSWORD

    XE="${XE} -s ${SERVER} -u ${USERNAME} -pw ${PASSWORD}"

fi
```

Next the script checks its commandline arguments. It requires exactly one: the UUID of the VM which is to be cloned:

```
# Check if there's a VM by the uuid specified

${XE} vm-list params=uuid | grep -q " ${vmuuid}$"

if [ $? -ne 0 ]; then

        echo "error: no vm uuid \"${vmuuid}\" found"

        exit 2

fi
```

The script then checks the power state of the VM and if it is running, it attempts a clean shutdown. The event system is used to wait for the VM to enter state "Halted".

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTE:

The xe CLI supports a command-line argument `--minimal` which causes it to print its output without excess whitespace or formatting, ideal for use from scripts. If multiple values are returned they are comma-separated.

```
# Check the power state of the vm

name=$(${XE} vm-list uuid=${vmuuid} params=name-label --minimal)

state=$(${XE} vm-list uuid=${vmuuid} params=power-state --minimal)

wasrunning=0



# If the VM state is running, we shutdown the vm first

if [ "${state}" = "running" ]; then

        ${XE} vm-shutdown uuid=${vmuuid}

        ${XE} event-wait class=vm power-state=halted uuid=${vmuuid}

        wasrunning=1

fi
```

The VM is then cloned and the new VM has its `name_label` set to `cloned_vm`.

```
# Clone the VM

newuuid=$(${XE} vm-clone uuid=${vmuuid} new-name-label=cloned_vm)
```

Finally, if the original VM had been running and was shutdown, both it and the new VM are started.

```
# If the VM state was running before cloning, we start it again

# along with the new VM.
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
if [ "$wasrunning" -eq 1 ]; then

        ${XE} vm-start uuid=${vmuuid}

        ${XE} vm-start uuid=${newuuid}

fi
```

Using HTTP to interact with XenServer®

XenServer exposes an HTTP(S) interface on each host that can be used to perform various operations.

Each operation is performed by constructing an HTTP(S) GET or POST call to a dedicated URL path. Parameters are appended to the URL following a question mark (?) and separated by ampersands (&). The calls require authentication by providing a valid management API session reference as a query parameter. For example:

```
wget https://<server>/services?session_id=<session_opaque_ref>
```

NOTE:

Basic auth using a username and password is also available:

```
wget https://username:password@&lt;server>/services
```

However, this method is not recommended.
The following sections describe in greater detail some of the available operations.

Importing and exporting VMs as XVA packages

Because the import and export of VMs can take some time to complete, an asynchronous HTTP(S) interface is provided for these operations.

VM Export

To export a VM as an XVA package using the XenServer Management API, construct an HTTP(S) GET call. The call accepts the following query parameters:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PARAMETER	DESCRIPTION
`session_id`	The opaque reference of the session being used to authenticate.
`uuid`	The `uuid` of the VM; required only if not using the `ref` parameter.
`ref`	The opaque reference of the VM; required only if not using the `uuid` parameter.
`task_id`	(Optional) The opaque reference of a task object with which to monitor the progress of the operation; the task object needs to be created first.

For example, to export a VM to the XVA package `vm.xva` using the Linux command line utility cURL:

```
curl https://server/export?
session_id=<session_opaque_ref>&task_id=<task_opaque_ref>&uuid=<vm_uuid>
-o vm.xva
```

Note that you can perform this operation on the pool coordinator even if the VM runs on a pool supporter. The request might result in a redirect if the VM's disks are placed on an SR accessible only by a pool member.

If you want to export only the VM metadata without the disks, use the HTTP(S) handler `export_metadata`. For example:

```
curl https://server/export_metadata?
session_id=<session_opaque_ref>&ref=<vm_opaque_reference> -o vm_meta.xva
```

VM Import

To import a VM from an XVA package, construct an HTTP(S) PUT call. The call accepts the following query parameters:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PARAMETER	DESCRIPTION
`session_id`	The opaque reference of the session being used to authenticate.
`sr_id`	(Optional) The opaque reference of the SR on which the imported VM's disks will be placed. If not specified, the disks will be imported to `Pool.default_SR`. If the latter if not set, the error `DEFAULT_SR_NOT_FOUND` is returned.
`restore`	(Optional) If `true`, the import is treated as replacing the original VM from which the XVA package was exported. The implication of this currently is that the MAC addresses on the VIFs are exactly as the export was, which will lead to conflicts if the original VM is still running.
`force`	(Optional) If `true`, any checksum failures will be ignored (the default is to destroy the VM if a checksum error is detected).
`task_id`	(Optional) The opaque reference of a task object with which to monitor the progress of the operation; the task object needs to be created first

For example, to import a VM from package `vm.xva` placing its disks on a specific SR:

```
curl -T vm.xva http://server/import?
session_id=<session_opaque_ref>&sr_id=<sr_opaque_ref>
```

Or, to import a VM from package `vm.xva` placing its disks on the default SR, while monitoring the progress of the operation:

```
curl -T vm.xva http://server/import?
session_id=<session_opaque_ref>&task_id=<task_opaque_ref>
```

To import just the metadata, use the HTTP(S) handler `import_metadata`:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
curl -T vm.xva http://server/import_metadata?
session_id=<session_opaque_ref>
```

VDI Import

To import just a Virtual Disk Image (VDI), construct an HTTP(s) PUT call. HTTPS is preferred. The call accepts the following query parameters:

PARAMETER	DESCRIPTION
`session_id`	The opaque reference of the session being used to authenticate.
`vdi`	(Optional) The opaque reference or UUID of the VDI to import into, VDI must already exist. If not supplied `sr_id` or `sr_uuid` must be supplied.
`sr_id`	(Optional) The opaque ref of the SR to import into, a new VDI of appropriate size will be created, only supported for `raw` format
`sr_uuid`	(Optional) The UUID of the SR to import into, a new VDI of appropriate size will be created, only supported for `raw` format
`format`	(Optional) The format of the data stream, raw (the default), vhd, tar.
`task_id`	(Optional) The opaque reference of a task object with which to monitor the progress of the operation; the task object needs to be created first
`chunked`	(Optional) If `true` the import uses chunked transfers, only supported for `raw` format. NOTE: this does not imply the use of HTTP Chunked Transfer Coding as per section 7.1 of RFC 9112 and specifying these header options will return an error. Once a connection is established the client shall send sequences of [signed 64 bit offset, signed 32 bit length, data of length], finishing with a 0 offset, 0 length

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PARAMETER	DESCRIPTION
	header and no data. Negative values for offset or length are not valid. Offset and length are Little Endian. The VDI must already exist, i.e. `vdi` must be specified and `sr_id` and `sr_uuid` must not be.

For example, to import a raw disk data file, i.e. where there is no structured format in the file (as in the `vhd` case) while monitoring the progress of the operation.

```
curl -T file.raw 'https://server/import_raw_vdi?
session_id=<session_opaque_ref>&task_id=<task_opaque_ref>&vdi=<vdi_opaque_
ref>&format=raw'
```

Example: to send 1 MB extents of random data at 10 MB intervals using the `chunked` option, in Python

```
from io import BytesIO

import os

import requests

from struct import pack

from urllib import parse




def upload(session_ref, vdi_ref, base_url):

    request_url = parse.urljoin(base_url, 'import_raw_vdi')

    params = {'session_id': session_ref, 'vdi': vdi_ref, 'format': 'raw',
'chunked': True}



    stream_data = BytesIO()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    # Write 1MB random data at 10 MB intervals
    for i in range(0, 9):
        offset = i * 10 * 1024 * 1024
        size = 1 * 1024 * 1024
        data = os.urandom(size)

        stream_data.write(pack('<q', offset))
        stream_data.write(pack('<i', size))
        stream_data.write(data)

    # Write completion header
    stream_data.write(pack('<q', 0))
    stream_data.write(pack('<i', 0))

    upload_data = stream_data.getvalue()

    r = requests.put(request_url, params=params, data=upload_data,
headers={'Content-Length': str(len(upload_data))})
    r.raise_for_status()
```

The `Content-Length` header is required and must therefore be predetermined even if the data to be uploaded will be produced by a generator function feeding the HTTP request this must include the length of the final completion header.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Xen® Virtual Appliance (XVA) VM Import Format

XenServer supports a human-readable VM input format called XVA. This section describes the syntax and structure of the format.

An XVA is essentially a `tar` archive containing XML metadata and a set of disk images. A VM represented by an XVA is not intended to be directly executable. The data within an XVA package is intended for either archiving on permanent storage or for being transmitted to a VM server - such as a XenServer host - where it can be extracted and run.

XVA is a hypervisor-neutral packaging format; it is possible to create simple tools to instantiate an XVA VM on any other platform. XVA does not specify any particular runtime format; for example disks may be instantiated as file images, LVM volumes, QCoW images, VMDK or VHD images. An XVA VM may be instantiated any number of times, each instantiation may have a different runtime format.

Unlike other formats like the Open Virtual Appliance (OVA) format, XVA does not:

- specify any particular serialization or transport format
- provide any mechanism for customizing VMs (or templates) on install
- address how a VM may be upgraded post-install
- define how multiple VMs, acting as an appliance, may communicate.

The extracted content of an XVA file is a directory containing, at a minimum, a file called `ova.xml`. This file contains the metadata describing the VM packaged within the XVA. Each disk image is stored within a sub-directory and is referenced from the `ova.xml`. For example, the XVA describing a VM with two disks would contain the following metadata file and sub-directories:

```
$ls -al

total 72

drwx------+ 1 user1 Domain Users     0 Nov  7 10:35 .

drwxrwx---+ 1 user1 Domain Users     0 Nov  7 10:35 ..

-r-x------+ 1 user1 Domain Users 56046 Jan  1  1970 ova.xml

drwx------+ 1 user1 Domain Users     0 Nov  7 10:34 Ref_6
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
drwx------+ 1 user1 Domain Users     0 Nov  7 10:34 Ref_9
```

The file `ova.xml` contains the VM metadata in XML-RPC format, for example:

```
<value>
  <struct>
    <member>
      <name>objects</name>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member><name>class</name><value>VM</value></member>
                <member><name>id</name><value>Ref:VM</value></member>
                <member>
                  <name>snapshot</name>
                  <value>
                    <struct>
                      <member><name>power_state</name><value>Halted</
value></member>
                      <member><name>name_label</name><value>Test VM</
value></member>
                      <member><name>memory_static_max</
name><value>4294967296</value></member>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
                        <member><name>VCPUs_max</name><value>2</value></
member>

                        <member><name>VIFs</
name><value><array><data><value>Ref:VIF</value></data></array></value></
member>

                        <member><name>PV_bootloader</name><value>pygrub</
value></member>

                        <member><name>HVM_boot_policy</name><value>BIOS
order</value></member>

                        <!-- ... -->

                      </struct>

                    </value>

                  </member>

                </struct>

              </value>

              <value>

                <struct>

                  <member><name>class</name><value>VIF</value></member>

                  <member><name>id</name><value>Ref:VIF</value></member>

                  <member>

                    <name>snapshot</name>

                    <value>

                      <!--...-->

                    </value>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
                </member>

              </struct>

            </value>

            <!--...-->

          </data>

        </array>

      </value>

    </member>

  </struct>

</value>
```

A single disk image is represented by a directory containing a sequence of files as follows:

```
$ls -al Ref_6

total 2054

drwx------+ 1 user1 Domain Users       0 Nov  7 13:40 .

drwx------+ 1 user1 Domain Users       0 Nov  7 10:35 ..

-r-x------+ 1 user1 Domain Users 1048576 Jan  1  1970 00000000

-r-x------+ 1 user1 Domain Users      16 Jan  1  1970 00000000.xxhash

-r-x------+ 1 user1 Domain Users 1048576 Jan  1  1970 00000499

-r-x------+ 1 user1 Domain Users      16 Jan  1  1970 00000499.xxhash

...
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Each of the extensionless files contains a block of raw disk image data of size 1MiB. The filename is the block number in decimal. The small size was chosen to be safely under the maximum file size limit of several filesystems. If these files are concatenated, the original image is recovered.

Each of the data block files is accompanied by a file with the same base name and the extension `.xxhash`, which contains the checksum of the corresponding data block file calculated using the xxHash algorithm.

NOTE:

In XenServer versions prior to Citrix Hypervisor 8.1, the checksums of the disk image blocks were calculated using the SHA1 algorithm, and the files containing the checksums had the extension `.checksum`. For example, a directory representing a disk image would contain the following sequence of files:

```
$ls -al

total 3079

drwx------+ 1 user1 Domain Users       0 Nov  7 14:20 .

drwx------+ 1 user1 Domain Users       0 Nov  7 10:35 ..

-rwx------+ 1 user1 Domain Users 1048576 Aug  4 09:50 0000000000000000000

-rwx------+ 1 user1 Domain Users      40 Aug  4 09:50
0000000000000000000.checksum

-rwx------+ 1 user1 Domain Users 1048576 Aug  4 09:50 0000000000000000001

-rwx------+ 1 user1 Domain Users      40 Aug  4 09:50
0000000000000000001.checksum

-rwx------+ 1 user1 Domain Users 1048576 Aug  4 09:50 0000000000000000003

-rwx------+ 1 user1 Domain Users      40 Aug  4 09:50
0000000000000000003.checksum

...
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Getting XenServer Operational Metrics

XenServer records metrics about the operation of various aspects of your XenServer installation. The metrics are stored persistently for long term access and analysis of historical trends.

Metrics available

A wide range of metrics are available for XenServer including: C-State, P-State, IOPS, Latency and many more. Not all of these are turned on by default. The full range of metrics available for both Hosts and VMs are detailed in section Monitor XenServer Performance of XenServer's documentation. This chapter also details how you can explore these metrics via XenCenter.

The data store for operational metrics

Metrics are stored in Round Robin Databases (RRDs), which are maintained for individual VMs (including the control domain) and the server. Each database consists of multiple Round Robin Archives (RRAs) and is of a fixed size. This is because each RRA is in effect a circular buffer with a predefined maximum capacity.

The VM RRDs are resident on the server on which the VM is running, or the pool coordinator when the VM is not running. This means that the location of the VM must be known in order to retrieve the associated data.

The RRDs are also backed up every day. Metrics are persisted for a maximum of one year, and are stored at different granularities.

RRDs are saved to disk as uncompressed XML. The size of each RRD when written to disk ranges from 200KiB to approximately 1.2MiB when the RRD stores the full year of metrics.

NOTE:

If metrics cannot be written to disk, for example when a disk is full, metrics will be lost and the last saved version of the RRD will be used.

Data Granularity

Each archive in the database samples its particular metric at a specified granularity. The values are stored at intervals of:

- 5 seconds for the past 10 minutes
- one minute for the past 2 hours
- one hour for the past week

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- one day for the past year

The sampling that takes place every 5 seconds records actual data points, however, the following RRAs contain Consolidated Data Points (CDPs) which are produced by applying the Consolidation Functions (CFs) to a number of actual data points, storing historical data in a far more compressed format. The CFs supported by XenServer are:

- AVERAGE
- MIN
- MAX

Downloading RRDs

Metrics can be downloaded over HTTP(s), for example using `wget`.

NOTE:

In early versions of the XenServer Management API, instantaneous operational metrics could be obtained using the `VM_metrics`, `VM_guest_metrics`, `host_metrics` classes and associated methods. These methods have been deprecated in favor of using the HTTP(S) interface described in this chapter to download the metrics from the RRDs on the VMs and servers. The legacy metrics APIs will not return any values.

Metrics can be downloaded in `xport` style XML or JSON format. See rrddump and rrdxport for information about the XML format. The JSON format has the same structure as the XML.

Starting from xapi version 23.17.0, the server decides which format to return using the HTTP header `Accept`. To download metrics in XML format use `'Accept: text/xml'`, while to download metircs in JSON format use `'Accept: application/json'`. When both formats are accepted, for example, when using `'Accept: */*'` (which is the default for the `wget` client), the server returns JSON. The content type is provided in the reponse's headers.

NOTE:

When no header is specified, for example, when using `'Accept: '`, the server returns XML; to retrieve JSON you need to use the query parameter `json`. The header takes precedence over the query parameter and is the recommended method for deciding the returned format.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Downloading the whole RRD

RRDs can be downloaded from the XenServer host on which they reside by using the HTTP(S) handler registered at `/host_rrd` or `/vm_rrd`.

For example, to download a host RRD:

```
wget https://<server>/host_rrd?session_id=<session_opaque_reference>
```

To download a VM RRD, you must also specify the VM's `uuid` as a query parameter:

```
wget https://<server>/vm_rrd?
session_id=<session_opaque_reference>&uuid=<vm_uuid>
```

Getting updates from the RRD

So as to not have to download the whole database when you have most of the data already, a query can also be made to retrieve just updates. This is achieved via the HTTP(S) handler `/rrd_updates`.

Updates are described in relation to a start time which is specified as epoch time (number of seconds since Jan 1st, 1970) in a query parameter named `start`.

To obtain an update of all VM metrics on a server, the URL would be of the form:

```
wget http://<server>/rrd_updates?
session_id=<session_opaque_reference>&start=<start_time>
```

The response contains data for every VM resident on the particular host that is being queried. This means that it is not possible to query a particular VM on its own. To differentiate which column in the export is associated with which VM, the `legend` field is prefixed with the VM's `uuid`. The data set also contains a prefix describing the CF used to collect the data (`MAX/MIN/AVERAGE`).

To obtain host updates too, add the query parameter `host=true`:

```
wget http://<server>/rrd_updates?
session_id=<session_opaque_reference>&start=<start_time>&host=true
```

When using RRD updates, you should be aware of the following:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `/rrd_updates` will only return data for the metrics that are currently being collected. For example, if you have an unplugged VBD, `/vm_rrd` will return the historical data for that VBD from when it was in use, but `/rrd_updates` won't.
- The value of the `start` parameter is used to deduce the granularity of the data `/rrd_updates` will return. If you specify a shorter time period, you will get more detailed metrics. For example, if you specify a `start` value that is 9 minutes before "now", you'll get 108 rows from the 10 minute RRA, but if you specify 11 minutes before "now", you'll get 11 rows from the 2 hour RRA.
- Of particular importance is to use the same definition of "now" as the server is using, since the server is likely to be in UTC and your client may be in a different time zone.
- The data returned by each call to `/rrd_updates` will contain a value (in the `end` XML tag or JSON field) that can be used as the `start` parameter for the next call.

Additional `/rrd_updates` parameters

PARAMETER	VALUE	DESCRIPTION
`cf`	`average|min|max`	the data consolidation mode
`interval`	`<interval>`	the interval between values to be reported

NOTE:

By default only `average` metrics are available. To obtain `min` and `max` metrics for a VM, run the following command:

```
xe pool-param-set uuid=&lt;pool_uuid&gt; other-
config:create_min_max_in_new_VM_RRDs
```

Analysing RRDs

Once you have obtained the requested RRDs, you can parse and analyze the results.

This might be useful in if you want to perform simple tasks such as determining what the current `Network` throughput is, or perhaps carrying out your own, more complex analysis on the data points.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Some examples written in Python to demonstrate how you can parse and use the metrics can be found at XenServer Operational Metrics.

Example RRDs

Whole RRDs

Below is an example XML for a whole host RRD. The table highlights certain XML tags of interest.

XML TAG	DESCRIPTION
`ds`	A data source field.
`name`	The name of this data source.
`type`	Defines the data source type. Can be `GAUGE`, `COUNTER`, `DERIVE` or `ABSOLUTE`.
`minimal_heartbeat`	Defines the maximum number of seconds before a `ds` value is considered unknown.
`min`	The minimum acceptable value. Values less than this number are considered unknown. This is optional.
`max`	The maximum acceptable value. Values exceeding this number are considered unknown. This is optional.

```
<?xml version="1.0"?>

<rrd>

  <version>0003</version>

  <step>5</step>

  <lastupdate>1213616574</lastupdate>

  <ds>

    <name>memory_total_kib</name>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    <type>GAUGE</type>
    <minimal_heartbeat>300.0000</minimal_heartbeat>
    <min>0.0</min>
    <max>Infinity</max>
    <last_ds>2070172</last_ds>
    <value>9631315.6300</value>
    <unknown_sec>0</unknown_sec>
  </ds>
  <ds>
   <!-- other data sources - the order of the data sources is important
        and defines the ordering of the columns in the archives below -->
  </ds>
  <rra>
    <cf>AVERAGE</cf>
    <pdp_per_row>1</pdp_per_row>
    <params>
      <xff>0.5000</xff>
    </params>
    <cdp_prep> <!-- This is for internal use -->
      <ds>
        <primary_value>0.0</primary_value>
        <secondary_value>0.0</secondary_value>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        <value>0.0</value>

        <unknown_datapoints>0</unknown_datapoints>

      </ds>

      ...other data sources - internal use only...

    </cdp_prep>

    <database>

     <row>

        <v>2070172.0000</v>  <!-- columns correspond to the DSs defined
above -->

        <v>1756408.0000</v>

        <v>0.0</v>

        <v>0.0</v>

        <v>732.2130</v>

        <v>0.0</v>

        <v>782.9186</v>

        <v>0.0</v>

        <v>647.0431</v>

        <v>0.0</v>

        <v>0.0001</v>

        <v>0.0268</v>

        <v>0.0100</v>

        <v>0.0</v>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        <v>615.1072</v>

     </row>

     ...

  </rra>

  ... other archives ...

</rrd>
```

Here is an example JSON for a whole host RRD:

```
{

    "version": "0003",

    "step": "5",

    "lastupdate": "1698240426.8141",

    "ds": [

        {

            "name": "memory_total_kib",

            "type": "GAUGE",

            "minimal_heartbeat": "300.0000",

            "min": "0.0",

            "max": "Infinity",

            "last_ds": "33371836",

            "value": "60540111.1548",

            "unknown_sec": "0"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        }
        ... other data sources...
    ],
    "rra": [
        {
            "cf": "AVERAGE",
            "pdp_per_row": "1",
            "params": {
                "xff": "0.5000"
            },
            "cdp_prep": {
                "ds": [
                    {
                        "primary_value": "0.00",
                        "secondary_value": "0.00",
                        "value": "0.0",
                        "unknown_datapoints": "0.00"
                    }
                    ... other data sources - internal use only...
                ]
            },
            "database": [
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
                [
                    "0.0084",
                    "0.0090",
                    "0.0113",
                    "0.0072",
                    "0.0104",
                    "0.0099",
                    "0.0112",
                    "0.0131",
                    "0.0101",
                    "0.0100",
                    ... other values for the first data source...
                ],
                ... other data sources...
            ]
        },
        ... other archives...
    ]
}
```

Note that a VM RRD is identically structured, but with different data sources.

RRD updates

Example `rrd_updates` XML for one VM and no host updates:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
<xport>

  <meta>

    <start>1213578000</start>

    <step>3600</step>

    <end>1213617600</end>

    <rows>12</rows>

    <columns>12</columns>

    <legend>

      <entry>AVERAGE:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu1</entry>
<!-- nb - each data source might have multiple entries for different
consolidation functions -->

      <entry>AVERAGE:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu0</entry>

      <entry>AVERAGE:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:memory</
entry>

      <entry>MIN:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu1</entry>

      <entry>MIN:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu0</entry>

      <entry>MIN:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:memory</entry>

      <entry>MAX:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu1</entry>

      <entry>MAX:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu0</entry>

      <entry>MAX:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:memory</entry>

      <entry>LAST:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu1</entry>

      <entry>LAST:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:cpu0</entry>

      <entry>LAST:vm:ecd8d7a0-1be3-4d91-bd0e-4888c0e30ab3:memory</entry>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    </legend>
  </meta>
  <data>
    <row>
      <t>1213617600</t>
      <v>0.0</v> <!-- once again, the order or the columns is defined by
the legend above -->
      <v>0.0282</v>
      <v>209715200.0000</v>
      <v>0.0</v>
      <v>0.0201</v>
      <v>209715200.0000</v>
      <v>0.0</v>
      <v>0.0445</v>
      <v>209715200.0000</v>
      <v>0.0</v>
      <v>0.0243</v>
      <v>209715200.0000</v>
    </row>
   ...
  </data>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
</xport>
```

Example `rrd_updates` JSON for one VM and no host updates:

```
{
    "meta": {
        "start": "1213660800",
        "step": "86400",
        "end": "1698192000",
        "rows": "366",
        "columns": "101",
        "legend": [
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu0",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu1",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu2",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu3",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu4",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu5",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu6",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:cpu7",
            "AVERAGE:vm:b36c4883-7dca-41c6-9f35-ec91826ec29f:memory",
            ...
        ]
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    },
    "data": [
        {
            "t": "1698192000",
            "values": [
                "0.0082",
                "0.0080",
                "0.0072",
                "0.0072",
                "0.0069",
                "0.0073",
                "0.0069",
                "0.0076",
                "2724740352.0000",
                "NaN",
                "NaN",
                ...
            ]
        },
        ...
    ]
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
}
```

XenServer® Management API extensions

The Management API is a general and comprehensive interface to managing the life-cycles of virtual machines, and offers a lot of flexibility in the way that Management API providers may implement specific functionality (for example, storage provisioning, or console handling). XenServer has several extensions which provide useful functionality used in our own XenCenter® interface. The workings of these mechanisms are described in this chapter.

Extensions to the Management API are often provided by specifying `other-config` map keys to various objects. The use of this parameter indicates that the functionality is supported for that particular release of XenServer, but *not* as a long-term feature. We are constantly evaluating promoting functionality into the API, but this requires the nature of the interface to be well-understood. Developer feedback as to how you are using some of these extensions is always welcome to help us make these decisions.

VM console forwarding

Most Management API graphical interfaces will want to gain access to the VM consoles, in order to render them to the user as if they were physical machines. There are several types of consoles available, depending on the type of guest or if the physical host console is being accessed:

Console access

OPERATING SYSTEM	TEXT	GRAPHICAL	OPTIMIZED GRAPHICAL
Windows	No	VNC, using an API call	RDP, directly from guest
Linux	Yes, through VNC and an API call	No	VNC, directly from guest
Physical Host	Yes, through VNC and an API call	No	No

Hardware-assisted VMs, such as Windows, directly provide a graphical console over VNC. There is no text-based console, and guest networking is not necessary to use the graphical console.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Once guest networking has been established, it is more efficient to setup Remote Desktop Access and use an RDP client to connect directly (this must be done outside of the Management API).

Paravirtual VMs, such as Linux guests, provide a native text console directly. XenServer provides a utility (called `vncterm`) to convert this text-based console into a graphical VNC representation. Guest networking is not necessary for this console to function. As with Windows above, Linux distributions often configure VNC within the guest, and directly connect to it over a guest network interface.

The physical host console is only available as a `vt100` console, which is exposed through the Management API as a VNC console by using `vncterm` in the control domain.

RFB (Remote Framebuffer) is the protocol which underlies VNC, specified in The RFB Protocol. Third-party developers are expected to provide their own VNC viewers, and many freely available implementations can be adapted for this purpose. RFB 3.3 is the minimum version which viewers must support.

Retrieving VNC consoles using the API

VNC consoles are retrieved using a special URL passed through to the host agent. The sequence of API calls is as follows:

1. Client to Master/443: RPC: `Session.login_with_password()`.
2. Master/443 to Client: Returns a session reference to be used with subsequent calls.
3. Client to Master/443: RPC: `VM.get_by_name_label()`.
4. Master/443 to Client: Returns a reference to a particular VM (or the "control domain" if you want to retrieve the physical host console).
5. Client to Master/443: RPC: `VM.get_consoles()`.
6. Master/443 to Client: Returns a list of console objects associated with the VM.
7. Client to Master/443: RPC: `VM.get_location()`.
8. Returns a URI describing where the requested console is located. The URIs are of the form: `https://192.168.0.1/console?ref=OpaqueRef:c038533a-af99-a0ff-9095-c1159f2dc6a0` or `https://192.168.0.1/console?uuid=026e34fe-f0f2-20ee-5344-46d1aa922d5b`
9. Client to 192.168.0.1: HTTP CONNECT "/console?ref=(...)". You will also need to pass in "session_id=<session reference>" as a cookie.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The final HTTP CONNECT is slightly non-standard since the HTTP/1.1 RFC specifies that it must only be a host and a port, rather than a URL. Once the HTTP connect is complete, the connection can subsequently directly be used as a VNC server without any further HTTP protocol action.

This scheme requires direct access from the client to the control domain's IP, and will not work correctly if there are Network Address Translation (NAT) devices blocking such connectivity. You can use the CLI to retrieve the console URI from the client and perform a connectivity check.

Retrieve the VM UUID by running:

```
xe vm-list params=uuid --minimal name-label=name
```

Retrieve the console information:

```
xe console-list vm-uuid=uuid

uuid ( RO): 714f388b-31ed-67cb-617b-0276e35155ef

vm-uuid ( RO): 8acb7723-a5f0-5fc5-cd53-9f1e3a7d3069

vm-name-label ( RO): etch

protocol ( RO): RFB

location ( RO): https://192.168.0.1/console?ref=(...)
```

Use command-line utilities like `ping` to test connectivity to the IP address provided in the `location` field.

Disabling VNC forwarding for Linux VM

When creating and destroying Linux VMs, the host agent automatically manages the `vncterm` processes which convert the text console into VNC. Advanced users who want to directly access the text console can disable VNC forwarding for that VM. The text console can then only be accessed directly from the control domain directly, and graphical interfaces such as XenCenter will not be able to render a console for that VM.

Before starting the guest, set the following parameter on the VM record:

```
xe vm-param-set uuid=uuid other-config:disable_pv_vnc=1
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Start the VM.

Use the CLI to retrieve the underlying domain ID of the VM with:

```
xe vm-list params=dom-id uuid=uuid --minimal
```

On the host console, connect to the text console directly by:

```
/usr/lib/xen/bin/xenconsole domain_id
```

This configuration is an advanced procedure, and we do not recommend that the text console is directly used for heavy I/O operations. Instead, connect to the guest over SSH or some other network-based connection mechanism.

Adding xenstore entries to VMs

Developers might want to install guest agents into VMs which take special action based on the type of the VM. In order to communicate this information into the guest, a special xenstore name-space known as `vm-data` is available which is populated at VM creation time. It is populated from the `xenstore-data` map in the VM record.

Set the `xenstore-data` parameter in the VM record:

```
xe vm-param-set uuid=vm_uuid xenstore-data:vm-data/foo=bar
```

Start the VM.

If it is a Linux-based VM, install the XenServer VM Tools and use the command `xenstore-read` to verify that the node exists in xenstore.

NOTE:

Only prefixes beginning with `vm-data` are permitted, and anything not in this name-space will be silently ignored when starting the VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Security enhancements

The control domain in XenServer has various security enhancements in order to harden it against attack from malicious guests. These changes do not result in any loss of correct functionality, but the changes are documented here as variations of behavior from other distributions.

- The socket interface, `xenstored`, access using `libxenstore`. Interfaces are restricted by `xs_restrict()`.
- The device `/dev/xen/evtchn`, which is accessed by calling `xs_evtchn_open()` in `libxenctrl`. A handle can be restricted using `xs_evtchn_restrict()`.
- The device `/proc/xen/privcmd`, accessed through `xs_interface_open()` in `libxenctrl`. A handle is restricted using `xc_interface_restrict()`. Some privileged commands are naturally hard to restrict (for example, the ability to make arbitrary hypercalls), and these are simply prohibited on restricted handles.
- A restricted handle cannot later be granted more privilege, and so the interface must be closed and re-opened. Security is only gained if the process cannot subsequently open more handles.

The control domain privileged user-space interfaces can now be restricted to only work for certain domains. There are three interfaces affected by this change:

- The `qemu` device emulation processes and `vncterm` terminal emulation processes run as a non-root user ID and are restricted into an empty directory. They uses the restriction API above to drop privileges where possible.
- Access to xenstore is rate-limited to prevent malicious guests from causing a denial of service on the control domain. This is implemented as a token bucket with a restricted fill-rate, where most operations take one token and opening a transaction takes 20. The limits are set high enough that they are never hit when running even a large number of concurrent guests under loaded operation.
- The VNC guest consoles are bound only to the `localhost` interface, so that they are not exposed externally even if the control domain packet filter is disabled by user intervention.

Advanced settings for network interfaces

Virtual and physical network interfaces have some advanced settings that can be configured using the `other-config` map parameter. There is a set of custom ethtool settings and some miscellaneous settings.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ethtool settings

Developers might want to configure custom ethtool settings for physical and virtual network interfaces. This is accomplished with `ethtool-<option>` keys in the `other-config` map parameter.

KEY	DESCRIPTION	VALID SETTINGS
ethtool-rx	Specify if RX checksumming is enabled	`on` or `true` to enable the setting, `off` or `false` to disable it
ethtool-tx	Specify if TX checksumming is enabled	`on` or `true` to enable the setting, `off` or `false` to disable it
ethtool-sg	Specify if scatter-gather is enabled	`on` or `true` to enable the setting, `off` or `false` to disable it
ethtool-tso	Specify if tcp segmentation offload is enabled	`on` or `true` to enable the setting, `off` or `false` to disable it
ethtool-ufo	Specify if UDP fragmentation offload is enabled	`on` or `true` to enable the setting, `off` or `false` to disable it
ethtool-gso	Specify if generic segmentation offload is enabled	`on` or `true` to enable the setting, `off` or `false` to disable it
ethtool-autoneg	Specify if autonegotiation is enabled	`on` or `true` to enable the setting, `off` or `false` to disable it
ethtool-speed	Set the device speed in Mb/s	10, 100, or 1000
ethtool-duplex	Set full or half duplex mode	half or full

For example, to enable TX checksumming on a virtual NIC using the xe CLI:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
xe vif-param-set uuid=&lt;VIF UUID> other-config:ethtool-tx="on"
```

or:

```
xe vif-param-set uuid=&lt;VIF UUID> other-config:ethtool-tx="true"
```

To set the duplex setting on a physical NIC to half duplex using the xe CLI:

```
xe vif-param-set uuid=&lt;VIF UUID> other-config:ethtool-duplex="half"
```

Miscellaneous settings

You can also set a promiscuous mode on a VIF or PIF by setting the `promiscuous` key to `on`. For example, to enable promiscuous mode on a physical NIC using the xe CLI:

```
xe pif-param-set uuid=&lt;PIF UUID> other-config:promiscuous="on"
```

or:

```
xe pif-param-set uuid=&lt;PIF UUID> other-config:promiscuous="true"
```

The VIF and PIF objects have a `MTU` parameter that is read-only and provide the current setting of the maximum transmission unit for the interface. You can override the default maximum transmission unit of a physical or virtual NIC with the `mtu` key in the `other-config` map parameter. For example, to reset the MTU on a virtual NIC to use jumbo frames using the xe CLI:

```
xe vif-param-set uuid=&lt;VIF UUID> other-config:mtu=9000
```

Note that changing the MTU of underlying interfaces is an advanced and experimental feature, and may lead to unexpected side-effects if you have varying MTUs across NICs in a single resource pool.

Internationalization for SR names

The SRs created at install time now have an `other_config` key indicating how their names may be internationalized.

`other_config["i18n-key"]` may be one of

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `local-hotplug-cd`
- `local-hotplug-disk`
- `local-storage`
- `XenServer-tools`

Additionally, `other_config["i18n-original-value-<field name>"]` gives the value of that field when the SR was created. If XenCenter sees a record where `SR.name_label` equals `other_config["i18n-original-value-name_label"]` (that is, the record has not changed since it was created during XenServer installation), then internationalization will be applied. In other words, XenCenter will disregard the current contents of that field, and instead use a value appropriate to the user's own language.

If you change `SR.name_label` for your own purpose, then it no longer is the same as `other_config["i18n-original-value-name_label"]`. Therefore, XenCenter does not apply internationalization, and instead preserves your given name.

Hiding objects from XenCenter

Networks, PIFs, and VMs can be hidden from XenCenter by adding the key `HideFromXenCenter=true` to the `other_config` parameter for the object. This capability is intended for ISVs who know what they are doing, not general use by everyday users. For example, you might want to hide certain VMs because they are cloned VMs that are not intended to be used directly by general users in your environment.

In XenCenter, hidden Networks, PIFs, and VMs can be made visible, using the View menu.

Other extensions

The following section details the assumptions and API extensions that we have made, over and above the documented API. Extensions are encoded as particular key-value pairs in dictionaries such as `VM.other_config`.

Pool

KEY	SEMANTICS
pool.other_config["rolling_upgrade_in_progress"]	Present if the pool is in the middle of a rolling upgrade.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Host

KEY	SEMANTICS
host.other_config["iscsi_iqn"]	The host's iSCSI IQN.
host.license_params["expiry"]	The expiry date of the host's license, in ISO 8601, UTC.
host.license_params["sku_type"]	The host license type i.e. Server or Enterprise.
host.license_params["restrict_pooling"]	Returns `true` if pooling is restricted by the host.
host.license_params["restrict_connection"]	The number of connections that can be made from XenCenter is restricted.
host.license_params["restrict_qos"]	Returns `true` if Quality of Service settings are enabled on the host.
host.license_params["restrict_vlan"]	Returns `true` if creation of virtual networks is restricted on the host.
host.license_params["restrict_pool_attached_storage"]	Returns `true` if the creation of shared storage is restricted on this host.
host.software_version["product_version"]	Returns the host's product version.
host.software_version["build_number"]	Returns the host's build number.
host.software_version["xapi"]	Returns the host's api revision number.
host.software_version["package-linux"]	Returns "installed" if the Linux pack has been installed.
host.software_version["oem_build_number"]	If the host is the OEM version, return its revision number.
host.logging["syslog_destination"]	Gets or sets the destination for the XenServer system logger (null for local logging).
host.logging["multipathing"]	"true" if storage multipathing is enabled on this host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

KEY	SEMANTICS
host.logging["boot_time"]	A floating point Unix time giving the time that the host booted.
host.logging["agent_start_time"]	A floating point Unix time giving the time that the control domain management daemon started.

VM

KEY	SEMANTICS
VM.other_config["default_template"]	This template is one that was installed by . This is used to selectively hide these in the tree view, to use a different icon for them, and to disallow deletion.
VM.other_config["xensource_internal"]	This template is special, such as the P2V server template. These are completely hidden by the UI.
VM.other_config["install_distro"] == "rhlike"	This template is for RHEL 5, or CentOS equivalents. This is used to prompt for the Install Repository during install, including support for install from ISO / CD, and to modify NFS URLs to suit these installers.
VM.other_config["install-repository"] == "cdrom"	Requests an install from a repository in the VM's attached CD drive, rather than a URL.
VM.other_config["auto_poweron"]	Gets or sets whether the VM starts when the server boots, "true" or "false".
VM.other_config["ignore_excessive_vcpus"]	Gets or sets to ignore XenCenter warning if a VM has more VCPUs than its host has physical CPUs, `true` to ignore.
VM.other_config["HideFromXenCenter"]	Gets or sets whether XenCenter will show the VM in the treeview, "true" to hide.
VM.other_config["import_task"]	Gets the import task that created this VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

KEY	SEMANTICS
VM.HVM_boot_params["order"]	Gets or sets the VM's boot order on HVM VM's only, for example, "CDN" will boot in the following order - First boot disk, CD drive, Network.
VM.VCPU_params["weight"]	Gets or sets the IONice value for the VM's VCPUs, ranges from 1 to 65536, 65536 being the highest.
VM.pool_migrate(..., options['live'])	`true` indicates live migration. XenCenter always uses this.
VM.other_config["install-methods"]	A comma-separated list of install methods available for this template. Can include "cdrom", "nfs", "http" or "ftp".
VM.other_config["last_shutdown_time"]	The time that this VM was last shut down or rebooted, formatted as a UTC ISO8601 datetime.
VM.other_config["p2v_source_machine"]	The source machine, if this VM was imported by a P2V process.
VM.other_config["p2v_import_date"]	The date the VM was imported, if it was imported by a P2V process. Formatted as a UTC ISO8601 datetime.

SR

KEY	SEMANTICS
SR.other_config["auto-scan"]	The SR will be automatically scanned for changes. Set on all SRs created by XenCenter.
SR.sm_config["type"]	Set as type `cd` for SRs which are physical CD drives.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDI

KEY	SEMANTICS
VDI.sm_config["vmhint"]	The UUID of the VM that this VDI supports. This is set when VDIs are created through the user interface, to improve performance for certain storage backends.

VBD

KEY	SEMANTICS
VBD.other_config["owner"]	If set, then this disk may be deleted when the VM is uninstalled.
VBD.other_config["class"]	Set to an integer, corresponding to the Best Effort setting of `ionice`.

Network

KEY	SEMANTICS
network.other_config["automatic"]	The New VM wizard will create a VIF connected to this network by default, if this key has any value other than `false`.
network.other_config["import_task"]	Gets the import task that created this network.

VM_guest_metrics

KEY	SEMANTICS
PV_drivers_version["major"]	Gets the major version of the XenServer VM Tools on the VM.
PV_drivers_version["minor"]	Gets the minor version of the XenServer VM Tools on the VM.
PV_drivers_version["micro"]	Gets the micro (build number) of the XenServer VM Tools on the VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Task

KEY	SEMANTICS
task.other_config["object_creation"] == "complete"	For the task associated with a VM import, this flag will be set when all the objects (VMs, networks) have been created. This is useful in the import VM wizard for us to then go and re-map all the networks that need it.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Management API reference

This section provides the API reference information for all supported versions of XenServer®:

- XenServer 8.4

API overview

The API has the following key features:

- **Management of all aspects of the XenServer host.**

  The API allows you to manage VMs, storage, networking, host configuration, and pools. Performance and status metrics can also be queried from the API.

- **Persistent Object Model.**

  The results of all side-effecting operations (for example: object creation, deletion, and parameter changes) are persisted in a server-side database that is managed by XenServer.

- **An event mechanism.**

  Through the API, clients can register to be notified when persistent (server-side) objects are changed. This enables applications to track datamodel changes performed by concurrently running clients.

- **Synchronous and asynchronous invocation.**

  All API calls can be invoked synchronously (that is, block until completion). Any API call that might be long-running can also be invoked *asynchronously*. Asynchronous calls return immediately with a reference to a *task* object. This task object can be queried (through the API) for progress and status information. When an asynchronously invoked operation completes, the result (or errorcode) is available from the task object.

- **Remotable and Cross-Platform.**

  The client issuing the API calls doesn't have to be resident on the host being managed. The client also does not have to be connected to the host over SSH to run the API. API calls use the RPC protocol to transmit requests and responses over the network.

- **Secure and Authenticated Access.**

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The RPC API backend running on the host accepts secure socket connections. This allows a client to run the APIs over the HTTPS protocol. Further, all the API calls run in the context of a login session generated through user name and password validation at the server. This provides secure and authenticated access to the XenServer installation.

XenServer Management API Deprecation Policy

Items that will be removed in a future release are marked as deprecated.

By default, deprecated APIs and product functionality continue to be supported up to and including the next XenServer Long Term Service Release (LTSR). Deprecated items are usually removed in releases following that LTSR.

In exceptional cases, an item might be deprecated and removed before the next LTSR. For example, a change might be required to improve security. If this happens, customers are made aware of the change to the API or the product functionality.

This deprecation policy applies only to APIs and functionality that are documented at the following locations:

- This developer documentation
- XenServer Product Documentation

XenServer 9 Management API

Version **26.16.1**

QUICK LINKS:

- Classes
- Relationships Between Classes
- Types
- ErrorHandling

This document defines the XenServer® Management API - an interface for remotely configuring and controlling virtualized guests running on a Xen-enabled host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The API is presented here as a set of Remote Procedure Calls (RPCs). There are two supported wire formats, one based upon XML-RPC and one based upon JSON-RPC (v1.0 and v2.0 are both recognized). No specific language bindings are prescribed, although examples are given in the Python programming language.

Although we adopt some terminology from object-oriented programming, future client language bindings may or may not be object-oriented. The API reference uses the terminology *classes* and *objects*. For our purposes a *class* is simply a hierarchical namespace; an *object* is an instance of a class with its fields set to specific values. Objects are persistent and exist on the server-side. Clients may obtain opaque references to these server-side objects and then access their fields via get/set RPCs.

For each class we specify a list of fields along with their *types* and *qualifiers*. A qualifier is one of:

- *RO/runtime*: the field is Read Only. Furthermore, its value is automatically computed at runtime. For example, current CPU load and disk IO throughput.
- *RO/constructor*: the field must be manually set when a new object is created, but is then Read Only for the duration of the object's life. For example, the maximum memory addressable by a guest is set before the guest boots.
- *RW*: the field is Read/Write. For example, the name of a VM.

Types

The following types are used to specify methods and fields in the API Reference:

- `string`: Text strings.
- `int`: 64-bit integers.
- `float`: IEEE double-precision floating-point numbers.
- `bool`: Boolean.
- `datetime`: Date and timestamp.
- `c ref`: Reference to an object of class `c`.
- `t set`: Arbitrary-length set of values of type `t`.
- `(k -> v) map`: Mapping from values of type `k` to values of type `v`.
- `e enum`: Enumeration type with name `e`. Enums are defined in the API reference together with classes that use them.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Note that there are a number of cases where `ref`s are *doubly linked*. For example, a `VM` has a field called `VIFs` of type `VIF ref set`; this field lists the network interfaces attached to a particular VM. Similarly, the `VIF` class has a field called `VM` of type `VM ref` which references the VM to which the interface is connected. These two fields are *bound together*, in the sense that creating a new VIF causes the `VIFs` field of the corresponding VM object to be updated automatically.

The API reference lists explicitly the fields that are bound together in this way. It also contains a diagram that shows relationships between classes. In this diagram an edge signifies the existence of a pair of fields that are bound together, using standard crows-foot notation to signify the type of relationship (for example, one-many, many-many).

RPCs associated with fields

Each field, `f`, has an RPC accessor associated with it that returns `f`'s value:

- `get_f (r)`: takes a `ref`, `r` that refers to an object and returns the value of `f`.

Each field, `f`, with qualifier *RW* and whose outermost type is `set` has the following additional RPCs associated with it:

- `add_f(r, v)`: adds a new element `v` to the set. Note that sets cannot contain duplicate values, hence this operation has no action in the case that `v` is already in the set.
- `remove_f(r, v)`: removes element `v` from the set.

Each field, `f`, with qualifier *RW* and whose outermost type is `map` has the following additional RPCs associated with it:

- `add_to_f(r, k, v)`: adds new pair `k -> v` to the mapping stored in `f` in object `r`. Attempting to add a new pair for duplicate key, `k`, fails with a `MAP_DUPLICATE_KEY` error.
- `remove_from_f(r, k)`: removes the pair with key `k` from the mapping stored in `f` in object `r`.

Each field whose outermost type is neither `set` nor `map`, but whose qualifier is *RW* has an RPC accessor associated with it that sets its value:

- `set_f(r, v)`: sets the field `f` on object `r` to value `v`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with classes

- Most classes have a *constructor* RPC named `create` that takes as parameters all fields marked *RW* and *RO/constructor*. The result of this RPC is that a new *persistent* object is created on the server-side with the specified field values.
- Each class has a `get_by_uuid(uuid)` RPC that returns the object of that class that has the specified `uuid`.
- Each class that has a `name_label` field has a `get_by_name_label(name_label)` RPC that returns a set of objects of that class that have the specified `name_label`.
- Most classes have a `destroy(r)` RPC that explicitly deletes the persistent object specified by `r` from the system. This is a non-cascading delete - if the object being removed is referenced by another object then the `destroy` call will fail.

Apart from the RPCs enumerated above, most classes have additional RPCs associated with them. For example, the `VM` class has RPCs for cloning, suspending, starting etc. Such additional RPCs are described explicitly in the API reference.

Wire Protocol for Remote API Calls

API calls are sent over a network to a Xen-enabled host using an RPC protocol. Here we describe how the higher-level types used in our API Reference are mapped to primitive RPC types, covering the two supported wire formats XML-RPC and JSON-RPC.

XML-RPC Protocol

We specify the signatures of API functions in the following style:

```
(VM ref set)  VM.get_all()
```

This specifies that the function with name `VM.get_all` takes no parameters and returns a `set` of `VM ref`. These types are mapped onto XML-RPC types in a straight-forward manner:

- the types `float`, `bool`, `datetime`, and `string` map directly to the XML-RPC `<double>`, `<boolean>`, `<dateTime.iso8601>`, and `<string>` elements.
- all `ref` types are opaque references, encoded as the XML-RPC's `<string>` type. Users of the API should not make assumptions about the concrete form of these strings and should not expect them to remain valid after the client's session with the server has terminated.
- fields named `uuid` of type `string` are mapped to the XML-RPC `<string>` type. The string itself is the OSF DCE UUID presentation format (as output by `uuidgen`).

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `int` is assumed to be 64-bit in our API and is encoded as a string of decimal digits (rather than using XML-RPC's built-in 32-bit `<i4>` type).
- values of `enum` types are encoded as strings. For example, the value `destroy` of `enum on_normal_exit`, would be conveyed as:

```
<value><string>destroy</string></value>
```

- for all our types, `t`, our type `t set` simply maps to XML-RPC's `<array>` type, so, for example, a value of type `string set` would be transmitted like this:

```
<array>

  <data>

    <value><string>CX8</string></value>

    <value><string>PSE36</string></value>

    <value><string>FPU</string></value>

  </data>

</array>
```

- for types `k` and `v`, our type `(k -> v) map` maps onto an XML-RPC `<struct>`, with the key as the name of the struct. Note that the `(k -> v) map` type is only valid when `k` is a `string`, `ref`, or `int`, and in each case the keys of the maps are stringified as above. For example, the `(string -> float) map` containing the mappings *Mike > 2.3* and *John > 1.2* would be represented as:

```
<value>

  <struct>

    <member>

      <name>Mike</name>

      <value><double>2.3</double></value>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    </member>
    <member>
      <name>John</name>
      <value><double>1.2</double></value>
    </member>
  </struct>
</value>
```

- our `void` type is transmitted as an empty string.

XML-RPC Return Values and Status Codes

The return value of an RPC call is an XML-RPC `<struct>`.

- The first element of the struct is named `Status`; it contains a string value indicating whether the result of the call was a `Success` or a `Failure`.

If the `Status` is `Success` then the struct contains a second element named `Value`:

- The element of the struct named `Value` contains the function's return value.

If the `Status` is `Failure` then the struct contains a second element named `ErrorDescription`:

- The element of the struct named `ErrorDescription` contains an array of string values. The first element of the array is an error code; the rest of the elements are strings representing error parameters relating to that code.

For example, an XML-RPC return value from the `host.get_resident_VMs` function may look like this:

```
<struct>
   <member>
     <name>Status</name>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
      <value>Success</value>
    </member>
    <member>
       <name>Value</name>
       <value>
         <array>
            <data>
              <value>81547a35-205c-a551-c577-00b982c5fe00</value>
              <value>61c85a22-05da-b8a2-2e55-06b0847da503</value>
              <value>1d401ec4-3c17-35a6-fc79-cee6bd9811fe</value>
            </data>
         </array>
      </value>
    </member>
</struct>
```

JSON-RPC Protocol

We specify the signatures of API functions in the following style:

```
(VM ref set)  VM.get_all()
```

This specifies that the function with name `VM.get_all` takes no parameters and returns a `set` of `VM ref`. These types are mapped onto JSON-RPC types in the following manner:

- the types `float` and `bool` map directly to the JSON types `number` and `boolean`, while `datetime` and `string` are represented as the JSON `string` type.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- all `ref` types are opaque references, encoded as the JSON `string` type. Users of the API should not make assumptions about the concrete form of these strings and should not expect them to remain valid after the client's session with the server has terminated.
- fields named `uuid` of type `string` are mapped to the JSON `string` type. The string itself is the OSF DCE UUID presentation format (as output by `uuidgen`).
- `int` is assumed to be 64-bit in our API and is encoded as a JSON `number` without decimal point or exponent, preserved as a string.
- values of `enum` types are encoded as the JSON `string` type. For example, the value `destroy` of `enum on_normal_exit`, would be conveyed as:

```
"destroy"
```

- for all our types, `t`, our type `t set` simply maps to the JSON `array` type, so, for example, a value of type `string set` would be transmitted like this:

```
[ "CX8", "PSE36", "FPU" ]
```

- for types `k` and `v`, our type `(k -> v) map` maps onto a JSON object which contains members with name `k` and value `v`. Note that the `(k -> v) map` type is only valid when `k` is a `string`, `ref`, or `int`, and in each case the keys of the maps are stringified as above. For example, the `(string -> float) map` containing the mappings *Mike > 2.3* and *John > 1.2* would be represented as:

```
{

  "Mike": 2.3,

  "John": 1.2

}
```

- our `void` type is transmitted as an empty string.

Both versions 1.0 and 2.0 of the JSON-RPC wire format are recognised and, depending on your client library, you can use either of them.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

JSON-RPC v1.0

JSON-RPC v1.0 requests

An API call is represented by sending a single JSON object to the server, which contains the members `method`, `params`, and `id`.

- `method`: A JSON `string` containing the name of the function to be invoked.
- `params`: A JSON `array` of values, which represents the parameters of the function to be invoked.
- `id`: A JSON `string` or `integer` representing the call id. Note that, diverging from the JSON-RPC v1.0 specification the API does not accept *notification* requests (requests without responses), i.e. the id cannot be `null`.

For example, the body of a JSON-RPC v1.0 request to retrieve the resident VMs of a host may look like this:

```
{
  "method": "host.get_resident_VMs",
  "params": [
    "OpaqueRef:74f1a19cd-b660-41e3-a163-10f03e0eae67",
    "OpaqueRef:08c34fc9-f418-4f09-8274-b9cb25cd8550"
  ],
  "id": "xyz"
}
```

In the above example, the first element of the `params` array is the reference of the open session to the host, while the second is the host reference.

JSON-RPC v1.0 return values

The return value of a JSON-RPC v1.0 call is a single JSON object containing the members `result`, `error`, and `id`.

- `result`: If the call is successful, it is a JSON value (`string`, `array` etc.) representing the return value of the invoked function. If an error has occurred, it is `null`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `error`: If the call is successful, it is `null`. If the call has failed, it a JSON `array` of `string` values. The first element of the array is an error code; the remainder of the array are strings representing error parameters relating to that code.
- `id`: The call id. It is a JSON `string` or `integer` and it is the same id as the request it is responding to.

For example, a JSON-RPC v1.0 return value from the `host.get_resident_VMs` function may look like this:

```
{

  "result": [

      "OpaqueRef:604f51e7-630f-4412-83fa-b11c6cf008ab",

      "OpaqueRef:670d08f5-cbeb-4336-8420-ccd56390a65f"

  ],

  "error": null,

  "id": "xyz"

}
```

while the return value of the same call made on a logged out session may look like this:

```
{

  "result": null,

  "error": [

      "SESSION_INVALID",

      "OpaqueRef:93f1a23cd-a640-41e3-b163-10f86e0eae67"

  ],

  "id": "xyz"
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
}
```

JSON-RPC v2.0

JSON-RPC V2.0 REQUESTS

An API call is represented by sending a single JSON object to the server, which contains the members `jsonrpc`, `method`, `params`, and `id`.

- `jsonrpc`: A JSON `string` specifying the version of the JSON-RPC protocol. It is exactly "2.0".
- `method`: A JSON `string` containing the name of the function to be invoked.
- `params`: A JSON `array` of values, which represents the parameters of the function to be invoked. Although the JSON-RPC v2.0 specification allows this member to be omitted, in practice all API calls accept at least one parameter.
- `id`: A JSON `string` or `integer` representing the call id. Note that, diverging from the JSON-RPC v2.0 specification it cannot be null. Neither can it be omitted because the API does not accept *notification* requests (requests without responses).

For example, the body of a JSON-RPC v2.0 request to retrieve the VMs resident on a host may look like this:

```
 {
   "jsonrpc": "2.0",
   "method": "host.get_resident_VMs",
   "params": [
     "OpaqueRef:c90cd28f-37ec-4dbf-88e6-f697ccb28b39",
     "OpaqueRef:08c34fc9-f418-4f09-8274-b9cb25cd8550"
   ],
   "id": 3
}
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

As before, the first element of the `parameter` array is the reference of the open session to the host, while the second is the host reference.

JSON-RPC v2.0 return values

The return value of a JSON-RPC v2.0 call is a single JSON object containing the members `jsonrpc`, either `result` or `error` depending on the outcome of the call, and `id`.

- `jsonrpc`: A JSON `string` specifying the version of the JSON-RPC protocol. It is exactly "2.0".
- `result`: If the call is successful, it is a JSON value (`string`, `array` etc.) representing the return value of the invoked function. If an error has occurred, it does not exist.
- `error`: If the call is successful, it does not exist. If the call has failed, it is a single structured JSON object (see below).
- `id`: The call id. It is a JSON `string` or `integer` and it is the same id as the request it is responding to.

The `error` object contains the members `code`, `message`, and `data`.

- `code`: The API does not make use of this member and only retains it for compliance with the JSON-RPC v2.0 specification. It is a JSON `integer` which has a non-zero value.
- `message`: A JSON `string` representing an API error code.
- `data`: A JSON array of `string` values representing error parameters relating to the aforementioned API error code.

For example, a JSON-RPC v2.0 return value from the `host.get_resident_VMs` function may look like this:

```
{

  "jsonrpc": "2.0",

  "result": [

      "OpaqueRef:604f51e7-630f-4412-83fa-b11c6cf008ab",

      "OpaqueRef:670d08f5-cbeb-4336-8420-ccd56390a65f"

  ],

  "id": 3
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
}
```

while the return value of the same call made on a logged out session may look like this:

```
{
  "jsonrpc": "2.0",
  "error": {
      "code": 1,
      "message": "SESSION_INVALID",
      "data": [
          "OpaqueRef:c90cd28f-37ec-4dbf-88e6-f697ccb28b39"
      ]
  },
  "id": 3
}
```

Errors

When a low-level transport error occurs, or a request is malformed at the HTTP or RPC level, the server may send an HTTP 500 error response, or the client may simulate the same. The client must be prepared to handle these errors, though they may be treated as fatal.

For example, the following malformed request when using the XML-RPC protocol:

```
$curl -D - -X POST https://server -H 'Content-Type: application/xml' \
  -d '<?xml version="1.0"?>
  <methodCall>
    <methodName>session.logout</methodName>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  </methodCall>'
```

results to the following response:

```
HTTP/1.1 500 Internal Error

content-length: 297

content-type:text/html

connection:close

cache-control:no-cache, no-store



<html><body><h1>HTTP 500 internal server error</h1>An unexpected error
occurred;

 please wait a while and try again. If the problem persists, please
contact your

 support representative.<h1> Additional information </
h1>Xmlrpc.Parse_error(&quo

t;close_tag&quot;, &quot;open_tag&quot;, _)</body></html>
```

When using the JSON-RPC protocol:

```
$curl -D - -X POST https://server/jsonrpc -H 'Content-Type: application/
json' \

  -d '{

      "jsonrpc": "2.0",

      "method": "session.login_with_password",

      "id": 0
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  }'
```

the response is:

```
HTTP/1.1 500 Internal Error

content-length: 308

content-type:text/html

connection:close

cache-control:no-cache, no-store



<html><body><h1>HTTP 500 internal server error</h1>An unexpected error
occurred;

 please wait a while and try again. If the problem persists, please
contact your

 support representative.<h1> Additional information </
h1>Jsonrpc.Malformed_metho

d_request(&amp;quot;{jsonrpc=...,method=...,id=...}&amp;quot;)</body></html>
```

All other failures are reported with a more structured error response, to allow better automatic response to failures, proper internationalization of any error message, and easier debugging.

On the wire, these are transmitted like this when using the XML-RPC protocol:

```
<struct>

    <member>

        <name>Status</name>

        <value>Failure</value>

    </member>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    <member>
        <name>ErrorDescription</name>
        <value>
            <array>
                <data>
                    <value>MAP_DUPLICATE_KEY</value>
                    <value>Customer</value>
                    <value>eSpiel Inc.</value>
                    <value>eSpiel Incorporated</value>
                </data>
            </array>
        </value>
    </member>
</struct>
```

Note that `ErrorDescription` value is an array of string values. The first element of the array is an error code; the remainder of the array are strings representing error parameters relating to that code. In this case, the client has attempted to add the mapping *Customer* > *eSpiel Incorporated* to a Map, but it already contains the mapping *Customer* > *eSpiel Inc.*, hence the request has failed.

When using the JSON-RPC protocol v2.0, the above error is transmitted as:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": 1,
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        "message": "MAP_DUPLICATE_KEY",
        "data": [
            "Customer",
            "eSpiel Inc.",
            "eSpiel Incorporated"
        ]
    },
    "id": 3
}
```

Finally, when using the JSON-RPC protocol v1.0:

```
{
    "result": null,
    "error": [
        "MAP_DUPLICATE_KEY",
        "Customer",
        "eSpiel Inc.",
        "eSpiel Incorporated"
    ],
    "id": "xyz"
}
```

Each possible error code is documented in the last section of the API reference.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Note on References vs UUIDs

References are opaque types - encoded as XML-RPC and JSON-RPC strings on the wire - understood only by the particular server which generated them. Servers are free to choose any concrete representation they find convenient; clients should not make any assumptions or attempt to parse the string contents. References are not guaranteed to be permanent identifiers for objects; clients should not assume that references generated during one session are valid for any future session. References do not allow objects to be compared for equality. Two references to the same object are not guaranteed to be textually identical.

UUIDs are intended to be permanent identifiers for objects. They are guaranteed to be in the OSF DCE UUID presentation format (as output by `uuidgen`). Clients may store UUIDs on disk and use them to look up objects in subsequent sessions with the server. Clients may also test equality on objects by comparing UUID strings.

The API provides mechanisms for translating between UUIDs and opaque references. Each class that contains a UUID field provides:

- A `get_by_uuid` method that takes a UUID and returns an opaque reference to the server-side object that has that UUID;
- A `get_uuid` function (a regular "field getter" RPC) that takes an opaque reference and returns the UUID of the server-side object that is referenced by it.

Making RPC Calls

Transport Layer

The following transport layers are currently supported:

- HTTP/HTTPS for remote administration
- HTTP over Unix domain sockets for local administration

Session Layer

The RPC interface is session-based; before you can make arbitrary RPC calls you must login and initiate a session. For example:

```
(session ref) session.login_with_password(string uname, string pwd,

                string version, string originator)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

where `uname` and `password` refer to your username and password, as defined by the Xen administrator, while `version` and `originator` are optional. The `session ref` returned by `session.login_with_password` is passed to subsequent RPC calls as an authentication token. Note that a session reference obtained by a login request to the XML-RPC backend can be used in subsequent requests to the JSON-RPC backend, and vice versa.

A session can be terminated with the `session.logout` function:

```
void  session.logout(session ref session_id)
```

Synchronous and Asynchronous Invocation

Each method call (apart from methods on the `Session` and `Task` objects and "getters" and "setters" derived from fields) can be made either synchronously or asynchronously. A synchronous RPC call blocks until the return value is received; the return value of a synchronous RPC call is exactly as specified above.

Only synchronous API calls are listed explicitly in this document. All their asynchronous counterparts are in the special `Async` namespace. For example, the synchronous call `VM.clone(...)` has an asynchronous counterpart, `Async.VM.clone(...)`, that is non-blocking.

Instead of returning its result directly, an asynchronous RPC call returns an identifier of type `task ref` which is subsequently used to track the status of a running asynchronous RPC.

Note that an asynchronous call may fail immediately, before a task has even been created. When using the XML-RPC wire protocol, this eventuality is represented by wrapping the returned `task ref` in an XML-RPC struct with a `Status`, `ErrorDescription`, and `Value` fields, exactly as specified above; the `task ref` is provided in the `Value` field if `Status` is set to `Success`. When using the JSON-RPC protocol, the `task ref` is wrapped in a response JSON object as specified above and it is provided by the value of the `result` member of a successful call.

The RPC call

```
(task ref set)  Task.get_all(session ref session_id)
```

returns a set of all task identifiers known to the system. The status (including any returned result and error codes) of these can then be queried by accessing the fields of the `Task` object in the usual way. Note that, in order to get a consistent snapshot of a task's state, it is advisable to call the `get_record` function.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Example interactive session

This section describes how an interactive session might look, using python XML-RPC and JSON-RPC client libraries.

First, initialise python:

```
$ python3

>>>
```

Using the XML-RPC Protocol

Import the library `xmlrpc.client` and create a python object referencing the remote server as shown below:

```
>>> import xmlrpc.client

>>> xen = xmlrpc.client.ServerProxy("https://localhost:443")
```

Note that you may need to disable SSL certificate validation to establish the connection, this can be done as follows:

```
>>> import ssl

>>> ctx = ssl._create_unverified_context()

>>> xen = xmlrpc.client.ServerProxy("https://localhost:443", context=ctx)
```

Acquire a session reference by logging in with a username and password; the session reference is returned under the key `Value` in the resulting dictionary (error-handling omitted for brevity):

```
>>> session = xen.session.login_with_password("user", "passwd",

...                                           "version", "originator")
['Value']
```

This is what the call looks like when serialized

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
<?xml version='1.0'?>

<methodCall>

    <methodName>session.login_with_password</methodName>

    <params>

        <param><value><string>user</string></value></param>

        <param><value><string>passwd</string></value></param>

        <param><value><string>version</string></value></param>

        <param><value><string>originator</string></value></param>

    </params>

</methodCall>
```

Next, the user may acquire a list of all the VMs known to the system (note the call takes the session reference as the only parameter):

```
>>> all_vms = xen.VM.get_all(session)['Value']

>>> all_vms

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]
```

The VM references here have the form `OpaqueRef:X` (though they may not be that simple in reality) and you should treat them as opaque strings. *Templates* are VMs with the `is_a_template` field set to `true`. We can find the subset of template VMs using a command like the following:

```
>>> all_templates = filter(lambda x: xen.VM.get_is_a_template(session, x)
['Value'],

                            all_vms)
```

Once a reference to a VM has been acquired, a lifecycle operation may be invoked:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
>>> xen.VM.start(session, all_templates[0], False, False)

{'Status': 'Failure', 'ErrorDescription': ['VM_IS_TEMPLATE',
'OpaqueRef:X']}
```

In this case the `start` message has been rejected, because the VM is a template, and so an error response has been returned. These high-level errors are returned as structured data (rather than as XML-RPC faults), allowing them to be internationalized.

Rather than querying fields individually, whole *records* may be returned at once. To retrieve the record of a single object as a python dictionary:

```
>>> record = xen.VM.get_record(session, all_templates[0])['Value']

>>> record['power_state']

'Halted'

>>> record['name_label']

'Windows 10 (64-bit)'
```

To retrieve all the VM records in a single call:

```
>>> records = xen.VM.get_all_records(session)['Value']

>>> list(records.keys())

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]

>>> records['OpaqueRef:1']['name_label']

'Red Hat Enterprise Linux 7'
```

Using the JSON-RPC Protocol

For this example we are making use of the package `jsonrpcclient` and the `requests` library due to their simplicity, although other packages can also be used.

First, import the `requests` and `jsonrpcclient` libraries:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
>>> import requests

>>> import jsonrpcclient
```

Now we construct a utility method to make using these libraries easier:

```
>>> def jsonrpccall(method, params):

...     r = requests.post("https://localhost:443/jsonrpc",

...                       json=jsonrpcclient.request(method,
params=params),

...                       verify=False)

...     p = jsonrpcclient.parse(r.json())

...     if isinstance(p, jsonrpcclient.Ok):

...         return p.result

...     raise Exception(p.message, p.data)
```

Acquire a session reference by logging in with a username and password:

```
>>> session = jsonrpccall("session.login_with_password",

...                       ("user", "password", "version", "originator"))
```

`jsonrpcclient` uses the JSON-RPC protocol v2.0, so this is what the serialized request looks like:

```
{

  "jsonrpc": "2.0",

  "method": "session.login_with_password",

  "params": ["user", "passwd", "version", "originator"],
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  "id": 0

}
```

Next, the user may acquire a list of all the VMs known to the system (note the call takes the session reference as the only parameter):

```
>>> all_vms = jsonrpccall("VM.get_all", (session,))

>>> all_vms

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]
```

The VM references here have the form `OpaqueRef:X` (though they may not be that simple in reality) and you should treat them as opaque strings. *Templates* are VMs with the `is_a_template` field set to `true`. We can find the subset of template VMs using a command like the following:

```
>>> all_templates = filter(

...     lambda x: jsonrpccall("VM.get_is_a_template", (session, x)),

...     all_vms)
```

Once a reference to a VM has been acquired, a lifecycle operation may be invoked:

```
>>> try:

...     jsonrpccall("VM.start", (session, next(all_templates), False,
False))

... except Exception as e:

...     e

...

Exception('VM_IS_TEMPLATE', ['OpaqueRef:1', 'start'])
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

In this case the `start` message has been rejected because the VM is a template, hence an error response has been returned. These high-level errors are returned as structured data, allowing them to be internationalized.

Rather than querying fields individually, whole *records* may be returned at once. To retrieve the record of a single object as a python dictionary:

```
>>> record = jsonrpccall("VM.get_record", (session, next(all_templates)))

>>> record['power_state']

'Halted'

>>> record['name_label']

'Windows 10 (64-bit)'
```

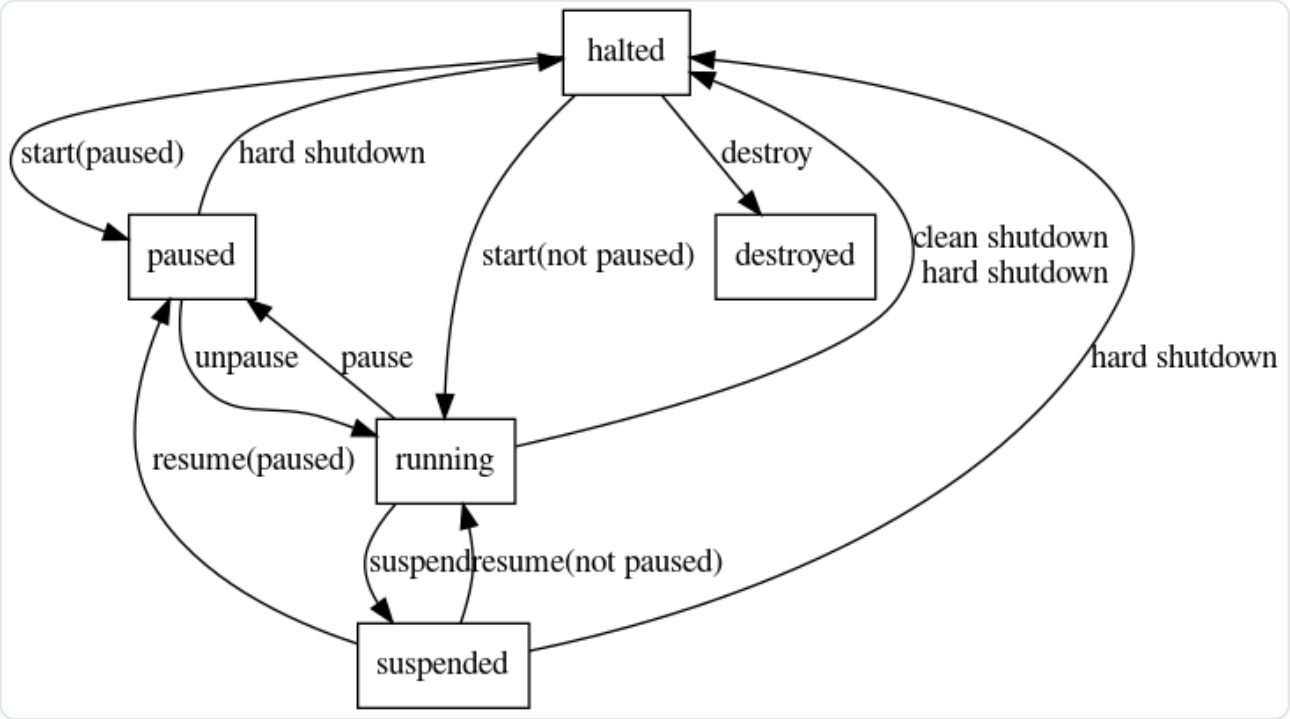
To retrieve all the VM records in a single call:

```
>>> records = jsonrpccall("VM.get_all_records", (session,))

>>> records.keys()

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]

>>> records['OpaqueRef:1']['name_label']

'Red Hat Enterprise Linux 7'
```

VM Lifecycle

The following figure shows the states that a VM can be in and the API calls that can be used to move the VM between these states.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



VM boot parameters

The `VM` class contains a number of fields that control the way in which the VM is booted. With reference to the fields defined in the VM class (see later in this document), this section outlines the boot options available and the mechanisms provided for controlling them.

VM booting is controlled by setting one of the two mutually exclusive groups: "PV" and "HVM". If `HVM.boot_policy` is an empty string, then paravirtual domain building and booting will be used; otherwise the VM will be loaded as a HVM domain, and booted using an emulated BIOS.

When paravirtual booting is in use, the `PV_bootloader` field indicates the bootloader to use. It may be "pygrub", in which case the platform's default installation of pygrub will be used, or a full path within the control domain to some other bootloader. The other fields, `PV_kernel`, `PV_ramdisk`, `PV_args`, and `PV_bootloader_args` will be passed to the bootloader unmodified, and interpretation of those fields is then specific to the bootloader itself, including the possibility that the bootloader will ignore some or all of those given values. Finally the paths of all bootable disks are added to the bootloader commandline (a disk is bootable if its VBD has the bootable flag set). There may be zero, one, or many bootable disks; the bootloader decides which disk (if any) to boot from.

If the bootloader is pygrub, then the menu.lst is parsed, if present in the guest's filesystem, otherwise the specified kernel and ramdisk are used, or an autodetected kernel is used if nothing is specified and autodetection is possible. `PV_args` is appended to the kernel command line, no matter which mechanism is used for finding the kernel.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If `PV_bootloader` is empty but `PV_kernel` is specified, then the kernel and ramdisk values will be treated as paths within the control domain. If both `PV_bootloader` and `PV_kernel` are empty, then the behaviour is as if `PV_bootloader` were specified as "pygrub".

When using HVM booting, `HVM_boot_policy` and `HVM_boot_params` specify the boot handling. Only one policy is currently defined, "BIOS order". In this case, `HVM_boot_params` should contain one key-value pair "order" = "N" where N is the string that will be passed to QEMU. Optionally `HVM_boot_params` can contain another key-value pair "firmware" with values "bios" or "uefi" (default is "bios" if absent). By default Secure Boot is not enabled, it can be enabled when "uefi" is enabled by setting `VM.platform["secureboot"]` to true.

Classes

The following classes are defined:

NAME	DESCRIPTION
auth	Management of remote authentication services
blob	A placeholder for a binary blob
Bond	A Network bond that combines physical network interfaces, also known as link aggregation
Certificate	An X509 certificate used for TLS connections
Cluster	Cluster-wide Cluster metadata
Cluster_host	Cluster member metadata
console	A console
crashdump	**Deprecated**. A VM crashdump
data_source	Data sources for logging in RRDs
DR_task	DR task
Driver_variant	UNSUPPORTED. Variant of a host driver
event	Asynchronous event registration and handling

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
Feature	A new piece of functionality
GPU_group	A group of compatible GPUs across the resource pool
host	A physical host
host_cpu	**Deprecated**. A physical CPU
host_crashdump	Represents a host crash dump
Host_driver	UNSUPPORTED. A multi-version driver on a host
host_metrics	The metrics associated with a host
host_patch	**Deprecated**. Represents a patch stored on a server
LVHD	LVHD SR specific operations
message	An message for the attention of the administrator
network	A virtual network
network_sriov	network-sriov which connects logical pif and physical pif
Observer	Describes an observer which will control observability activity in the Toolstack
PBD	The physical block devices through which hosts access SRs
PCI	A PCI device
PGPU	A physical GPU (pGPU)
PIF	A physical network interface (note separate VLANs are represented as several PIFs)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
PIF_metrics	The metrics associated with a physical network interface
pool	Pool-wide information
pool_patch	**Deprecated**. Pool-wide patches
pool_update	Pool-wide updates to the host software
probe_result	A set of properties that describe one result element of SR.probe. Result elements and properties can change dynamically based on changes to the the SR.probe input-parameters or the target.
PUSB	A physical USB device
PVS_cache_storage	Describes the storage that is available to a PVS site for caching purposes
PVS_proxy	a proxy connects a VM/VIF with a PVS site
PVS_server	individual machine serving provisioning (block) data
PVS_site	machines serving blocks of data for provisioning VMs
Repository	Repository for updates
role	A set of permissions associated with a subject
SDN_controller	Describes the SDN controller that is to connect with the pool
secret	A secret
session	A session
SM	A storage manager plugin
SR	A storage repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
sr_stat	A set of high-level properties associated with an SR.
subject	A user or group that can log in xapi
task	A long-running asynchronous task
tunnel	A tunnel for network traffic
USB_group	A group of compatible USBs across the resource pool
user	**Deprecated**. A user of the system
VBD	A virtual block device
VBD_metrics	**Removed**. The metrics associated with a virtual block device
VDI	A virtual disk image
vdi_nbd_server_info	Details for connecting to a VDI using the Network Block Device protocol
VGPU	A virtual GPU (vGPU)
VGPU_type	A type of virtual GPU
VIF	A virtual network interface
VIF_metrics	**Removed**. The metrics associated with a virtual network device
VLAN	A VLAN mux/demux
VM	A virtual machine (or 'guest').
VM_appliance	VM appliance
VM_group	A VM group

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
VM_guest_metrics	The metrics reported by the guest (as opposed to inferred from outside)
VM_metrics	The metrics associated with a VM
VMPP	**Removed**. VM Protection Policy
VMSS	VM Snapshot Schedule
VTPM	A virtual TPM device
VUSB	Describes the vusb device

Relationships Between Classes

Fields that are bound together are shown in the following table:

OBJECT.FIELD	*OBJECT.FIELD*	*RELATIONSHIP*
`VM.snapshot_of`	`VM.snapshots`	one-to-many
`VDI.snapshot_of`	`VDI.snapshots`	one-to-many
`VM.parent`	`VM.children`	one-to-many
`task.subtask_of`	`task.subtasks`	one-to-many
`PIF.bond_slave_of`	`Bond.slaves`	one-to-many
`Bond.master`	`PIF.bond_master_of`	one-to-many
`VLAN.tagged_PIF`	`PIF.VLAN_slave_of`	one-to-many
`tunnel.access_PIF`	`PIF.tunnel_access_PIF_of`	one-to-many
`tunnel.transport_PIF`	`PIF.tunnel_transport_PIF_of`	one-to-many
`PBD.host`	`host.PBDs`	one-to-many

© 1997–2026 Citrix Systems, Inc. All rights reserved.

OBJECT.FIELD	OBJECT.FIELD	RELATIONSHIP
`PBD.SR`	`SR.PBDs`	one-to-many
`VBD.VDI`	`VDI.VBDs`	one-to-many
`crashdump.VDI`	`VDI.crash_dumps`	one-to-many
`VBD.VM`	`VM.VBDs`	one-to-many
`crashdump.VM`	`VM.crash_dumps`	one-to-many
`VIF.VM`	`VM.VIFs`	one-to-many
`VIF.network`	`network.VIFs`	one-to-many
`Cluster_host.cluster`	`Cluster.cluster_hosts`	one-to-many
`PIF.host`	`host.PIFs`	one-to-many
`PIF.network`	`network.PIFs`	one-to-many
`VDI.SR`	`SR.VDIs`	one-to-many
`VTPM.VM`	`VM.VTPMs`	one-to-many
`console.VM`	`VM.consoles`	one-to-many
`VM.resident_on`	`host.resident_VMs`	one-to-many
`host_cpu.host`	`host.host_CPUs`	one-to-many
`host_crashdump.host`	`host.crashdumps`	one-to-many
`host_patch.host`	`host.patches`	one-to-many
`host_patch.pool_patch`	`pool_patch.host_patches`	one-to-many
`host.updates`	`pool_update.hosts`	many-to-many
`subject.roles`	`subject.roles`	unknown type
`role.subroles`	`role.subroles`	many-to-many

© 1997–2026 Citrix Systems, Inc. All rights reserved.

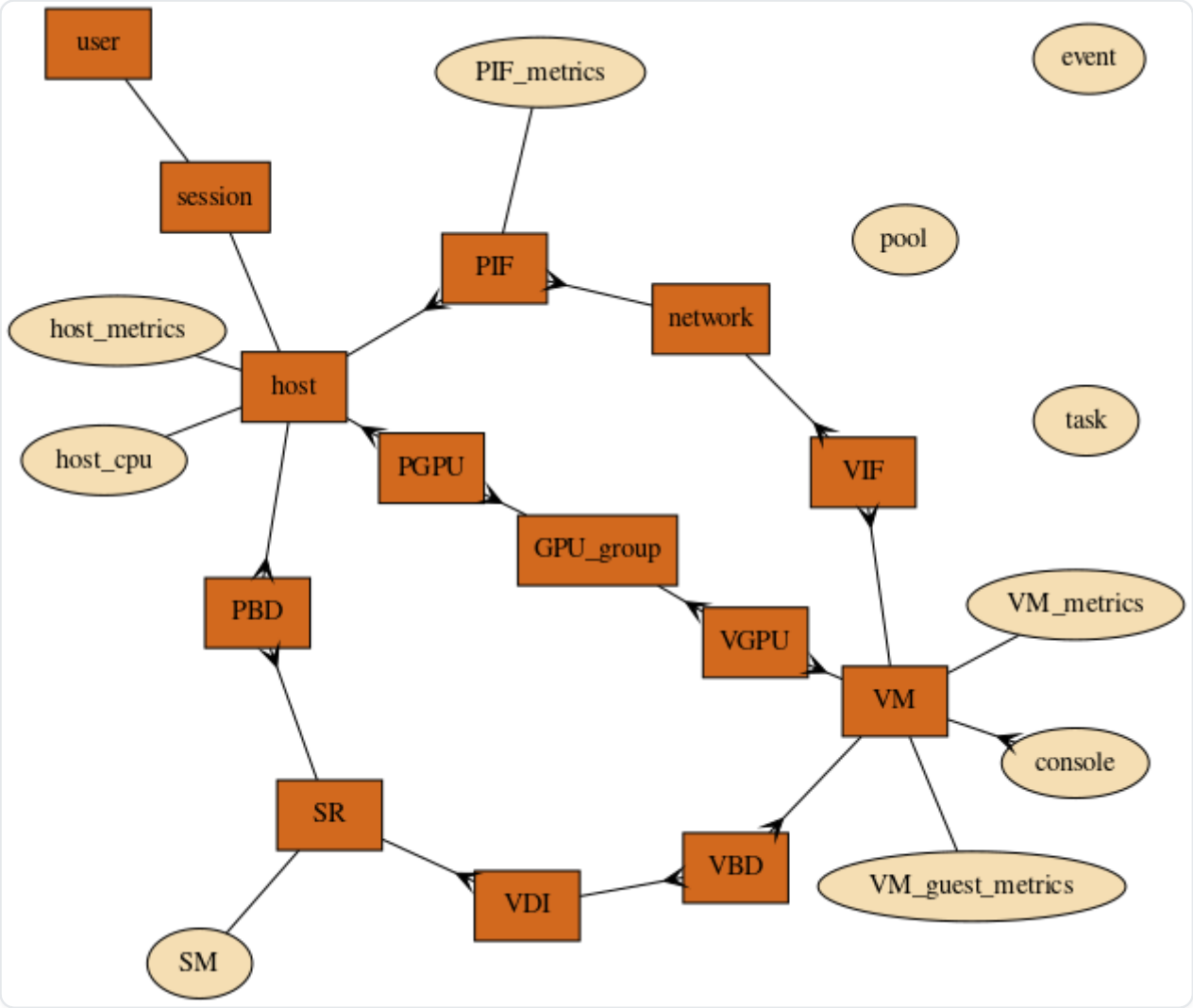
OBJECT.FIELD	OBJECT.FIELD	RELATIONSHIP
`VM.protection_policy`	`VMPP.VMs`	one-to-many
`VM.snapshot_schedule`	`VMSS.VMs`	one-to-many
`VM.appliance`	`VM_appliance.VMs`	one-to-many
`PGPU.GPU_group`	`GPU_group.PGPUs`	one-to-many
`VGPU.GPU_group`	`GPU_group.VGPUs`	one-to-many
`VGPU.type`	`VGPU_type.VGPUs`	one-to-many
`VGPU.VM`	`VM.VGPUs`	one-to-many
`VGPU.resident_on`	`PGPU.resident_VGPUs`	one-to-many
`PGPU.supported_VGPU_types`	`VGPU_type.supported_on_PGPUs`	many-to-many
`PGPU.enabled_VGPU_types`	`VGPU_type.enabled_on_PGPUs`	many-to-many
`GPU_group.supported_VGPU_types`	`VGPU_type.supported_on_GPU_groups`	many-to-many
`GPU_group.enabled_VGPU_types`	`VGPU_type.enabled_on_GPU_groups`	many-to-many
`PCI.host`	`host.PCIs`	one-to-many
`PGPU.host`	`host.PGPUs`	one-to-many
`VDI.metadata_of_pool`	`pool.metadata_VDIs`	one-to-many
`SR.introduced_by`	`DR_task.introduced_SRs`	one-to-many
`PVS_server.site`	`PVS_site.servers`	one-to-many
`PVS_proxy.site`	`PVS_site.proxies`	one-to-many
`PVS_cache_storage.site`	`PVS_site.cache_storage`	one-to-many

© 1997–2026 Citrix Systems, Inc. All rights reserved.

OBJECT.FIELD	*OBJECT.FIELD*	*RELATIONSHIP*
`PUSB.host`	`host.PUSBs`	one-to-many
`PUSB.USB_group`	`USB_group.PUSBs`	one-to-many
`VUSB.USB_group`	`USB_group.VUSBs`	one-to-many
`VUSB.VM`	`VM.VUSBs`	one-to-many
`Feature.host`	`host.features`	one-to-many
`network_sriov.physical_PIF`	`PIF.sriov_physical_PIF_of`	one-to-many
`network_sriov.logical_PIF`	`PIF.sriov_logical_PIF_of`	one-to-many
`Certificate.host`	`host.certificates`	one-to-many
`VM.groups`	`VM_group.VMs`	many-to-many
`Driver_variant.driver`	`Host_driver.variants`	one-to-many

The following figure represents bound fields (as specified above) using crow's foot notation to specify one-to-one, one-to-many, or many-to-many relationships:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Types

Primitives

The following primitive types are used to specify methods and fields in the API Reference:

TYPE	DESCRIPTION
`string`	text strings
`int`	64-bit integers
`float`	IEEE double-precision floating-point numbers
`bool`	boolean
`datetime`	date and timestamp

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Higher-order types

The following type constructors are used:

TYPE	DESCRIPTION
`c ref`	reference to an object of class `c`
`t set`	a set of elements of type `t`
`(a -> b) map`	a table mapping values of type `a` to values of type `b`

Enumeration types

The following enumeration types are used:

Enum after_apply_guidance

NAMED VALUE	DESCRIPTION
`restartHost`	This patch requires the host to be restarted once applied.
`restartHVM`	This patch requires HVM guests to be restarted once applied.
`restartPV`	This patch requires PV guests to be restarted once applied.
`restartXAPI`	This patch requires XAPI to be restarted once applied.

Enum allocation_algorithm

NAMED VALUE	DESCRIPTION
`breadth_first`	vGPUs of a given type are allocated evenly across supporting pGPUs.
`depth_first`	vGPUs of a given type are allocated on supporting pGPUs until they are full.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum bond_mode

NAMED VALUE	DESCRIPTION
`active-backup`	Active/passive bonding: only one NIC is carrying traffic
`balance-slb`	Source-level balancing
`lacp`	Link aggregation control protocol

Enum certificate_purpose

NAMED VALUE	DESCRIPTION
`ldaps`	Trusted certificates that are for LDAPS connections to external authentication services.
`licensing`	Trusted certificates that are for licensing purpose.

Enum certificate_type

NAMED VALUE	DESCRIPTION
`ca`	Certificate that is trusted by the whole pool
`host`	Certificate that identifies a single host to entities outside the pool
`host_internal`	Certificate that identifies a single host to other pool members
`pinned`	Pinned leaf certificate that is trusted by the whole pool

Enum cls

NAMED VALUE	DESCRIPTION
`Certificate`	Certificate
`Host`	Host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`Pool`	Pool
`PVS_proxy`	PVS_proxy
`SR`	SR
`VDI`	VDI
`VM`	VM
`VMPP`	VMPP
`VMSS`	VMSS

Enum cluster_host_operation

NAMED VALUE	DESCRIPTION
`destroy`	completely destroying a cluster host
`disable`	disabling cluster membership on a particular host
`enable`	enabling cluster membership on a particular host

Enum cluster_operation

NAMED VALUE	DESCRIPTION
`add`	adding a new member to the cluster
`destroy`	completely destroying a cluster
`disable`	disabling any cluster member
`enable`	enabling any cluster member
`remove`	removing a member from the cluster

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum console_protocol

NAMED VALUE	DESCRIPTION
`rdp`	Remote Desktop Protocol
`rfb`	Remote FrameBuffer protocol (as used in VNC)
`vt100`	VT100 terminal

Enum domain_type

NAMED VALUE	DESCRIPTION
`hvm`	HVM; Fully Virtualised
`pv`	PV: Paravirtualised
`pv_in_pvh`	PV inside a PVH container
`pvh`	PVH
`unspecified`	Not specified or unknown domain type

Enum event_operation

NAMED VALUE	DESCRIPTION
`add`	An object has been created
`del`	An object has been deleted
`mod`	An object has been modified

Enum host_allowed_operations

NAMED VALUE	DESCRIPTION
`apply_updates`	Indicates this host is being updated
`enable`	Indicates this host is in the process of enabling
`evacuate`	Indicates this host is evacuating

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`power_on`	Indicates this host is in the process of being powered on
`provision`	Indicates this host is able to provision another VM
`reboot`	Indicates this host is in the process of rebooting
`shutdown`	Indicates this host is in the process of shutting itself down
`vm_migrate`	This host is the migration target of a VM
`vm_resume`	This host is resuming a VM
`vm_start`	This host is starting a VM

Enum host_display

NAMED VALUE	DESCRIPTION
`disable_on_reboot`	The host will stop outputting its console to a physical display device on next boot
`disabled`	This host is not outputting its console to a physical display device
`enable_on_reboot`	The host will start outputting its console to a physical display device on next boot
`enabled`	This host is outputting its console to a physical display device

Enum host_ntp_mode

NAMED VALUE	DESCRIPTION
`Custom`	Using custom NTP servers configured by user to sync time

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`DHCP`	Using NTP servers assigned by DHCP to sync time
`Disabled`	NTP is disabled on the host
`Factory`	Using built-in NTP servers to sync time

Enum host_numa_affinity_policy

NAMED VALUE	DESCRIPTION
`any`	VMs are spread across all available NUMA nodes
`best_effort`	VMs are placed on the smallest number of NUMA nodes that they fit using soft-pinning, but the policy doesn't guarantee a balanced placement, falling back to the 'any' policy.
`default_policy`	Use the NUMA affinity policy that is the default for the current version

Enum host_sched_gran

NAMED VALUE	DESCRIPTION
`core`	core scheduling
`cpu`	CPU scheduling
`socket`	socket scheduling

Enum image_format_type

NAMED VALUE	DESCRIPTION
`qcow2`	Qemu Copy-On-Write version 2
`raw`	Plain disk image
`vhd`	Virtual Hard Disk

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum ip_configuration_mode

NAMED VALUE	DESCRIPTION
`DHCP`	Acquire an IP address by DHCP
`None`	Do not acquire an IP address
`Static`	Static IP address configuration

Enum ipv6_configuration_mode

NAMED VALUE	DESCRIPTION
`Autoconf`	Router assigned prefix delegation IPv6 allocation
`DHCP`	Acquire an IPv6 address by DHCP
`None`	Do not acquire an IPv6 address
`Static`	Static IPv6 address configuration

Enum latest_synced_updates_applied_state

NAMED VALUE	DESCRIPTION
`no`	The host is outdated with the latest updates synced from remote CDN
`unknown`	If the host is up to date with the latest updates synced from remote CDN is unknown
`yes`	The host is up to date with the latest updates synced from remote CDN

Enum livepatch_status

NAMED VALUE	DESCRIPTION
`ok`	There is no applicable live patch
`ok_livepatch_complete`	An applicable live patch exists for every required component

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`ok_livepatch_incomplete`	An applicable live patch exists but it is not sufficient

Enum network_default_locking_mode

NAMED VALUE	DESCRIPTION
`disabled`	Treat all VIFs on this network with locking_mode = 'default' as if they have locking_mode = 'disabled'
`unlocked`	Treat all VIFs on this network with locking_mode = 'default' as if they have locking_mode = 'unlocked'

Enum network_operations

NAMED VALUE	DESCRIPTION
`attaching`	Indicates this network is attaching to a VIF or PIF

Enum network_purpose

NAMED VALUE	DESCRIPTION
`insecure_nbd`	Network Block Device service without integrity or confidentiality: NOT RECOMMENDED
`nbd`	Network Block Device service using TLS

Enum on_boot

NAMED VALUE	DESCRIPTION
`persist`	Standard behaviour.
`reset`	When a VM containing this VDI is started, the contents of the VDI are reset to the state they were in when this flag was last set.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum on_crash_behaviour

NAMED VALUE	DESCRIPTION
`coredump_and_destroy`	record a coredump and then destroy the VM state
`coredump_and_restart`	record a coredump and then restart the VM
`destroy`	destroy the VM state
`preserve`	leave the crashed VM paused
`rename_restart`	rename the crashed VM and start a new copy
`restart`	restart the VM

Enum on_normal_exit

NAMED VALUE	DESCRIPTION
`destroy`	destroy the VM state
`restart`	restart the VM

Enum on_softreboot_behavior

NAMED VALUE	DESCRIPTION
`destroy`	destroy the VM state
`preserve`	leave the VM paused
`restart`	restart the VM
`soft_reboot`	perform soft-reboot

Enum origin

NAMED VALUE	DESCRIPTION
`bundle`	The origin of the repository is a local bundle file
`remote`	The origin of the repository is a remote one

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`remote_pool`	The origin of the repository is a remote pool

Enum pci_dom0_access

NAMED VALUE	DESCRIPTION
`disable_on_reboot`	On host reboot dom0 will be blocked from accessing this device
`disabled`	dom0 cannot access this device
`enable_on_reboot`	On host reboot dom0 will be allowed to access this device
`enabled`	dom0 can access this device as normal

Enum persistence_backend

NAMED VALUE	DESCRIPTION
`xapi`	This VTPM is persisted in XAPI's DB

Enum pif_igmp_status

NAMED VALUE	DESCRIPTION
`disabled`	IGMP Snooping is disabled in the corresponding backend bridge.'
`enabled`	IGMP Snooping is enabled in the corresponding backend bridge.'
`unknown`	IGMP snooping status is unknown. If this is a VLAN master, then please consult the underlying VLAN slave PIF.

Enum placement_policy

NAMED VALUE	DESCRIPTION
`anti_affinity`	Anti-affinity placement policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`normal`	Default placement policy

Enum pool_allowed_operations

NAMED VALUE	DESCRIPTION
`apply_updates`	Indicates this pool is in the process of applying updates
`cert_refresh`	A certificate refresh and distribution is in progress
`cluster_create`	Indicates this pool is in the process of creating a cluster
`configure_repositories`	Indicates this pool is in the process of configuring repositories
`copy_primary_host_certs`	Indicates the primary host is sending its certificates to another host
`designate_new_master`	Indicates this pool is in the process of changing master
`eject`	Ejection of a host from the pool is under way
`exchange_ca_certificates_on_join`	Indicates this pool is exchanging ca certificates with a new joiner
`exchange_certificates_on_join`	Indicates this pool is exchanging internal certificates with a new joiner
`exchange_crls_on_join`	Indicates this pool is exchanging CRLs with a new joiner
`get_updates`	Indicates this pool is in the process of getting updates
`ha_disable`	Indicates this pool is in the process of disabling HA

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`ha_enable`	Indicates this pool is in the process of enabling HA
`sync_bundle`	Indicates this pool is in the process of syncing bundle
`sync_updates`	Indicates this pool is in the process of syncing updates
`tls_verification_enable`	Indicates this pool is in the process of enabling TLS verification

Enum pool_guest_secureboot_readiness

NAMED VALUE	DESCRIPTION
`not_ready`	Pool is not ready for SecureBoot, mandatory auth files are missing
`ready`	Pool is ready for SecureBoot, all auth files are present
`ready_no_dbx`	Pool is ready for SecureBoot, but there is no dbx auth file

Enum primary_address_type

NAMED VALUE	DESCRIPTION
`IPv4`	Primary address is the IPv4 address
`IPv6`	Primary address is the IPv6 address

Enum pvs_proxy_status

NAMED VALUE	DESCRIPTION
`caching`	The proxy is currently caching data
`incompatible_protocol_version`	The PVS protocol in use is not compatible with the PVS proxy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`incompatible_write_cache_mode`	The PVS device is configured to use an incompatible write-cache mode
`initialised`	The proxy is setup but has not yet cached anything
`stopped`	The proxy is not currently running

Enum sdn_controller_protocol

NAMED VALUE	DESCRIPTION
`pssl`	Passive ssl connection
`ssl`	Active ssl connection

Enum sr_health

NAMED VALUE	DESCRIPTION
`healthy`	Storage is fully available
`recovering`	Storage is busy recovering, for example, rebuilding mirrors.
`unavailable`	Storage is unavailable, a host reboot will be required
`unreachable`	Storage is unreachable but may be recoverable with admin intervention

Enum sriov_configuration_mode

NAMED VALUE	DESCRIPTION
`manual`	Configure network sriov manually
`modprobe`	Configure network sriov by modprobe, need reboot

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`sysfs`	Configure network sriov by sysfs, do not need reboot
`unknown`	Unknown mode

Enum storage_operations

NAMED VALUE	DESCRIPTION
`destroy`	Destroying the SR
`forget`	Forgetting about SR
`pbd_create`	Creating a PBD for this SR
`pbd_destroy`	Destroying one of this SR's PBDs
`plug`	Plugging a PBD into this SR
`scan`	Scanning backends for new or deleted VDIs
`unplug`	Unplugging a PBD from this SR
`update`	Refresh the fields on the SR
`vdi_blocked`	Blocking other operations for a VDI
`vdi_clone`	Cloneing a VDI
`vdi_copy`	Copying the VDI
`vdi_create`	Creating a new VDI
`vdi_data_destroy`	Deleting the data of the VDI
`vdi_destroy`	Destroying a VDI
`vdi_disable_cbt`	Disabling changed block tracking for a VDI
`vdi_enable_cbt`	Enabling changed block tracking for a VDI
`vdi_force_unlock`	Forcefully unlocking the VDI

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`vdi_forget`	Forgetting about the VDI
`vdi_generate_config`	Generating the configuration of the VDI
`vdi_introduce`	Introducing a new VDI
`vdi_list_changed_blocks`	Exporting a bitmap that shows the changed blocks between two VDIs
`vdi_mirror`	Mirroring a VDI
`vdi_resize`	Resizing a VDI
`vdi_resize_online`	Resizing the VDI online
`vdi_revert`	Reverting a VDI to the snapshot
`vdi_set_on_boot`	Setting the on_boot field of the VDI
`vdi_snapshot`	Snapshotting a VDI
`vdi_update`	Refreshing the fields on the VDI

Enum task_allowed_operations

NAMED VALUE	DESCRIPTION
`cancel`	refers to the operation "cancel"
`destroy`	refers to the operation "destroy"

Enum task_status_type

NAMED VALUE	DESCRIPTION
`cancelled`	task has been cancelled
`cancelling`	task is being cancelled
`failure`	task has failed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`pending`	task is in progress
`success`	task was completed successfully

Enum telemetry_frequency

NAMED VALUE	DESCRIPTION
`daily`	Run telemetry task daily
`monthly`	Run telemetry task monthly
`weekly`	Run telemetry task weekly

Enum tristate_type

NAMED VALUE	DESCRIPTION
`no`	Known to be false
`unspecified`	Unknown or unspecified
`yes`	Known to be true

Enum tunnel_protocol

NAMED VALUE	DESCRIPTION
`gre`	GRE protocol
`vxlan`	VxLAN Protocol

Enum update_after_apply_guidance

NAMED VALUE	DESCRIPTION
`restartHost`	This update requires the host to be restarted once applied.
`restartHVM`	This update requires HVM guests to be restarted once applied.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`restartPV`	This update requires PV guests to be restarted once applied.
`restartXAPI`	This update requires XAPI to be restarted once applied.

Enum update_guidances

NAMED VALUE	DESCRIPTION
`reboot_host`	Indicates the updated host should reboot as soon as possible
`reboot_host_on_kernel_livepatch_failure`	Indicates the updated host should reboot as soon as possible since one or more kernel livepatch(es) failed to be applied.
`reboot_host_on_livepatch_failure`	Indicates the updated host should reboot as soon as possible since one or more livepatch(es) failed to be applied.
`reboot_host_on_xen_livepatch_failure`	Indicates the updated host should reboot as soon as possible since one or more xen livepatch(es) failed to be applied.
`restart_device_model`	Indicates the device model of a running VM should restart as soon as possible
`restart_toolstack`	Indicates the Toolstack running on the updated host should restart as soon as possible
`restart_vm`	Indicates the VM should restart as soon as possible

Enum update_sync_frequency

NAMED VALUE	DESCRIPTION
`daily`	The update synchronizations happen every day

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`weekly`	The update synchronizations happen every week on the chosen day

Enum vbd_mode

NAMED VALUE	DESCRIPTION
`RO`	only read-only access will be allowed
`RW`	read-write access will be allowed

Enum vbd_operations

NAMED VALUE	DESCRIPTION
`attach`	Attempting to attach this VBD to a VM
`eject`	Attempting to eject the media from this VBD
`insert`	Attempting to insert new media into this VBD
`pause`	Attempting to pause a block device backend
`plug`	Attempting to hotplug this VBD
`unpause`	Attempting to unpause a block device backend
`unplug`	Attempting to hot unplug this VBD
`unplug_force`	Attempting to forcibly unplug this VBD

Enum vbd_type

NAMED VALUE	DESCRIPTION
`CD`	VBD will appear to guest as CD
`Disk`	VBD will appear to guest as disk
`Floppy`	VBD will appear as a floppy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vdi_operations

NAMED VALUE	DESCRIPTION
`blocked`	Operations on this VDI are temporarily blocked
`clone`	Cloning the VDI
`copy`	Copying the VDI
`data_destroy`	Deleting the data of the VDI
`destroy`	Destroying the VDI
`disable_cbt`	Disabling changed block tracking for a VDI
`enable_cbt`	Enabling changed block tracking for a VDI
`force_unlock`	Forcibly unlocking the VDI
`forget`	Forget about the VDI
`generate_config`	Generating static configuration
`list_changed_blocks`	Exporting a bitmap that shows the changed blocks between two VDIs
`mirror`	Mirroring the VDI
`resize`	Resizing the VDI
`resize_online`	Resizing the VDI which may or may not be online
`revert_from`	Reverting this VDI to a clone of a snapshot
`revert_to`	Reverting a VDI to a clone of this snapshot
`set_on_boot`	Setting the on_boot field of the VDI
`snapshot`	Snapshotting the VDI
`update`	Refreshing the fields of the VDI

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vdi_type

NAMED VALUE	DESCRIPTION
`cbt_metadata`	Metadata about a snapshot VDI that has been deleted: the set of blocks that changed between some previous version of the disk and the version tracked by the snapshot.
`crashdump`	a disk that stores VM crashdump information
`ephemeral`	a disk that may be reformatted on upgrade
`ha_statefile`	a disk used for HA storage heartbeating
`metadata`	a disk used for HA Pool metadata
`pvs_cache`	a disk that stores PVS cache data
`redo_log`	a disk used for a general metadata redo-log
`rrd`	a disk that stores SR-level RRDs
`suspend`	a disk that stores a suspend image
`system`	a disk that may be replaced on upgrade
`user`	a disk that is always preserved on upgrade

Enum vgpu_type_implementation

NAMED VALUE	DESCRIPTION
`gvt_g`	vGPU using Intel GVT-g
`mxgpu`	vGPU using AMD MxGPU
`nvidia`	vGPU using NVIDIA hardware
`nvidia_sriov`	vGPU using NVIDIA hardware with SR-IOV
`passthrough`	Pass through an entire physical GPU to a guest

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vif_ipv4_configuration_mode

NAMED VALUE	DESCRIPTION
`DHCP`	Acquire an IP address by DHCP
`None`	Follow the default IPv4 configuration of the guest (this is guest-dependent)
`Static`	Static IPv4 address configuration

Enum vif_ipv6_configuration_mode

NAMED VALUE	DESCRIPTION
`Autoconf`	Acquire an IPv6 address automatically
`None`	Follow the default IPv6 configuration of the guest (this is guest-dependent)
`Static`	Static IPv6 address configuration

Enum vif_locking_mode

NAMED VALUE	DESCRIPTION
`disabled`	No traffic is permitted
`locked`	Only traffic to a specific MAC and a list of IPv4 or IPv6 addresses is permitted
`network_default`	No specific configuration set - default network policy applies
`unlocked`	All traffic is permitted

Enum vif_operations

NAMED VALUE	DESCRIPTION
`attach`	Attempting to attach this VIF to a VM
`plug`	Attempting to hotplug this VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`unplug`	Attempting to hot unplug this VIF

Enum vm_appliance_operation

NAMED VALUE	DESCRIPTION
`clean_shutdown`	Clean shutdown
`hard_shutdown`	Hard shutdown
`shutdown`	Shutdown
`start`	Start

Enum vm_operations

NAMED VALUE	DESCRIPTION
`assert_operation_valid`	
`awaiting_memory_live`	Waiting for the memory settings to change
`call_plugin`	refers to the operation "call_plugin"
`changing_dynamic_range`	Changing the memory dynamic range
`changing_memory_limits`	Changing the memory limits
`changing_memory_live`	Changing the memory settings
`changing_NVRAM`	Changing NVRAM for a halted VM.
`changing_shadow_memory`	Changing the shadow memory for a halted VM.
`changing_shadow_memory_live`	Changing the shadow memory for a running VM.
`changing_static_range`	Changing the memory static range
`changing_VCPUs`	Changing VCPU settings for a halted VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`changing_VCPUs_live`	Changing VCPU settings for a running VM.
`checkpoint`	refers to the operation "checkpoint"
`clean_reboot`	refers to the operation "clean_reboot"
`clean_shutdown`	refers to the operation "clean_shutdown"
`clone`	refers to the operation "clone"
`copy`	refers to the operation "copy"
`create_template`	refers to the operation "create_template"
`create_vtpm`	Creating and adding a VTPM to this VM
`csvm`	refers to the operation "csvm"
`data_source_op`	Add, remove, query or list data sources
`destroy`	refers to the act of uninstalling the VM
`export`	exporting a VM to a network stream
`get_boot_record`	refers to the operation "get_boot_record"
`hard_reboot`	refers to the operation "hard_reboot"
`hard_shutdown`	refers to the operation "hard_shutdown"
`import`	importing a VM from a network stream
`make_into_template`	Turning this VM into a template
`metadata_export`	exporting VM metadata to a network stream
`migrate_send`	refers to the operation "migrate_send"
`pause`	refers to the operation "pause"
`pool_migrate`	refers to the operation "pool_migrate"

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`power_state_reset`	refers to the operation "power_state_reset"
`provision`	refers to the operation "provision"
`query_services`	refers to the operation "query_services"
`resume`	refers to the operation "resume"
`resume_on`	refers to the operation "resume_on"
`revert`	refers to the operation "revert"
`reverting`	Reverting the VM to a previous snapshotted state
`send_sysrq`	refers to the operation "send_sysrq"
`send_trigger`	refers to the operation "send_trigger"
`shutdown`	refers to the operation "shutdown"
`snapshot`	refers to the operation "snapshot"
`snapshot_with_quiesce`	refers to the operation "snapshot_with_quiesce"
`start`	refers to the operation "start"
`start_on`	refers to the operation "start_on"
`suspend`	refers to the operation "suspend"
`sysprep`	Performing a Windows sysprep on this VM
`unpause`	refers to the operation "unpause"
`update_allowed_operations`	

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vm_power_state

NAMED VALUE	DESCRIPTION
`Halted`	VM is offline and not using any resources
`Paused`	All resources have been allocated but the VM itself is paused and its vCPUs are not running
`Running`	Running
`Suspended`	VM state has been saved to disk and it is nolonger running. Note that disks remain in-use while the VM is suspended.

Enum vm_secureboot_certificates_state

NAMED VALUE	DESCRIPTION
`ok`	The VM's certificates do not need to be updated (including the case where Secure Boot does not apply to this VM, e.g. BIOS VM).
`update_available`	The Secure Boot certificates are due to expire or have already expired.
`update_on_boot`	An update of the certificates will be triggered whenever the VM boots. This includes VM.start, VM.reboot and a guest-triggered reboot.

Enum vm_secureboot_readiness

NAMED VALUE	DESCRIPTION
`certs_incomplete`	Secured boot is enabled on this VM and the certificates defined in its EFI variables are incomplete
`disabled`	Secureboot is disabled on this VM
`first_boot`	Secured boot is enabled on this VM and its NVRAM.EFI-variables are empty

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`not_supported`	VM's firmware is not UEFI
`ready`	Secured boot is enabled on this VM and PK, KEK, db and dbx are defined in its EFI variables
`ready_no_dbx`	Secured boot is enabled on this VM and PK, KEK, db but not dbx are defined in its EFI variables
`setup_mode`	Secured boot is enabled on this VM and PK is not defined in its EFI variables

Enum vm_uefi_mode

NAMED VALUE	DESCRIPTION
`setup`	clears a VM's EFI variables related to Secure Boot and places it into Setup Mode
`user`	resets a VM's EFI variables related to Secure Boot to the defaults, placing it into User Mode

Enum vmpp_archive_frequency

NAMED VALUE	DESCRIPTION
`always_after_backup`	Archive after backup
`daily`	Daily archives
`never`	Never archive
`weekly`	Weekly backups

Enum vmpp_archive_target_type

NAMED VALUE	DESCRIPTION
`cifs`	CIFS target config

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`nfs`	NFS target config
`none`	No target config

Enum vmpp_backup_frequency

NAMED VALUE	DESCRIPTION
`daily`	Daily backups
`hourly`	Hourly backups
`weekly`	Weekly backups

Enum vmpp_backup_type

NAMED VALUE	DESCRIPTION
`checkpoint`	The backup is a checkpoint
`snapshot`	The backup is a snapshot

Enum vmss_frequency

NAMED VALUE	DESCRIPTION
`daily`	Daily snapshots
`hourly`	Hourly snapshots
`weekly`	Weekly snapshots

Enum vmss_type

NAMED VALUE	DESCRIPTION
`checkpoint`	The snapshot is a checkpoint
`snapshot`	The snapshot is a disk snapshot
`snapshot_with_quiesce`	Support for VSS has been removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vtpm_operations

NAMED VALUE	DESCRIPTION
`destroy`	Destroy a VTPM

Enum vusb_operations

NAMED VALUE	DESCRIPTION
`attach`	Attempting to attach this VUSB to a VM
`plug`	Attempting to plug this VUSB into a VM
`unplug`	Attempting to hot unplug this VUSB

Class: auth

Management of remote authentication services

Fields for class: auth

Class auth has no fields.

RPCs associated with class: auth

RPC name: getgroupmembership

Overview:

This calls queries the external directory service to obtain the transitivelyclosed set of groups that the the subjectidentifier is member of.

Signature:

```
string set get_group_membership (session ref session_ref, string
subject_identifier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`subject_identifier`	A string containing the subject_identifier, unique in the external directory service

Minimum Role: read-only

Return Type: `string set`

set of subjectidentifiers that provides the group membership of subjectidentifier passed as argument, it contains, recursively, all groups a subjectidentifier is member of.

RPC name: getsubjectidentifier

Overview:

This call queries the external directory service to obtain the subjectidentifier as a string from the humanreadable subjectname

Signature:

```
string get_subject_identifier (session ref session_ref, string
subject_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`subject_name`	The human-readable subject_name, such as a username or a groupname

Minimum Role: read-only

Return Type: `string`

the subjectidentifier obtained from the external directory service

RPC name: getsubjectinformationfromidentifier

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

This call queries the external directory service to obtain the user information for example, username, organization etc. from the specified subjectidentifier

Signature:

```
(string -> string) map get_subject_information_from_identifier (session ref session_ref, string subject_identifier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`subject_identifier`	A string containing the subject_identifier, unique in the external directory service

Minimum Role: read-only

Return Type: `(string -> string) map`

keyvalue pairs containing at least a key called subjectname

Class: blob

A placeholder for a binary blob

Fields for class: blob

FIELD	TYPE	QUALIFIER	DESCRIPTION
`last_updated`	`datetime`	*RO/constructor*	Time at which the data in the blob was last updated
`mime_type`	`string`	*RO/constructor*	The mime type associated with this object. Defaults to 'application/octet-stream' if the empty string is supplied

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`public`	`bool`	*RW*	True if the blob is publicly accessible
`size`	`int`	*RO/runtime*	Size of the binary data, in bytes
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: blob

RPC name: create

Overview:

Create a placeholder for a binary blob

Signature:

```
blob ref create (session ref session_ref, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`mime_type`	The mime-type of the blob. Defaults to 'application/octet-stream' if the empty string is supplied
`bool`	`public`	True if the blob should be publicly available

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the created blob

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	The reference of the blob to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the blobs known to the system.

Signature:

```
blob ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `blob ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of blob references to blob records for all blobs known to the system.

Signature:

```
(blob ref -> blob record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(blob ref -> blob record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the blob instances with the given label.

Signature:

```
blob ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `blob ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the blob instance with the specified UUID.

Signature:

```
blob ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `blob ref`

reference to the object

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given blob.

Signature:

```
datetime get_last_updated (session ref session_ref, blob ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getmimetype

Overview:

Get the mimetype field of the given blob.

Signature:

```
string get_mime_type (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given blob.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_name_description (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given blob.

Signature:

```
string get_name_label (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpublic

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the public field of the given blob.

Signature:

```
bool get_public (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given blob.

Signature:

```
blob record get_record (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `blob record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsize

Overview:

Get the size field of the given blob.

Signature:

```
int get_size (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given blob.

Signature:

```
string get_uuid (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: setnamedescription

Overview:

Set the name/description field of the given blob.

Signature:

```
void set_name_description (session ref session_ref, blob ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given blob.

Signature:

```
void set_name_label (session ref session_ref, blob ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setpublic

Overview:

Set the public field of the given blob.

Signature:

```
void set_public (session ref session_ref, blob ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: Bond

A Network bond that combines physical network interfaces, also known as link aggregation

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Bond

FIELD	TYPE	QUALIFIER	DESCRIPTION
`auto_update_mac`	`bool`	*RO/runtime*	true if the MAC was taken from the primary slave when the bond was created, and false if the client specified the MAC
`links_up`	`int`	*RO/runtime*	Number of links up in this bond
`master`	`PIF ref`	*RO/constructor*	The bonded interface
`mode`	`bond_mode`	*RO/runtime*	The algorithm used to distribute traffic among the bonded NICs
`other_config`	`(string -> string) map`	*RW*	additional configuration
`primary_slave`	`PIF ref`	*RO/runtime*	The PIF of which the IP configuration and MAC were copied to the bond, and which will receive all configuration/VLANs/VIFs on the bond if the bond is destroyed
`properties`	`(string -> string) map`	*RO/runtime*	Additional configuration properties specific to the bond mode.
`slaves`	`PIF ref set`	*RO/runtime*	The interfaces which are part of this bond

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: Bond

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given Bond.

Signature:

```
void add_to_other_config (session ref session_ref, Bond ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create an interface bond

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Bond ref create (session ref session_ref, network ref network, PIF ref set
members, string MAC, bond_mode mode, (string -> string) map properties)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	Network to add the bonded PIF to
`PIF ref set`	`members`	PIFs to add to this bond
`string`	`MAC`	The MAC address to use on the bond itself. If this parameter is the empty string then the bond will inherit its MAC address from the primary slave.
`bond_mode`	`mode`	Bonding mode to use for the new bond
`(string -> string) map`	`properties`	Additional configuration parameters specific to the bond mode

Minimum Role: pool-operator

Return Type: `Bond ref`

The reference of the created Bond object

RPC name: destroy

Overview:

Destroy an interface bond

Signature:

```
void destroy (session ref session_ref, Bond ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	Bond to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the Bonds known to the system.

Signature:

```
Bond ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Bond ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Bond references to Bond records for all Bonds known to the system.

Signature:

```
(Bond ref -> Bond record) map get_all_records (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Bond ref -> Bond record) map`

records of all objects

RPC name: getautoupdatemac

Overview:

Get the autoupdatemac field of the given Bond.

Signature:

```
bool get_auto_update_mac (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the Bond instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Bond ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Bond ref`

reference to the object

RPC name: getlinksup

Overview:

Get the linksup field of the given Bond.

Signature:

```
int get_links_up (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmaster

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the master field of the given Bond.

Signature:

```
PIF ref get_master (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getmode

Overview:

Get the mode field of the given Bond.

Signature:

```
bond_mode get_mode (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bond_mode`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given Bond.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprimaryslave

Overview:

Get the primaryslave field of the given Bond.

Signature:

```
PIF ref get_primary_slave (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getproperties

Overview:

Get the properties field of the given Bond.

Signature:

```
(string -> string) map get_properties (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Bond.

Signature:

```
Bond record get_record (session ref session_ref, Bond ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Bond record`

all fields from the object

RPC name: getslaves

Overview:

Get the slaves field of the given Bond.

Signature:

```
PIF ref set get_slaves (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Bond.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given Bond. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, Bond ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setmode

Overview:

Change the bond mode

Signature:

```
void set_mode (session ref session_ref, Bond ref self, bond_mode value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	The bond
`bond_mode`	`value`	The new bond mode

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given Bond.

Signature:

```
void set_other_config (session ref session_ref, Bond ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setproperty

Overview:

Set the value of a property of the bond

Signature:

```
void set_property (session ref session_ref, Bond ref self, string name,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	The bond
`string`	`name`	The property name
`string`	`value`	The property value

Minimum Role: pool-operator

Return Type: `void`

Class: Certificate

An X509 certificate used for TLS connections

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Certificate

FIELD	TYPE	QUALIFIER	DESCRIPTION
`fingerprint`	`string`	*RO/constructor*	**Deprecated.** Use fingerprint_sha256 instead
`fingerprint_sha1`	`string`	*RO/constructor*	The certificate's SHA1 fingerprint / hash
`fingerprint_sha256`	`string`	*RO/constructor*	The certificate's SHA256 fingerprint / hash
`host`	`host ref`	*RO/constructor*	The host where the certificate is installed
`name`	`string`	*RO/runtime*	The name of the certificate, only present on certificates of type 'ca'
`not_after`	`datetime`	*RO/constructor*	Date before which the certificate is valid
`not_before`	`datetime`	*RO/constructor*	Date after which the certificate is valid
`purpose`	`certificate_purpose set`	*RO/constructor*	The purposes of the certificate
`type`	`certificate_type`	*RO/runtime*	The type of the certificate, either 'ca', 'host' or 'host_internal'
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: Certificate

RPC name: getall

Overview:

Return a list of all the Certificates known to the system.

Signature:

```
Certificate ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Certificate ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Certificate references to Certificate records for all Certificates known to the system.

Signature:

```
(Certificate ref -> Certificate record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(Certificate ref -> Certificate record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the Certificate instance with the specified UUID.

Signature:

```
Certificate ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Certificate ref`

reference to the object

RPC name: getfingerprint

This message is deprecated.

Overview:

Get the fingerprint field of the given Certificate.

Signature:

```
string get_fingerprint (session ref session_ref, Certificate ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfingerprintsha1

Overview:

Get the fingerprintsha1 field of the given Certificate.

Signature:

```
string get_fingerprint_sha1 (session ref session_ref, Certificate ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfingerprintsha256

Overview:

Get the fingerprintsha256 field of the given Certificate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_fingerprint_sha256 (session ref session_ref, Certificate ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given Certificate.

Signature:

```
host ref get_host (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getname

Overview:

Get the name field of the given Certificate.

Signature:

```
string get_name (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnotafter

Overview:

Get the notafter field of the given Certificate.

Signature:

```
datetime get_not_after (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `datetime`

value of the field

RPC name: getnotbefore

Overview:

Get the notbefore field of the given Certificate.

Signature:

```
datetime get_not_before (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getpurpose

Overview:

Get the purpose field of the given Certificate.

Signature:

```
certificate_purpose set get_purpose (session ref session_ref, Certificate
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `certificate_purpose set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Certificate.

Signature:

```
Certificate record get_record (session ref session_ref, Certificate ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Certificate record`

all fields from the object

RPC name: gettype

Overview:

Get the type field of the given Certificate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
certificate_type get_type (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `certificate_type`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Certificate.

Signature:

```
string get_uuid (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: Cluster

Clusterwide Cluster metadata

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Cluster

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`cluster_operation set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`cluster_config`	`(string -> string) map`	*RO/constructor*	Contains read-only settings for the cluster, such as timeouts and other options. It can only be set at cluster create time
`cluster_hosts`	`Cluster_host ref set`	*RO/runtime*	A list of the cluster_host objects associated with the Cluster
`cluster_stack`	`string`	*RO/constructor*	Simply the string 'corosync'. No other cluster stacks are currently supported
`cluster_stack_version`	`int`	*RO/constructor*	Version of cluster stack, not writable via the API. Defaulting to 2 for backwards compatibility when upgrading from a cluster without this field, which means it is necessarily running version 2 of corosync, the only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			cluster stack supported so far.
`cluster_token`	`string`	*RO/constructor*	The secret key used by xapi-clusterd when it talks to itself on other hosts
`current_operations`	`(string -> cluster_operation) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`expected_hosts`	`int`	*RO/runtime*	Total number of hosts expected by the cluster stack
`is_quorate`	`bool`	*RO/runtime*	Whether the cluster stack thinks the cluster is quorate
`live_hosts`	`int`	*RO/runtime*	Current number of live hosts, according to the cluster stack
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`pending_forget`	`string set`	*RO/runtime*	Internal field used by Host.destroy to store the IP of cluster members marked as permanently dead but not yet removed
`pool_auto_join`	`bool`	*RO/constructor*	True if automatically joining new pool members to the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			cluster. This will be `true` in the first release
`quorum`	`int`	*RO/runtime*	Number of live hosts in order to be quorate
`token_timeout`	`float`	*RO/constructor*	The corosync token timeout in seconds
`token_timeout_coefficient`	`float`	*RO/constructor*	The corosync token timeout coefficient in seconds
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: Cluster

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given Cluster.

Signature:

```
void add_to_other_config (session ref session_ref, Cluster ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Creates a Cluster object and one Clusterhost object as its first member

Signature:

```
Cluster ref create (session ref session_ref, PIF ref PIF, string
cluster_stack, bool pool_auto_join, float token_timeout, float
token_timeout_coefficient)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`PIF`	The PIF to connect the cluster's first cluster_host to
`string`	`cluster_stack`	simply the string 'corosync'. No other cluster stacks are currently supported
`bool`	`pool_auto_join`	true if xapi is automatically joining new pool members to the cluster
`float`	`token_timeout`	Corosync token timeout in seconds
`float`	`token_timeout_coefficient`	Corosync token timeout coefficient in seconds

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `Cluster ref`

the new Cluster

Possible Error Codes: `INVALID_CLUSTER_STACK`, `INVALID_VALUE`, `PIF_ALLOWS_UNPLUG`, `REQUIRED_PIF_IS_UNPLUGGED`

RPC name: destroy

Overview:

Destroys a Cluster object and the one remaining Clusterhost member

Signature:

```
void destroy (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	the Cluster to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_DOES_NOT_HAVE_ONE_NODE`, `CLUSTER_STACK_IN_USE`

RPC name: getall

Overview:

Return a list of all the Clusters known to the system.

Signature:

```
Cluster ref set get_all (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Cluster ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Cluster references to Cluster records for all Clusters known to the system.

Signature:

```
(Cluster ref -> Cluster record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Cluster ref -> Cluster record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given Cluster.

Signature:

```
cluster_operation set get_allowed_operations (session ref session_ref,
Cluster ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `cluster_operation set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the Cluster instance with the specified UUID.

Signature:

```
Cluster ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Cluster ref`

reference to the object

RPC name: getclusterconfig

Overview:

Get the clusterconfig field of the given Cluster.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_cluster_config (session ref session_ref,
Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getclusterhosts

Overview:

Get the clusterhosts field of the given Cluster.

Signature:

```
Cluster_host ref set get_cluster_hosts (session ref session_ref, Cluster
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster_host ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getclusterstack

Overview:

Get the clusterstack field of the given Cluster.

Signature:

```
string get_cluster_stack (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getclusterstackversion

Overview:

Get the clusterstackversion field of the given Cluster.

Signature:

```
int get_cluster_stack_version (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getclustertoken

Overview:

Get the clustertoken field of the given Cluster.

Signature:

```
string get_cluster_token (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given Cluster.

Signature:

```
(string -> cluster_operation) map get_current_operations (session ref
session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> cluster_operation) map`

value of the field

RPC name: getexpectedhosts

Overview:

Get the expectedhosts field of the given Cluster.

Signature:

```
int get_expected_hosts (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getisquorate

Overview:

Get the isquorate field of the given Cluster.

Signature:

```
bool get_is_quorate (session ref session_ref, Cluster ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlivehosts

Overview:

Get the livehosts field of the given Cluster.

Signature:

```
int get_live_hosts (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnetwork

Overview:

Returns the network used by the cluster for interhost communication, i.e. the network shared by all cluster host PIFs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
network ref get_network (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	the Cluster with the network

Minimum Role: read-only

Return Type: `network ref`

network of cluster

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given Cluster.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, Cluster
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpendingforget

Overview:

Get the pendingforget field of the given Cluster.

Signature:

```
string set get_pending_forget (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getpoolautojoin

Overview:

Get the poolautojoin field of the given Cluster.

Signature:

```
bool get_pool_auto_join (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getquorum

Overview:

Get the quorum field of the given Cluster.

Signature:

```
int get_quorum (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Cluster.

Signature:

```
Cluster record get_record (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster record`

all fields from the object

RPC name: gettokentimeout

Overview:

Get the tokentimeout field of the given Cluster.

Signature:

```
float get_token_timeout (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: gettokentimeoutcoefficient

Overview:

Get the tokentimeoutcoefficient field of the given Cluster.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
float get_token_timeout_coefficient (session ref session_ref, Cluster ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Cluster.

Signature:

```
string get_uuid (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: poolcreate

Overview:

Attempt to create a Cluster from the entire pool

Signature:

```
Cluster ref pool_create (session ref session_ref, network ref network,
string cluster_stack, float token_timeout, float
token_timeout_coefficient)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	the single network on which corosync carries out its inter-host communications
`string`	`cluster_stack`	simply the string 'corosync'. No other cluster stacks are currently supported
`float`	`token_timeout`	Corosync token timeout in seconds
`float`	`token_timeout_coefficient`	Corosync token timeout coefficient in seconds

Minimum Role: pool-operator

Return Type: `Cluster ref`

the new Cluster

RPC name: pooldestroy

Overview:

Attempt to destroy the Clusterhost objects for all hosts in the pool and then destroy the Cluster.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void pool_destroy (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	The cluster to destroy.

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_STACK_IN_USE`, `CLUSTERING_DISABLED`, `CLUSTER_HOST_IS_LAST`

RPC name: poolforcedestroy

Overview:

Attempt to force destroy the Clusterhost objects, and then destroy the Cluster.

Signature:

```
void pool_force_destroy (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	The cluster to force destroy.

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_FORCE_DESTROY_FAILED`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: poolresync

Overview:

Resynchronise the clusterhost objects across the pool. Creates them where they need creating and then plugs them

Signature:

```
void pool_resync (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	The cluster to resync

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given Cluster. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, Cluster ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given Cluster.

Signature:

```
void set_other_config (session ref session_ref, Cluster ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: Clusterhost

Cluster member metadata

Fields for class: Clusterhost

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`cluster_host_operation set`	*RO/runtime*	list of the operations allowed in this state.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			This list is advisory only and the server state may have changed by the time this field is read by a client.
`cluster`	`Cluster ref`	*RO/constructor*	Reference to the Cluster object
`current_operations`	`(string -> cluster_host_operation) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`enabled`	`bool`	*RO/runtime*	Whether the cluster host believes that clustering should be enabled on this host. This field can be altered by calling the enable/disable message on a cluster host. Only enabled members run the underlying cluster stack. Disabled members are still considered a member of the cluster (see joined), and can be re-enabled by the user.
`host`	`host ref`	*RO/constructor*	Reference to the Host object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`joined`	`bool`	*RO/runtime*	Whether the cluster host has joined the cluster. Contrary to enabled, a host that is not joined is not considered a member of the cluster, and hence enable and disable operations cannot be performed on this host.
`last_update_live`	`datetime`	*RO/runtime*	Time when the live field was last updated based on information from the cluster stack
`live`	`bool`	*RO/runtime*	Whether the underlying cluster stack thinks we are live. This field is set automatically based on updates from the cluster stack and cannot be altered by the user.
`other_config`	`(string -> string) map`	*RO/constructor*	Additional configuration
`PIF`	`PIF ref`	*RO/constructor*	Reference to the PIF object
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: Clusterhost

RPC name: create

Overview:

Add a new host to an existing cluster.

Signature:

```
Cluster_host ref create (session ref session_ref, Cluster ref cluster,
host ref host, PIF ref pif)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`cluster`	Cluster to join
`host ref`	`host`	new cluster member
`PIF ref`	`pif`	Network interface to use for communication

Minimum Role: pool-operator

Return Type: `Cluster_host ref`

the newly created clusterhost object

Possible Error Codes: `PIF_NOT_ATTACHED_TO_HOST`, `NO_CLUSTER_HOSTS_REACHABLE`

RPC name: destroy

Overview:

Remove the host from an existing cluster. This operation is allowed even if a cluster host is not enabled.

Signature:

```
void destroy (session ref session_ref, Cluster_host ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to remove from the cluster

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_STACK_IN_USE`, `CLUSTERING_DISABLED`, `CLUSTER_HOST_IS_LAST`

RPC name: disable

Overview:

Disable cluster membership for an enabled cluster host.

Signature:

```
void disable (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to disable

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_STACK_IN_USE`

RPC name: enable

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable cluster membership for a disabled cluster host.

Signature:

```
void enable (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to enable

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `PIF_ALLOWS_UNPLUG`, `REQUIRED_PIF_IS_UNPLUGGED`

RPC name: forcedestroy

Overview:

Remove a host from an existing cluster forcefully.

Signature:

```
void force_destroy (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to remove from the cluster

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `CLUSTER_STACK_IN_USE`

RPC name: getall

Overview:

Return a list of all the Clusterhosts known to the system.

Signature:

```
Cluster_host ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Cluster_host ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Clusterhost references to Clusterhost records for all Clusterhosts known to the system.

Signature:

```
(Cluster_host ref -> Cluster_host record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(Cluster_host ref -> Cluster_host record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given Clusterhost.

Signature:

```
cluster_host_operation set get_allowed_operations (session ref
session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `cluster_host_operation set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the Clusterhost instance with the specified UUID.

Signature:

```
Cluster_host ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Cluster_host ref`

reference to the object

RPC name: getcluster

Overview:

Get the cluster field of the given Clusterhost.

Signature:

```
Cluster ref get_cluster (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster ref`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given Clusterhost.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> cluster_host_operation) map get_current_operations (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> cluster_host_operation) map`

value of the field

RPC name: getenabled

Overview:

Get the enabled field of the given Clusterhost.

Signature:

```
bool get_enabled (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given Clusterhost.

Signature:

```
host ref get_host (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getjoined

Overview:

Get the joined field of the given Clusterhost.

Signature:

```
bool get_joined (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getlastupdatelive

Overview:

Get the lastupdatelive field of the given Clusterhost.

Signature:

```
datetime get_last_update_live (session ref session_ref, Cluster_host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlive

Overview:

Get the live field of the given Clusterhost.

Signature:

```
bool get_live (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given Clusterhost.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPIF

Overview:

Get the PIF field of the given Clusterhost.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PIF ref get_PIF (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Clusterhost.

Signature:

```
Cluster_host record get_record (session ref session_ref, Cluster_host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster_host record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given Clusterhost.

Signature:

```
string get_uuid (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: console

A console

Fields for class: console

FIELD	TYPE	QUALIFIER	DESCRIPTION
`location`	`string`	*RO/runtime*	URI for the console service
`other_config`	`(string -> string) map`	*RW*	additional configuration
`protocol`	`console_protocol`	*RO/runtime*	the protocol used by this console
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VM`	`VM ref`	*RO/runtime*	VM to which this console is attached

RPCs associated with class: console

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given console.

Signature:

```
void add_to_other_config (session ref session_ref, console ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: create

Overview:

Create a new console instance, and return its handle.

Signature:

```
console ref create (session ref session_ref, console record args)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `console ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified console instance.

Signature:

```
void destroy (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the consoles known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
console ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `console ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of console references to console records for all consoles known to the system.

Signature:

```
(console ref -> console record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(console ref -> console record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the console instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
console ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `console ref`

reference to the object

RPC name: getlocation

Overview:

Get the location field of the given console.

Signature:

```
string get_location (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the otherconfig field of the given console.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprotocol

Overview:

Get the protocol field of the given console.

Signature:

```
console_protocol get_protocol (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `console_protocol`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given console.

Signature:

```
console record get_record (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `console record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given console.

Signature:

```
string get_uuid (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given console.

Signature:

```
VM ref get_VM (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given console. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, console ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given console.

Signature:

```
void set_other_config (session ref session_ref, console ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: crashdump

This class is deprecated.

A VM crashdump

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: crashdump

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference
`VDI`	`VDI ref`	*RO/constructor*	**Deprecated.** the virtual disk
`VM`	`VM ref`	*RO/constructor*	**Deprecated.** the virtual machine

RPCs associated with class: crashdump

RPC name: addtootherconfig

This message is deprecated.

Overview:

Add the given keyvalue pair to the otherconfig field of the given crashdump.

Signature:

```
void add_to_other_config (session ref session_ref, crashdump ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

This message is deprecated.

Overview:

Destroy the specified crashdump

Signature:

```
void destroy (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	The crashdump to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the crashdumps known to the system.

Signature:

```
crashdump ref set get_all (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `crashdump ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

Return a map of crashdump references to crashdump records for all crashdumps known to the system.

Signature:

```
(crashdump ref -> crashdump record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(crashdump ref -> crashdump record) map`

records of all objects

RPC name: getbyuuid

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a reference to the crashdump instance with the specified UUID.

Signature:

```
crashdump ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `crashdump ref`

reference to the object

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given crashdump.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given crashdump.

Signature:

```
crashdump record get_record (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `crashdump record`

all fields from the object

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given crashdump.

Signature:

```
string get_uuid (session ref session_ref, crashdump ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDI

This message is deprecated.

Overview:

Get the VDI field of the given crashdump.

Signature:

```
VDI ref get_VDI (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getVM

This message is deprecated.

Overview:

Get the VM field of the given crashdump.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM ref get_VM (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given crashdump. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, crashdump ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object
`string`	`key`	Key to remove

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given crashdump.

Signature:

```
void set_other_config (session ref session_ref, crashdump ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: datasource

Data sources for logging in RRDs

Fields for class: datasource

FIELD	TYPE	QUALIFIER	DESCRIPTION
`enabled`	`bool`	*RO/runtime*	true if the data source is being logged

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`max`	`float`	*RO/runtime*	the maximum value of the data source
`min`	`float`	*RO/runtime*	the minimum value of the data source
`name_description`	`string`	*RO/runtime*	a notes field containing human-readable description
`name_label`	`string`	*RO/runtime*	a human-readable name
`standard`	`bool`	*RO/runtime*	true if the data source is enabled by default. Non-default data sources cannot be disabled
`units`	`string`	*RO/runtime*	the units of the value
`value`	`float`	*RO/runtime*	current value of the data source

RPCs associated with class: datasource

Class datasource has no RPCs associated with it.

Class: DRtask

DR task

Fields for class: DRtask

FIELD	TYPE	QUALIFIER	DESCRIPTION
`introduced_SRs`	`SR ref set`	*RO/runtime*	All SRs introduced by this appliance
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: DRtask

RPC name: create

Overview:

Create a disaster recovery task which will query the supplied list of devices

Signature:

```
DR_task ref create (session ref session_ref, string type, (string ->
string) map device_config, string set whitelist)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`type`	The SR driver type of the SRs to introduce
`(string -> string) map`	`device_config`	The device configuration of the SRs to introduce
`string set`	`whitelist`	The devices to use for disaster recovery

Minimum Role: pool-operator

Return Type: `DR_task ref`

The reference of the created DRtask

RPC name: destroy

Overview:

Destroy the disaster recovery task, detaching and forgetting any SRs introduced which are no longer required

Signature:

```
void destroy (session ref session_ref, DR_task ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	The disaster recovery task to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the DRtasks known to the system.

Signature:

```
DR_task ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `DR_task ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of DRtask references to DRtask records for all DRtasks known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(DR_task ref -> DR_task record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(DR_task ref -> DR_task record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the DRtask instance with the specified UUID.

Signature:

```
DR_task ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `DR_task ref`

reference to the object

RPC name: getintroducedSRs

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the introducedSRs field of the given DRtask.

Signature:

```
SR ref set get_introduced_SRs (session ref session_ref, DR_task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given DRtask.

Signature:

```
DR_task record get_record (session ref session_ref, DR_task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `DR_task record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given DRtask.

Signature:

```
string get_uuid (session ref session_ref, DR_task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: Drivervariant

UNSUPPORTED. Variant of a host driver

Fields for class: Drivervariant

FIELD	TYPE	QUALIFIER	DESCRIPTION
`driver`	`Host_driver ref`	*RO/runtime*	Driver this variant is a part of
`hardware_present`	`bool`	*RO/runtime*	True if the hardware for this variant is present on the host
`name`	`string`	*RO/runtime*	Name identifying the driver variant within the driver

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`priority`	`float`	*RO/runtime*	Priority; this needs an explanation how this is ordered
`status`	`string`	*RO/runtime*	Development and release status of this variant, like 'alpha'
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`version`	`string`	*RO/runtime*	Unique version of this driver variant

RPCs associated with class: Drivervariant

RPC name: getall

Overview:

Return a list of all the Drivervariants known to the system.

Signature:

```
Driver_variant ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Driver_variant ref set`

references to all objects

RPC name: getallrecords

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return a map of Drivervariant references to Drivervariant records for all Drivervariants known to the system.

Signature:

```
(Driver_variant ref -> Driver_variant record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Driver_variant ref -> Driver_variant record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the Drivervariant instance with the specified UUID.

Signature:

```
Driver_variant ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Driver_variant ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getdriver

Overview:

Get the driver field of the given Drivervariant.

Signature:

```
Host_driver ref get_driver (session ref session_ref, Driver_variant ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Host_driver ref`

value of the field

RPC name: gethardwarepresent

Overview:

Get the hardwarepresent field of the given Drivervariant.

Signature:

```
bool get_hardware_present (session ref session_ref, Driver_variant ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getname

Overview:

Get the name field of the given Drivervariant.

Signature:

```
string get_name (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpriority

Overview:

Get the priority field of the given Drivervariant.

Signature:

```
float get_priority (session ref session_ref, Driver_variant ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Drivervariant.

Signature:

```
Driver_variant record get_record (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant record`

all fields from the object

RPC name: getstatus

Overview:

Get the status field of the given Drivervariant.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_status (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Drivervariant.

Signature:

```
string get_uuid (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the version field of the given Drivervariant.

Signature:

```
string get_version (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: select

Overview:

UNSUPPORTED Select this variant of a driver to become active after reboot or immediately if currently no version is active

Signature:

```
void select (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	Driver variant to become active (after reboot).

Minimum Role: pool-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: event

Asynchronous event registration and handling

Fields for class: event

FIELD	TYPE	QUALIFIER	DESCRIPTION
`class`	`string`	*RO/constructor*	The name of the class of the object that changed
`id`	`int`	*RO/constructor*	An ID, monotonically increasing, and local to the current session
`obj_uuid`	`string`	*RO/constructor*	**Deprecated.** The uuid of the object that changed
`operation`	`event_operation`	*RO/constructor*	The operation that was performed
`ref`	`string`	*RO/constructor*	A reference to the object that changed
`timestamp`	`datetime`	*RO/constructor*	**Deprecated.** The time at which the event occurred
snapshot	object record	*RO/runtime*	The record of the database object that was added, changed or deleted

RPCs associated with class: event

RPC name: from

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Blocking call which returns a new token and a possibly empty batch of events. The returned token can be used in subsequent calls to this function. It eliminates redundant events for example, same field updated multiple times.

Signature:

```
event batch from (session ref session_ref, string set classes, string
token, float timeout)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`classes`	register for events for the indicated classes
`string`	`token`	A token representing the point from which to generate database events. The empty string represents the beginning.
`float`	`timeout`	Return after this many seconds if no events match

Minimum Role: read-only

Return Type: `event batch`

a structure consisting of a token 'token', a map of valid references per object type 'validrefcounts', and a set of event records 'events'.

Possible Error Codes: `EVENT_FROM_TOKEN_PARSE_FAILURE`, `EVENT_SUBSCRIPTION_PARSE_FAILURE`

RPC name: getcurrentid

Overview:

Return the ID of the next event to be generated by the system

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
int get_current_id (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `int`

the event ID

RPC name: inject

Overview:

Injects an artificial event on the given object and returns the corresponding ID in the form of a token, which can be used as a point of reference for database events. For example, to check whether an object has reached the right state before attempting an operation, one can inject an artificial event on the object and wait until the token returned by consecutive event.from calls is lexicographically greater than the one returned by event.inject.

Signature:

```
string inject (session ref session_ref, string class, string ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`class`	class of the object
`string`	`ref`	A reference to the object that will be changed.

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

the event ID in the form of a token

RPC name: next

This message is deprecated.

Overview:

Blocking call which returns a possibly empty batch of events. This method is only recommended for legacy use.It stores events in a buffer of limited size, raising EVENTSLOST if too many events got generated. New development should use event.from which supersedes this method.

Signature:

```
event record set next (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `event record set`

A set of events

Possible Error Codes: `SESSION_NOT_REGISTERED`, `EVENTS_LOST`, `EVENT_SUBSCRIPTION_PARSE_FAILURE`

RPC name: register

This message is deprecated.

Overview:

Registers this session with the event system for a set of given classes. This method is only recommended for legacy use in conjunction with event.next.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void register (session ref session_ref, string set classes)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`classes`	the classes for which the session will register with the event system; specifying * as the desired class will register for all classes

Minimum Role: read-only

Return Type: `void`

RPC name: unregister

This message is deprecated.

Overview:

Removes this session's registration with the event system for a set of given classes. This method is only recommended for legacy use in conjunction with event.next.

Signature:

```
void unregister (session ref session_ref, string set classes)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`classes`	the classes for which the session's registration with the event system will be removed

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Class: Feature

A new piece of functionality

Fields for class: Feature

FIELD	TYPE	QUALIFIER	DESCRIPTION
`enabled`	`bool`	*RO/runtime*	Indicates whether the feature is enabled
`experimental`	`bool`	*RO/constructor*	Indicates whether the feature is experimental (as opposed to stable and fully supported)
`host`	`host ref`	*RO/runtime*	The host where this feature is available
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`version`	`string`	*RO/constructor*	The version of this feature

RPCs associated with class: Feature

RPC name: getall

Overview:

Return a list of all the Features known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Feature ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Feature ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Feature references to Feature records for all Features known to the system.

Signature:

```
(Feature ref -> Feature record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Feature ref -> Feature record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the Feature instances with the given label.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Feature ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `Feature ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the Feature instance with the specified UUID.

Signature:

```
Feature ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Feature ref`

reference to the object

RPC name: getenabled

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the enabled field of the given Feature.

Signature:

```
bool get_enabled (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getexperimental

Overview:

Get the experimental field of the given Feature.

Signature:

```
bool get_experimental (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given Feature.

Signature:

```
host ref get_host (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given Feature.

Signature:

```
string get_name_description (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given Feature.

Signature:

```
string get_name_label (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Feature.

Signature:

```
Feature record get_record (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Feature record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given Feature.

Signature:

```
string get_uuid (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

Overview:

Get the version field of the given Feature.

Signature:

```
string get_version (session ref session_ref, Feature ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: GPUgroup

A group of compatible GPUs across the resource pool

Fields for class: GPUgroup

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allocation_algorithm`	`allocation_algorithm`	*RW*	Current allocation of vGPUs to pGPUs for this group
`enabled_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	vGPU types supported on at least one of the pGPUs in this group
`GPU_types`	`string set`	*RO/runtime*	List of GPU types (vendor+device ID) that can be in this group
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PGPUs`	`PGPU ref set`	*RO/runtime*	List of pGPUs in the group
`supported_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	vGPU types supported on at least one of the pGPUs in this group
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VGPUs`	`VGPU ref set`	*RO/runtime*	List of vGPUs using the group

RPCs associated with class: GPUgroup

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given GPUgroup.

Signature:

```
void add_to_other_config (session ref session_ref, GPU_group ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Signature:

```
GPU_group ref create (session ref session_ref, string name_label, string
name_description, (string -> string) map other_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	
`string`	`name_description`	
`(string -> string) map`	`other_config`	

Minimum Role: pool-operator

Return Type: `GPU_group ref`

The reference of the created GPUgroup

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, GPU_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	The GPU group to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the GPUgroups known to the system.

Signature:

```
GPU_group ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `GPU_group ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of GPUgroup references to GPUgroup records for all GPUgroups known to the system.

Signature:

```
(GPU_group ref -> GPU_group record) map get_all_records (session ref
session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(GPU_group ref -> GPU_group record) map`

records of all objects

RPC name: getallocationalgorithm

Overview:

Get the allocationalgorithm field of the given GPUgroup.

Signature:

```
allocation_algorithm get_allocation_algorithm (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `allocation_algorithm`

value of the field

RPC name: getbynamelabel

Overview:

Get all the GPUgroup instances with the given label.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
GPU_group ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `GPU_group ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the GPUgroup instance with the specified UUID.

Signature:

```
GPU_group ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `GPU_group ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getenabledVGPUtypes

Overview:

Get the enabledVGPUtypes field of the given GPUgroup.

Signature:

```
VGPU_type ref set get_enabled_VGPU_types (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getGPUtypes

Overview:

Get the GPUtypes field of the given GPUgroup.

Signature:

```
string set get_GPU_types (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given GPUgroup.

Signature:

```
string get_name_description (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given GPUgroup.

Signature:

```
string get_name_label (session ref session_ref, GPU_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given GPUgroup.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPGPUs

Overview:

Get the PGPUs field of the given GPUgroup.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PGPU ref set get_PGPUs (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given GPUgroup.

Signature:

```
GPU_group record get_record (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group record`

all fields from the object

RPC name: getremainingcapacity

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
int get_remaining_capacity (session ref session_ref, GPU_group ref self,
VGPU_type ref vgpu_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	The GPU group to query
`VGPU_type ref`	`vgpu_type`	The VGPU_type for which the remaining capacity will be calculated

Minimum Role: read-only

Return Type: `int`

The number of VGPUs of the given type which can still be started on the PGPUs in the group

RPC name: getsupportedVGPUtypes

Overview:

Get the supportedVGPUtypes field of the given GPUgroup.

Signature:

```
VGPU_type ref set get_supported_VGPU_types (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given GPUgroup.

Signature:

```
string get_uuid (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVGPUs

Overview:

Get the VGPUs field of the given GPUgroup.

Signature:

```
VGPU ref set get_VGPUs (session ref session_ref, GPU_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given GPUgroup. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, GPU_group ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setallocationalgorithm

Overview:

Set the allocationalgorithm field of the given GPUgroup.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_allocation_algorithm (session ref session_ref, GPU_group ref
self, allocation_algorithm value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`allocation_algorithm`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given GPUgroup.

Signature:

```
void set_name_description (session ref session_ref, GPU_group ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given GPUgroup.

Signature:

```
void set_name_label (session ref session_ref, GPU_group ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given GPUgroup.

Signature:

```
void set_other_config (session ref session_ref, GPU_group ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`GPU_group ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: host

A physical host

Fields for class: host

FIELD	TYPE	QUALIFIER	DESCRIPTION
`address`	`string`	*RW*	The address by which this host can be contacted from any other host in the pool
`allowed_operations`	`host_allowed_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`API_version_major`	`int`	*RO/runtime*	major version number
`API_version_minor`	`int`	*RO/runtime*	minor version number
`API_version_vendor`	`string`	*RO/runtime*	identification of vendor

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`API_version_vendor_implementation`	`(string -> string) map`	*RO/runtime*	details of vendor implementation
`bios_strings`	`(string -> string) map`	*RO/runtime*	BIOS strings
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this host
`capabilities`	`string set`	*RO/constructor*	Xen capabilities
`certificates`	`Certificate ref set`	*RO/runtime*	List of certificates installed in the host
`chipset_info`	`(string -> string) map`	*RO/runtime*	Information about chipset features
`console_idle_timeout`	`int`	*RO/runtime*	The timeout in seconds after which idle console will be automatically terminated (0 means never)
`control_domain`	`VM ref`	*RO/runtime*	The control domain (domain 0)
`cpu_configuration`	`(string -> string) map`	*RO/runtime*	The CPU configuration on this host. May contain keys such as "nr_nodes", "sockets_per_node", "cores_per_socket", or "threads_per_core"
`cpu_info`	`(string -> string) map`	*RO/runtime*	Details about the physical CPUs on this host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`crash_dump_sr`	`SR ref`	*RW*	The SR in which VDIs for crash dumps are created
`crashdumps`	`host_crashdump ref set`	*RO/runtime*	Set of host crash dumps
`current_operations`	`(string -> host_allowed_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`display`	`host_display`	*RW*	indicates whether the host is configured to output its console to a physical display device
`edition`	`string`	*RO/runtime*	Product edition
`editions`	`string set`	*RO/runtime*	List of all available product editions
`enabled`	`bool`	*RO/runtime*	True if the host is currently enabled
`external_auth_configuration`	`(string -> string) map`	*RO/runtime*	configuration specific to external authentication service
`external_auth_service_name`	`string`	*RO/runtime*	name of external authentication service configured; empty if none configured.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`external_auth_type`	`string`	*RO/runtime*	type of external authentication service configured; empty if none configured.
`features`	`Feature ref set`	*RO/runtime*	List of features available on this host
`guest_VCPUs_params`	`(string -> string) map`	*RW*	VCPUs params to apply to all resident guests
`ha_network_peers`	`string set`	*RO/runtime*	The set of hosts visible via the network from this host
`ha_statefiles`	`string set`	*RO/runtime*	The set of statefiles accessible from this host
`host_CPUs`	`host_cpu ref set`	*RO/runtime*	The physical CPUs on this host
`hostname`	`string`	*RW*	The hostname of this host
`https_only`	`bool`	*RO/runtime*	Reflects whether port 80 is open (false) or not (true)
`iscsi_iqn`	`string`	*RO/constructor*	The initiator IQN for the host
`last_software_update`	`datetime`	*RO/runtime*	Date and time when the last software update was applied
`last_update_hash`	`string`	*RO/runtime*	The SHA256 checksum of updateinfo of the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			most recently applied update on the host
`latest_synced_updates_applied`	`latest_synced_updates_applied_state`	*RO/runtime*	Default as 'unknown', 'yes' if the host is up to date with updates synced from remote CDN, otherwise 'no'
`license_params`	`(string -> string) map`	*RO/runtime*	State of the current license
`license_server`	`(string -> string) map`	*RW*	Contact information of the license server
`local_cache_sr`	`SR ref`	*RO/constructor*	The SR that is used as a local cache
`logging`	`(string -> string) map`	*RW*	logging configuration
`max_cstate`	`string`	*RO/runtime*	The maximum C-state that the host is allowed to enter, "" means unlimited; "N" means limit to CN; "N,M" means limit to CN with max sub cstate M.
`memory_overhead`	`int`	*RO/runtime*	Virtualization memory overhead (bytes).
`metrics`	`host_metrics ref`	*RO/runtime*	metrics associated with this host
`multipathing`	`bool`	*RO/constructor*	Specifies whether multipathing is enabled

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`ntp_custom_servers`	`string set`	*RO/runtime*	Custom NTP servers configured by users, used in Custom NTP mode
`ntp_mode`	`host_ntp_mode`	*RO/runtime*	Indicates NTP servers are assigned by DHCP, or configured by user, or the factory servers, or NTP is disabled
`numa_affinity_policy`	`host_numa_affinity_policy`	*RO/runtime*	NUMA-aware VM memory and vCPU placement policy
`other_config`	`(string -> string) map`	*RW*	additional configuration
`patches`	`host_patch ref set`	*RO/runtime*	**Deprecated.** Set of host patches
`PBDs`	`PBD ref set`	*RO/runtime*	physical blockdevices
`PCIs`	`PCI ref set`	*RO/runtime*	List of PCI devices in the host
`pending_guidances`	`update_guidances set`	*RO/runtime*	The set of pending mandatory guidances after applying updates, which must be

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			applied, otherwise there may be, for example, VM failures
`pending_guidances_full`	`update_guidances set`	*RO/runtime*	The set of pending full guidances after applying updates, which a user should follow to make some updates, for example, specific hardware drivers or CPU features, fully effective, but the 'average user' doesn't need to
`pending_guidances_recommended`	`update_guidances set`	*RO/runtime*	The set of pending recommended guidances after applying updates, which most users should follow to make the updates effective, but if not followed, will not cause a failure
`PGPUs`	`PGPU ref set`	*RO/runtime*	List of physical GPUs in the host
`PIFs`	`PIF ref set`	*RO/runtime*	physical network interfaces
`power_on_config`	`(string -> string) map`	*RO/runtime*	The power on config
`power_on_mode`	`string`	*RO/runtime*	The power on mode
`PUSBs`	`PUSB ref set`	*RO/runtime*	List of physical USBs in the host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`resident_VMs`	`VM ref set`	*RO/runtime*	list of VMs currently resident on host
`sched_policy`	`string`	*RO/runtime*	Scheduler policy currently in force on this host
`secure_boot`	`bool`	*RO/runtime*	Whether the host has booted in secure boot mode
`software_version`	`(string -> string) map`	*RO/constructor*	version strings
`ssh_auto_mode`	`bool`	*RO/runtime*	Reflects whether SSH auto mode is enabled for the host
`ssh_enabled`	`bool`	*RO/runtime*	True if SSH access is enabled for the host
`ssh_enabled_timeout`	`int`	*RO/runtime*	The timeout in seconds after which SSH access will be automatically disabled (0 means never), this setting will be applied every time the SSH is enabled by XAPI
`ssh_expiry`	`datetime`	*RO/runtime*	The time in UTC after which the SSH access will be automatically disabled
`ssl_legacy`	`bool`	*RO/constructor*	**Deprecated.** Allow SSLv3 protocol and ciphersuites as used

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			by older server versions. This controls both incoming and outgoing connections. When this is set to a different value, the host immediately restarts its SSL/TLS listening service; typically this takes less than a second but existing connections to it will be broken. API login sessions will remain valid.
`supported_bootloaders`	`string set`	*RO/runtime*	a list of the bootloaders installed on the machine
`suspend_image_sr`	`SR ref`	*RW*	The SR in which VDIs for suspend images are created
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`timezone`	`string`	*RO/runtime*	The time zone identifier as defined in the IANA Time Zone Database
`tls_verification_enabled`	`bool`	*RO/runtime*	True if this host has TLS verifcation enabled

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uefi_certificates`	`string`	*RO/constructor*	**Deprecated.** The UEFI certificates allowing Secure Boot
`updates`	`pool_update ref set`	*RO/runtime*	Set of updates
`updates_requiring_reboot`	`pool_update ref set`	*RO/runtime*	List of updates which require reboot
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`virtual_hardware_platform_versions`	`int set`	*RO/runtime*	The set of versions of the virtual hardware platform that the host can offer to its guests

RPCs associated with class: host

RPC name: addtags

Overview:

Add the given value to the tags field of the given host. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtoguestVCPUsparams

Overview:

Add the given keyvalue pair to the guestVCPUsparams field of the given host.

Signature:

```
void add_to_guest_VCPUs_params (session ref session_ref, host ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtolicenseserver

Overview:

Add the given keyvalue pair to the licenseserver field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_license_server (session ref session_ref, host ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtologging

Overview:

Add the given keyvalue pair to the logging field of the given host.

Signature:

```
void add_to_logging (session ref session_ref, host ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given host.

Signature:

```
void add_to_other_config (session ref session_ref, host ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: applyedition

Overview:

Change to another edition, or reactivate the current edition after a license has expired. This may be subject to the successful checkout of an appropriate license.

Signature:

```
void apply_edition (session ref session_ref, host ref host, string
edition, bool force)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`edition`	The requested edition
`bool`	`force`	Update the license params even if the apply call fails

Minimum Role: pool-operator

Return Type: `void`

RPC name: applyrecommendedguidances

This message is removed.

Overview:

apply all recommended guidances both on the host and on all HVM VMs on the host after updates are applied on the host

Signature:

```
void apply_recommended_guidances (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host whose recommended guidances will be applied

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: applyupdates

Overview:

apply updates from current enabled repository on a host

Signature:

```
string set set apply_updates (session ref session_ref, host ref self,
string hash)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host where updates will be applied
`string`	`hash`	The hash of updateinfo to be applied which is returned by previous pool.sync_udpates

Minimum Role: client-cert

Return Type: `string set set`

The list of results after applying updates, including livepatch apply failures and recommended guidances

RPC name: assertcanevacuate

Overview:

Check this host can be evacuated.

Signature:

```
void assert_can_evacuate (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to evacuate

Minimum Role: pool-operator

Return Type: `void`

RPC name: backuprrds

Overview:

This causes the RRDs to be backed up to the master

Signature:

```
void backup_rrds (session ref session_ref, host ref host, float delay)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Schedule a backup of the RRDs of this host
`float`	`delay`	Delay in seconds from when the call is received to perform the backup

Minimum Role: pool-admin

Return Type: `void`

RPC name: bugreportupload

Overview:

Run xenbugtool yestoall and upload the output to support

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void bugreport_upload (session ref session_ref, host ref host, string url,
(string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host on which to run xen-bugtool
`string`	`url`	The URL to upload to
`(string -> string) map`	`options`	Extra configuration operations

Minimum Role: pool-operator

Return Type: `void`

RPC name: callextension

Overview:

Call an API extension on this host

Signature:

```
string call_extension (session ref session_ref, host ref host, string
call)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`call`	Rpc call for the extension

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `string`

Result from the extension

RPC name: callplugin

Overview:

Call an API plugin on this host

Signature:

```
string call_plugin (session ref session_ref, host ref host, string plugin,
string fn, (string -> string) map args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`plugin`	The name of the plugin
`string`	`fn`	The name of the function within the plugin
`(string -> string) map`	`args`	Arguments for the function

Minimum Role: pool-admin

Return Type: `string`

Result from the plugin

RPC name: computefreememory

Overview:

Computes the amount of free memory on the host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int compute_free_memory (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to send the request to

Minimum Role: read-only

Return Type: `int`

the amount of free memory on the host.

RPC name: computememoryoverhead

Overview:

Computes the virtualization memory overhead of a host.

Signature:

```
int compute_memory_overhead (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host for which to compute the memory overhead

Minimum Role: read-only

Return Type: `int`

the virtualization memory overhead of the host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this host

Signature:

```
blob ref create_new_blob (session ref session_ref, host ref host, string
name, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: declaredead

Overview:

Declare that a host is dead. This is a dangerous operation, and should only be called if the administrator is absolutely sure the host is definitely dead

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void declare_dead (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to declare is dead

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

Overview:

Destroy specified host record in database

Signature:

```
void destroy (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host record to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: disable

Overview:

Puts the host into a state in which no new VMs can be started. Currently active VMs on the host continue to execute.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void disable (session ref session_ref, host ref host, bool auto_enable)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to disable
`bool`	`auto_enable`	If true (default), the host will be re-enabled after a toolstack restart automatically. If false, the host will be disabled indefinitely, across toolstack restarts and host reboots, until re-enabled explicitly with Host.enable.

Minimum Role: client-cert

Return Type: `void`

RPC name: disabledisplay

Overview:

Disable console output to the physical display device next time this host boots

Signature:

```
host_display disable_display (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `host_display`

This host's physical display usage

RPC name: disableexternalauth

Overview:

This call disables external authentication on the local host

Signature:

```
void disable_external_auth (session ref session_ref, host ref host,
(string -> string) map config, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose external authentication should be disabled
`(string -> string) map`	`config`	Optional parameters as a list of key-values containing the configuration data
`bool`	`force`	Disable external auth even when not enabled

Minimum Role: pool-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: disablelocalstoragecaching

Overview:

Disable the use of a local SR for caching purposes

Signature:

```
void disable_local_storage_caching (session ref session_ref, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `void`

RPC name: disablessh

Overview:

Disable SSH access on the host. It will stop the service sshd only if it is running. It will also disable the service sshd only if it is enabled. A newly joined host in the pool or an ejected host from the pool would keep the original status.

Signature:

```
void disable_ssh (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: dmesg

Overview:

Get the host xen dmesg.

Signature:

```
string dmesg (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to query

Minimum Role: pool-operator

Return Type: `string`

dmesg string

RPC name: dmesgclear

Overview:

Get the host xen dmesg, and clear the buffer.

Signature:

```
string dmesg_clear (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`host`	The Host to query

Minimum Role: pool-operator

Return Type: `string`

dmesg string

RPC name: emergencyclearmandatoryguidance

Overview:

Clear the pending mandatory guidance on this host

Signature:

```
void emergency_clear_mandatory_guidance (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Return Type: `void`

RPC name: emergencydisabletlsverification

Overview:

Disable TLS verification for this host only

Signature:

```
void emergency_disable_tls_verification (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: emergencyhadisable

Overview:

This call disables HA on the local host. This should only be used with extreme care.

Signature:

```
void emergency_ha_disable (session ref session_ref, bool soft)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`bool`	`soft`	Disable HA temporarily, revert upon host reboot or further changes, idempotent

Minimum Role: pool-operator

Return Type: `void`

RPC name: emergencyreenabletlsverification

Overview:

Reenable TLS verification for this host only, and only after it was emergency disabled

Signature:

```
void emergency_reenable_tls_verification (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: emergencyresetservercertificate

Overview:

Delete the current TLS server certificate and replace by a new, selfsigned one. This should only be used with extreme care.

Signature:

```
void emergency_reset_server_certificate (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Return Type: `void`

RPC name: enable

Overview:

Puts the host into a state in which new VMs can be started.

Signature:

```
void enable (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to enable

Minimum Role: client-cert

Return Type: `void`

RPC name: enabledisplay

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable console output to the physical display device next time this host boots

Signature:

```
host_display enable_display (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `host_display`

This host's physical display usage

RPC name: enableexternalauth

Overview:

This call enables external authentication on a host

Signature:

```
void enable_external_auth (session ref session_ref, host ref host, (string
-> string) map config, string service_name, string auth_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose external authentication should be enabled
`(string -> string) map`	`config`	A list of key-values containing the configuration data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`service_name`	The name of the service
`string`	`auth_type`	The type of authentication (for example, AD for Active Directory)

Minimum Role: pool-admin

Return Type: `void`

RPC name: enablelocalstoragecaching

Overview:

Enable the use of a local SR for caching purposes

Signature:

```
void enable_local_storage_caching (session ref session_ref, host ref host,
SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`SR ref`	`sr`	The SR to use as a local cache

Minimum Role: pool-operator

Return Type: `void`

RPC name: enablessh

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable SSH access on the host. It will start the service sshd only if it is not running. It will also enable the service sshd only if it is not enabled. A newly joined host in the pool or an ejected host from the pool would keep the original status.

Signature:

```
void enable_ssh (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: evacuate

Overview:

Migrate all VMs off of this host, where possible.

Signature:

```
void evacuate (session ref session_ref, host ref host, network ref
network, int evacuate_batch_size)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to evacuate
`network ref`	`network`	Optional preferred network for migration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`int`	`evacuate_batch_size`	The maximum number of VMs to be migrated per batch 0 will use the value `evacuation-batch-size` defined in xapi.conf

Minimum Role: client-cert

Return Type: `void`

RPC name: externalauthsetldaps

Overview:

This call enables or disables LDAPS for external authentication on the host

Signature:

```
void external_auth_set_ldaps (session ref session_ref, host ref host, bool ldaps, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose LDAPS configuration should be set
`bool`	`ldaps`	Whether to enable or disable LDAPS
`bool`	`force`	Force the operation even if already in the desired state

Minimum Role: pool-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: forgetdatasourcearchives

Overview:

Forget the recorded statistics related to the specified data source

Signature:

```
void forget_data_source_archives (session ref session_ref, host ref host,
string data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`data_source`	The data source whose archives are to be forgotten

Minimum Role: pool-operator

Return Type: `void`

RPC name: getaddress

Overview:

Get the address field of the given host.

Signature:

```
string get_address (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getall

Overview:

Return a list of all the hosts known to the system.

Signature:

```
host ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of host references to host records for all hosts known to the system.

Signature:

```
(host ref -> host record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(host ref -> host record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given host.

Signature:

```
host_allowed_operations set get_allowed_operations (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_allowed_operations set`

value of the field

RPC name: getAPIversionmajor

Overview:

Get the APIversion/major field of the given host.

Signature:

```
int get_API_version_major (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getAPIversionminor

Overview:

Get the APIversion/minor field of the given host.

Signature:

```
int get_API_version_minor (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getAPIversionvendor

Overview:

Get the APIversion/vendor field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_API_version_vendor (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getAPIversionvendorimplementation

Overview:

Get the APIversion/vendorimplementation field of the given host.

Signature:

```
(string -> string) map get_API_version_vendor_implementation (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbiosstrings

Overview:

Get the biosstrings field of the given host.

Signature:

```
(string -> string) map get_bios_strings (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given host.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getbynamelabel

Overview:

Get all the host instances with the given label.

Signature:

```
host ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `host ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the host instance with the specified UUID.

Signature:

```
host ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host ref`

reference to the object

RPC name: getcapabilities

Overview:

Get the capabilities field of the given host.

Signature:

```
string set get_capabilities (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getcertificates

Overview:

Get the certificates field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Certificate ref set get_certificates (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Certificate ref set`

value of the field

RPC name: getchipsetinfo

Overview:

Get the chipsetinfo field of the given host.

Signature:

```
(string -> string) map get_chipset_info (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getconsoleidletimeout

Overview:

Get the consoleidletimeout field of the given host.

Signature:

```
int get_console_idle_timeout (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getcontroldomain

Overview:

Get the controldomain field of the given host.

Signature:

```
VM ref get_control_domain (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VM ref`

value of the field

RPC name: getcpuconfiguration

Overview:

Get the cpuconfiguration field of the given host.

Signature:

```
(string -> string) map get_cpu_configuration (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcpuinfo

Overview:

Get the cpuinfo field of the given host.

Signature:

```
(string -> string) map get_cpu_info (session ref session_ref, host ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcrashdumpsr

Overview:

Get the crashdumpsr field of the given host.

Signature:

```
SR ref get_crash_dump_sr (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getcrashdumps

Overview:

Get the crashdumps field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host_crashdump ref set get_crashdumps (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_crashdump ref set`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given host.

Signature:

```
(string -> host_allowed_operations) map get_current_operations (session
ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> host_allowed_operations) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getdatasources

Overview:

Signature:

```
data_source record set get_data_sources (session ref session_ref, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to interrogate

Minimum Role: read-only

Return Type: `data_source record set`

A set of data sources

RPC name: getdisplay

Overview:

Get the display field of the given host.

Signature:

```
host_display get_display (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `host_display`

value of the field

RPC name: getedition

Overview:

Get the edition field of the given host.

Signature:

```
string get_edition (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: geteditions

Overview:

Get the editions field of the given host.

Signature:

```
string set get_editions (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getenabled

Overview:

Get the enabled field of the given host.

Signature:

```
bool get_enabled (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getexternalauthconfiguration

Overview:

Get the externalauthconfiguration field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_external_auth_configuration (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getexternalauthservicename

Overview:

Get the externalauthservicename field of the given host.

Signature:

```
string get_external_auth_service_name (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getexternalauthtype

Overview:

Get the externalauthtype field of the given host.

Signature:

```
string get_external_auth_type (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfeatures

Overview:

Get the features field of the given host.

Signature:

```
Feature ref set get_features (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `Feature ref set`

value of the field

RPC name: getguestVCPUsparams

Overview:

Get the guestVCPUsparams field of the given host.

Signature:

```
(string -> string) map get_guest_VCPUs_params (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethanetworkpeers

Overview:

Get the hanetworkpeers field of the given host.

Signature:

```
string set get_ha_network_peers (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gethastatefiles

Overview:

Get the hastatefiles field of the given host.

Signature:

```
string set get_ha_statefiles (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gethostCPUs

Overview:

Get the hostCPUs field of the given host.

Signature:

```
host_cpu ref set get_host_CPUs (session ref session_ref, host ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_cpu ref set`

value of the field

RPC name: gethostname

Overview:

Get the hostname field of the given host.

Signature:

```
string get_hostname (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethttpsonly

Overview:

Get the httpsonly field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_https_only (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getiscsiiqn

Overview:

Get the iscsiiqn field of the given host.

Signature:

```
string get_iscsi_iqn (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getlastsoftwareupdate

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the lastsoftwareupdate field of the given host.

Signature:

```
datetime get_last_software_update (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlastupdatehash

Overview:

Get the lastupdatehash field of the given host.

Signature:

```
string get_last_update_hash (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getlatestsyncedupdatesapplied

Overview:

Get the latestsyncedupdatesapplied field of the given host.

Signature:

```
latest_synced_updates_applied_state get_latest_synced_updates_applied
(session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `latest_synced_updates_applied_state`

value of the field

RPC name: getlicenseparams

Overview:

Get the licenseparams field of the given host.

Signature:

```
(string -> string) map get_license_params (session ref session_ref, host
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlicenseserver

Overview:

Get the licenseserver field of the given host.

Signature:

```
(string -> string) map get_license_server (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlocalcachesr

Overview:

Get the localcachesr field of the given host.

Signature:

```
SR ref get_local_cache_sr (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getlog

Overview:

Get the host's log file

Signature:

```
string get_log (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to query

Minimum Role: read-only

Return Type: `string`

The contents of the host's primary log file

RPC name: getlogging

Overview:

Get the logging field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_logging (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getmanagementinterface

Overview:

Returns the management interface for the specified host

Signature:

```
PIF ref get_management_interface (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Which host's management interface is required

Minimum Role: pool-operator

Return Type: `PIF ref`

The management interface for the host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getmaxcstate

Overview:

Get the maxcstate field of the given host.

Signature:

```
string get_max_cstate (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getmemoryoverhead

Overview:

Get the memory/overhead field of the given host.

Signature:

```
int get_memory_overhead (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getmetrics

Overview:

Get the metrics field of the given host.

Signature:

```
host_metrics ref get_metrics (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_metrics ref`

value of the field

RPC name: getmultipathing

Overview:

Get the multipathing field of the given host.

Signature:

```
bool get_multipathing (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given host.

Signature:

```
string get_name_description (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given host.

Signature:

```
string get_name_label (session ref session_ref, host ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getntpcustomservers

Overview:

Get the ntpcustomservers field of the given host.

Signature:

```
string set get_ntp_custom_servers (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getntpmode

Overview:

Get the ntpmode field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host_ntp_mode get_ntp_mode (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_ntp_mode`

value of the field

RPC name: getntpserversstatus

Overview:

Get the NTP servers status on the host

Signature:

```
(string -> string) map get_ntp_servers_status (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `(string -> string) map`

The map of NTP server to its status, status may be synced/combined/uncombined/error/variable/unreachable/unknown

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getntpsynchronized

Overview:

Returns true if the system clock on the host is synchronized with the NTP servers.

Signature:

```
bool get_ntp_synchronized (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `bool`

true if the system clock on the host is synchronized with the NTP servers.

RPC name: getnumaaffinitypolicy

Overview:

Get the numaaffinitypolicy field of the given host.

Signature:

```
host_numa_affinity_policy get_numa_affinity_policy (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `host_numa_affinity_policy`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given host.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpatches

This message is deprecated.

Overview:

Get the patches field of the given host.

Signature:

```
host_patch ref set get_patches (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_patch ref set`

value of the field

RPC name: getPBDs

Overview:

Get the PBDs field of the given host.

Signature:

```
PBD ref set get_PBDs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PBD ref set`

value of the field

RPC name: getPCIs

Overview:

Get the PCIs field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PCI ref set get_PCIs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref set`

value of the field

RPC name: getpendingguidances

Overview:

Get the pendingguidances field of the given host.

Signature:

```
update_guidances set get_pending_guidances (session ref session_ref, host
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpendingguidancesfull

Overview:

Get the pendingguidancesfull field of the given host.

Signature:

```
update_guidances set get_pending_guidances_full (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getpendingguidancesrecommended

Overview:

Get the pendingguidancesrecommended field of the given host.

Signature:

```
update_guidances set get_pending_guidances_recommended (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getPGPUs

Overview:

Get the PGPUs field of the given host.

Signature:

```
PGPU ref set get_PGPUs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getPIFs

Overview:

Get the PIFs field of the given host.

Signature:

```
PIF ref set get_PIFs (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref set`

value of the field

RPC name: getpoweronconfig

Overview:

Get the poweronconfig field of the given host.

Signature:

```
(string -> string) map get_power_on_config (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpoweronmode

Overview:

Get the poweronmode field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_power_on_mode (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPUSBs

Overview:

Get the PUSBs field of the given host.

Signature:

```
PUSB ref set get_PUSBs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PUSB ref set`

value of the field

RPC name: getrecord

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a record containing the current state of the given host.

Signature:

```
host record get_record (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host record`

all fields from the object

RPC name: getresidentVMs

Overview:

Get the residentVMs field of the given host.

Signature:

```
VM ref set get_resident_VMs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getschedgran

Overview:

Gets xen's schedgran on a host

Signature:

```
host_sched_gran get_sched_gran (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Return Type: `host_sched_gran`

The host's schedgran

RPC name: getschedpolicy

Overview:

Get the schedpolicy field of the given host.

Signature:

```
string get_sched_policy (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getsecureboot

Overview:

Get the secureboot field of the given host.

Signature:

```
bool get_secure_boot (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getservercertificate

Overview:

Get the installed server public TLS certificate.

Signature:

```
string get_server_certificate (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

The installed server public TLS certificate, in PEM form.

RPC name: getserverlocaltime

Overview:

This call queries the host's clock for the current time in the host's local timezone

Signature:

```
datetime get_server_localtime (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose clock should be queried

Minimum Role: read-only

Return Type: `datetime`

The current local time

RPC name: getservertime

Overview:

This call queries the host's clock for the current time

Signature:

```
datetime get_servertime (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose clock should be queried

Minimum Role: read-only

Return Type: `datetime`

The current time

RPC name: getsoftwareversion

Overview:

Get the softwareversion field of the given host.

Signature:

```
(string -> string) map get_software_version (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsshautomode

Overview:

Get the sshautomode field of the given host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
bool get_ssh_auto_mode (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsshenabled

Overview:

Get the sshenabled field of the given host.

Signature:

```
bool get_ssh_enabled (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsshenabledtimeout

Overview:

Get the sshenabledtimeout field of the given host.

Signature:

```
int get_ssh_enabled_timeout (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getsshexpiry

Overview:

Get the sshexpiry field of the given host.

Signature:

```
datetime get_ssh_expiry (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `datetime`

value of the field

RPC name: getssllegacy

This message is deprecated.

Overview:

Get the ssllegacy field of the given host.

Signature:

```
bool get_ssl_legacy (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsupportedbootloaders

Overview:

Get the supportedbootloaders field of the given host.

Signature:

```
string set get_supported_bootloaders (session ref session_ref, host ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getsuspendimagesr

Overview:

Get the suspendimagesr field of the given host.

Signature:

```
SR ref get_suspend_image_sr (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getsystemstatuscapabilities

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_system_status_capabilities (session ref session_ref, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to interrogate

Minimum Role: read-only

Return Type: `string`

An XML fragment containing the system status capabilities.

RPC name: gettags

Overview:

Get the tags field of the given host.

Signature:

```
string set get_tags (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettimezone

Overview:

Get the timezone field of the given host.

Signature:

```
string get_timezone (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gettlsverificationenabled

Overview:

Get the tlsverificationenabled field of the given host.

Signature:

```
bool get_tls_verification_enabled (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: gettrackeduseragents

Overview:

Get the name, version list of tracked user agents on this host. If different versions of the same name are seen, keep the lastseen version. The oldest entry will be removed if reach the max num. Note that the list is cleared after host/XAPI restart

Signature:

```
(string -> string) map get_tracked_user_agents (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `(string -> string) map`

The name, version list of user agents that have been tracked on this host

RPC name: getueficertificates

This message is deprecated.

Overview:

Get the ueficertificates field of the given host.

Signature:

```
string get_uefi_certificates (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuncooperativeresidentVMs

This message is deprecated.

Overview:

Return a set of VMs which are not cooperating with the host's memory control system

Signature:

```
VM ref set get_uncooperative_resident_VMs (session ref session_ref, host
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host to query

Minimum Role: read-only

Return Type: `VM ref set`

VMs which are not cooperating

RPC name: getupdates

Overview:

Get the updates field of the given host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
pool_update ref set get_updates (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update ref set`

value of the field

RPC name: getupdatesrequiringreboot

Overview:

Get the updatesrequiringreboot field of the given host.

Signature:

```
pool_update ref set get_updates_requiring_reboot (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given host.

Signature:

```
string get_uuid (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvirtualhardwareplatformversions

Overview:

Get the virtualhardwareplatformversions field of the given host.

Signature:

```
int set get_virtual_hardware_platform_versions (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `int set`

value of the field

RPC name: getvmswhichpreventevacuation

Overview:

Return a set of VMs which prevent the host being evacuated, with perVM error codes

Signature:

```
(VM ref -> string set) map get_vms_which_prevent_evacuation (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host to query

Minimum Role: read-only

Return Type: `(VM ref -> string set) map`

VMs which block evacuation together with reasons

RPC name: hasextension

Overview:

Return true if the extension is available on the host

Signature:

```
bool has_extension (session ref session_ref, host ref host, string name)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`name`	The name of the API call

Minimum Role: pool-admin

Return Type: `bool`

True if the extension exists, false otherwise

RPC name: installservercertificate

Overview:

Install the TLS server certificate.

Signature:

```
void install_server_certificate (session ref session_ref, host ref host,
string certificate, string private_key, string certificate_chain)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`certificate`	The server certificate, in PEM form
`string`	`private_key`	The unencrypted private key used to sign the certificate, in PKCS#8 form
`string`	`certificate_chain`	The certificate chain, in PEM form

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: licenseadd

Overview:

Apply a new license to a host

Signature:

```
void license_add (session ref session_ref, host ref host, string contents)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to upload the license to
`string`	`contents`	The contents of the license file, base64 encoded

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `LICENSE_PROCESSING_ERROR`

RPC name: licenseapply

This message is removed.

Overview:

Apply a new license to a host

Signature:

```
void license_apply (session ref session_ref, host ref host, string
contents)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to upload the license to
`string`	`contents`	The contents of the license file, base64 encoded

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `LICENSE_PROCESSING_ERROR`

RPC name: licenseremove

Overview:

Remove any license file from the specified host, and switch that host to the unlicensed edition

Signature:

```
void license_remove (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host from which any license will be removed

Minimum Role: pool-operator

Return Type: `void`

RPC name: listmethods

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

List all supported methods

Signature:

```
string set list_methods (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `string set`

The name of every supported method.

RPC name: listtimezones

Overview:

List all available timezones on the host.

Signature:

```
string set list_timezones (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `string set`

The set of available timezones on the host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: localmanagementreconfigure

Overview:

Reconfigure the management network interface. Should only be used if Host.managementreconfigure is impossible because the network configuration is broken.

Signature:

```
void local_management_reconfigure (session ref session_ref, string
interface)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`interface`	name of the interface to use as a management interface

Minimum Role: pool-operator

Return Type: `void`

RPC name: managementdisable

Overview:

Disable the management network interface

Signature:

```
void management_disable (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: managementreconfigure

Overview:

Reconfigure the management network interface

Signature:

```
void management_reconfigure (session ref session_ref, PIF ref pif)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`pif`	reference to a PIF object corresponding to the management interface

Minimum Role: pool-operator

Return Type: `void`

RPC name: migratereceive

Overview:

Prepare to receive a VM, returning a token which can be passed to VM.migrate.

Signature:

```
(string -> string) map migrate_receive (session ref session_ref, host ref
host, network ref network, (string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`host`	The target host
`network ref`	`network`	The network through which migration traffic should be received.
`(string -> string) map`	`options`	Extra configuration operations

Minimum Role: vm-power-admin

Return Type: `(string -> string) map`

A value which should be passed to VM.migrate

RPC name: poweron

Overview:

Attempt to poweron the host if the capability exists.

Signature:

```
void power_on (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to power on

Minimum Role: pool-operator

Return Type: `void`

RPC name: querydatasource

Overview:

Query the latest value of the specified data source

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
float query_data_source (session ref session_ref, host ref host, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`data_source`	The data source to query

Minimum Role: read-only

Return Type: `float`

The latest value, averaged over the last 5 seconds

RPC name: reboot

Overview:

Reboot the host. This function can only be called if there are no currently running VMs on the host and it is disabled.

Signature:

```
void reboot (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to reboot

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: recorddatasource

Overview:

Start recording the specified data source

Signature:

```
void record_data_source (session ref session_ref, host ref host, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`data_source`	The data source to record

Minimum Role: pool-operator

Return Type: `void`

RPC name: refreshpackinfo

This message is deprecated.

Overview:

Refresh the list of installed Supplemental Packs.

Signature:

```
void refresh_pack_info (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to modify

Minimum Role: pool-operator

Return Type: `void`

RPC name: refreshservercertificate

Overview:

Replace the internal selfsigned host certficate with a new one.

Signature:

```
void refresh_server_certificate (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: removefromguestVCPUsparams

Overview:

Remove the given key and its corresponding value from the guestVCPUsparams field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_guest_VCPUs_params (session ref session_ref, host ref
self, string key)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromlicenseserver

Overview:

Remove the given key and its corresponding value from the licenseserver field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_license_server (session ref session_ref, host ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromlogging

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the logging field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_logging (session ref session_ref, host ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given host. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: rescandrivers

Overview:

Scan the host and update its driver information.

Signature:

```
void rescan_drivers (session ref session_ref, host ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host to be rescanned

Minimum Role: pool-admin

Return Type: `void`

RPC name: resetcpufeatures

This message is removed.

Overview:

Remove the feature mask, such that after a reboot all features of the CPU are enabled.

Signature:

```
void reset_cpu_features (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `void`

RPC name: resetservercertificate

Overview:

Delete the current TLS server certificate and replace by a new, selfsigned one. This should only be used with extreme care.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void reset_server_certificate (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: restartagent

Overview:

Restarts the agent after a 10 second pause. WARNING: this is a dangerous operation. Any operations in progress will be aborted, and unrecoverable data loss may occur. The caller is responsible for ensuring that there are no operations in progress when this method is called.

Signature:

```
void restart_agent (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host on which you want to restart the agent

Minimum Role: pool-operator

Return Type: `void`

RPC name: retrievewlbevacuaterecommendations

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Retrieves recommended host migrations to perform when evacuating the host from the wlb server. If a VM cannot be migrated from the host the reason is listed instead of a recommendation.

Signature:

```
(VM ref -> string set) map retrieve_wlb_evacuate_recommendations (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host to query

Minimum Role: read-only

Return Type: `(VM ref -> string set) map`

VMs and the reasons why they would block evacuation, or their target host recommended by the wlb server

RPC name: senddebugkeys

Overview:

Inject the given string as debugging keys into Xen

Signature:

```
void send_debug_keys (session ref session_ref, host ref host, string keys)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`keys`	The keys to send

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setaddress

Overview:

Set the address field of the given host.

Signature:

```
void set_address (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setconsoleidletimeout

Overview:

Set the console idle timeout for the host

Signature:

```
void set_console_idle_timeout (session ref session_ref, host ref self, int
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`int`	`value`	The console idle timeout in seconds

Minimum Role: pool-admin

Return Type: `void`

RPC name: setcpufeatures

This message is removed.

Overview:

Set the CPU features to be used after a reboot, if the given features string is valid.

Signature:

```
void set_cpu_features (session ref session_ref, host ref host, string features)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`features`	The features string (32 hexadecimal digits)

Minimum Role: pool-operator

Return Type: `void`

RPC name: setcrashdumpsr

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the crashdumpsr field of the given host.

Signature:

```
void set_crash_dump_sr (session ref session_ref, host ref self, SR ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setdisplay

Overview:

Set the display field of the given host.

Signature:

```
void set_display (session ref session_ref, host ref self, host_display
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`host_display`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setguestVCPUsparams

Overview:

Set the guestVCPUsparams field of the given host.

Signature:

```
void set_guest_VCPUs_params (session ref session_ref, host ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethostname

Overview:

Set the hostname field of the given host.

Signature:

```
void set_hostname (session ref session_ref, host ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethostnamelive

Overview:

Sets the host name to the specified string. Both the API and lowerlevel system hostname are changed immediately.

Signature:

```
void set_hostname_live (session ref session_ref, host ref host, string hostname)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose host name to set
`string`	`hostname`	The new host name

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `HOST_NAME_INVALID`

RPC name: sethttpsonly

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

updates the host firewall to open or close port 80 depending on the value

Signature:

```
void set_https_only (session ref session_ref, host ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The Host
`bool`	`value`	true - http port 80 will be blocked, false - http port 80 will be open

Minimum Role: pool-operator

Return Type: `void`

RPC name: setiscsiiqn

Overview:

Sets the initiator IQN for the host

Signature:

```
void set_iscsi_iqn (session ref session_ref, host ref host, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`value`	The value to which the IQN should be set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlicenseserver

Overview:

Set the licenseserver field of the given host.

Signature:

```
void set_license_server (session ref session_ref, host ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlogging

Overview:

Set the logging field of the given host.

Signature:

```
void set_logging (session ref session_ref, host ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setmaxcstate

Overview:

Sets xen's maxcstate on a host. See: https://xenbits.xen.org/docs/unstable/misc/xencommandline.htmlmaxcstatex86. "" means unlimited; "N" means limit to CN; "N,M" means limit to CN with max sub cstate M. Note: Only C0, C1, unlimited are supported currently.

Signature:

```
void set_max_cstate (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`string`	`value`	The max_cstate to apply to a host

Minimum Role: pool-operator

Return Type: `void`

RPC name: setmultipathing

Overview:

Specifies whether multipathing is enabled

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_multipathing (session ref session_ref, host ref host, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`bool`	`value`	Whether multipathing should be enabled

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given host.

Signature:

```
void set_name_description (session ref session_ref, host ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given host.

Signature:

```
void set_name_label (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setntpcustomservers

Overview:

Set the custom NTP servers for the host

Signature:

```
void set_ntp_custom_servers (session ref session_ref, host ref self,
string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	The host
`string set`	`value`	The set of custom NTP servers to configure

Minimum Role: pool-operator

Return Type: `void`

RPC name: setntpmode

Overview:

Set the NTP mode for the host

Signature:

```
void set_ntp_mode (session ref session_ref, host ref self, host_ntp_mode
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`host_ntp_mode`	`value`	The NTP mode to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnumaaffinitypolicy

Overview:

Set VM placement NUMA affinity policy

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_numa_affinity_policy (session ref session_ref, host ref self,
host_numa_affinity_policy value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`host_numa_affinity_policy`	`value`	The NUMA affinity policy to apply to a host

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given host.

Signature:

```
void set_other_config (session ref session_ref, host ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setpoweronmode

Overview:

Set the poweronmode, host, user and password

Signature:

```
void set_power_on_mode (session ref session_ref, host ref self, string
power_on_mode, (string -> string) map power_on_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`string`	`power_on_mode`	power-on-mode can be empty, wake-on-lan, IPMI or other
`(string -> string) map`	`power_on_config`	Power on config

Minimum Role: pool-operator

Return Type: `void`

RPC name: setschedgran

Overview:

Sets xen's schedgran on a host. See: https://xenbits.xen.org/docs/unstable/misc/xencommandline.htmlschedgranx86

Signature:

```
void set_sched_gran (session ref session_ref, host ref self,
host_sched_gran value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`host_sched_gran`	`value`	The sched-gran to apply to a host

Return Type: `void`

RPC name: setservertime

Overview:

Set the host's system clock when NTP is disabled.

Signature:

```
void set_servertime (session ref session_ref, host ref self, datetime
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`datetime`	`value`	A date/time to be set. When a timezone offset is missing, UTC is assumed.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsshautomode

Overview:

Set the SSH auto mode for the host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_ssh_auto_mode (session ref session_ref, host ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`bool`	`value`	The SSH auto mode for the host. When true, the SSH port is closed by default and it's open only in case the API is unavailable. When false, the SSH port is always open.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setsshenabledtimeout

Overview:

Set the SSH service enabled timeout for the host

Signature:

```
void set_ssh_enabled_timeout (session ref session_ref, host ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`int`	`value`	The SSH enabled timeout in seconds (0 means no timeout, max 2 days)

Minimum Role: pool-admin

Return Type: `void`

RPC name: setssllegacy

Overview:

Enable/disable SSLv3 for interoperability with older server versions. When this is set to a different value, the host immediately restarts its SSL/TLS listening service; typically this takes less than a second but existing connections to it will be broken. API login sessions will remain valid.

Signature:

```
void set_ssl_legacy (session ref session_ref, host ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`bool`	`value`	True to allow SSLv3 and ciphersuites as used in old XenServer versions

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsuspendimagesr

Overview:

Set the suspendimagesr field of the given host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_suspend_image_sr (session ref session_ref, host ref self, SR ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given host.

Signature:

```
void set_tags (session ref session_ref, host ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: settimezone

Overview:

Set the host's timezone.

Signature:

```
void set_timezone (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`string`	`value`	The time zone identifier as defined in the IANA Time Zone Database

Minimum Role: pool-operator

Return Type: `void`

RPC name: setueficertificates

This message is deprecated.

Overview:

Sets the UEFI certificates on a host

Signature:

```
void set_uefi_certificates (session ref session_ref, host ref host, string
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`value`	The certificates to apply to a host

Return Type: `void`

RPC name: shutdown

Overview:

Shutdown the host. This function can only be called if there are no currently running VMs on the host and it is disabled.

Signature:

```
void shutdown (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to shutdown

Minimum Role: pool-operator

Return Type: `void`

RPC name: shutdownagent

Overview:

Shuts the agent down after a 10 second pause. WARNING: this is a dangerous operation. Any operations in progress will be aborted, and unrecoverable data loss may occur. The caller is responsible for ensuring that there are no operations in progress when this method is called.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void shutdown_agent (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: syncdata

Overview:

This causes the synchronisation of the nondatabase data messages, RRDs and so on stored on the master to be synchronised with the host

Signature:

```
void sync_data (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to whom the data should be sent

Minimum Role: pool-admin

Return Type: `void`

RPC name: syslogreconfigure

Overview:

Reconfigure syslog logging

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void syslog_reconfigure (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Tell the host to reread its Host.logging parameters and reconfigure itself accordingly

Minimum Role: pool-operator

Return Type: `void`

Class: hostcpu

This class is deprecated.

A physical CPU

Fields for class: hostcpu

FIELD	TYPE	QUALIFIER	DESCRIPTION
`family`	`int`	*RO/runtime*	**Deprecated.** the family (number) of the physical CPU
`features`	`string`	*RO/runtime*	**Deprecated.** the physical CPU feature bitmap
`flags`	`string`	*RO/runtime*	**Deprecated.** the flags of the physical CPU (a decoded version of the features field)
`host`	`host ref`	*RO/runtime*	**Deprecated.** the host the CPU is in

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`model`	`int`	*RO/runtime*	**Deprecated.** the model number of the physical CPU
`modelname`	`string`	*RO/runtime*	**Deprecated.** the model name of the physical CPU
`number`	`int`	*RO/runtime*	**Deprecated.** the number of the physical CPU within the host
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`speed`	`int`	*RO/runtime*	**Deprecated.** the speed of the physical CPU
`stepping`	`string`	*RO/runtime*	**Deprecated.** the stepping of the physical CPU
`utilisation`	`float`	*RO/runtime*	**Deprecated.** the current CPU utilisation
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference
`vendor`	`string`	*RO/runtime*	**Deprecated.** the vendor of the physical CPU

RPCs associated with class: hostcpu

RPC name: addtootherconfig

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Add the given keyvalue pair to the otherconfig field of the given hostcpu.

Signature:

```
void add_to_other_config (session ref session_ref, host_cpu ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the hostcpus known to the system.

Signature:

```
host_cpu ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `host_cpu ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

Return a map of hostcpu references to hostcpu records for all hostcpus known to the system.

Signature:

```
(host_cpu ref -> host_cpu record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(host_cpu ref -> host_cpu record) map`

records of all objects

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the hostcpu instance with the specified UUID.

Signature:

```
host_cpu ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host_cpu ref`

reference to the object

RPC name: getfamily

This message is deprecated.

Overview:

Get the family field of the given hostcpu.

Signature:

```
int get_family (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getfeatures

This message is deprecated.

Overview:

Get the features field of the given hostcpu.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string get_features (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getflags

This message is deprecated.

Overview:

Get the flags field of the given hostcpu.

Signature:

```
string get_flags (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

This message is deprecated.

Overview:

Get the host field of the given hostcpu.

Signature:

```
host ref get_host (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getmodel

This message is deprecated.

Overview:

Get the model field of the given hostcpu.

Signature:

```
int get_model (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmodelname

This message is deprecated.

Overview:

Get the modelname field of the given hostcpu.

Signature:

```
string get_modelname (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnumber

This message is deprecated.

Overview:

Get the number field of the given hostcpu.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int get_number (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given hostcpu.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, host_cpu
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given hostcpu.

Signature:

```
host_cpu record get_record (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_cpu record`

all fields from the object

RPC name: getspeed

This message is deprecated.

Overview:

Get the speed field of the given hostcpu.

Signature:

```
int get_speed (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getstepping

This message is deprecated.

Overview:

Get the stepping field of the given hostcpu.

Signature:

```
string get_stepping (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getutilisation

This message is deprecated.

Overview:

Get the utilisation field of the given hostcpu.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
float get_utilisation (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given hostcpu.

Signature:

```
string get_uuid (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getvendor

This message is deprecated.

Overview:

Get the vendor field of the given hostcpu.

Signature:

```
string get_vendor (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given hostcpu. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host_cpu ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given hostcpu.

Signature:

```
void set_other_config (session ref session_ref, host_cpu ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: hostcrashdump

Represents a host crash dump

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: hostcrashdump

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	Host the crashdump relates to
`other_config`	`(string -> string) map`	*RW*	additional configuration
`size`	`int`	*RO/runtime*	Size of the crashdump
`timestamp`	`datetime`	*RO/runtime*	Time the crash happened
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: hostcrashdump

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given hostcrashdump.

Signature:

```
void add_to_other_config (session ref session_ref, host_crashdump ref
self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

Overview:

Destroy specified host crash dump, removing it from the disk.

Signature:

```
void destroy (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	The host crashdump to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the hostcrashdumps known to the system.

Signature:

```
host_crashdump ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `host_crashdump ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of hostcrashdump references to hostcrashdump records for all hostcrashdumps known to the system.

Signature:

```
(host_crashdump ref -> host_crashdump record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(host_crashdump ref -> host_crashdump record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the hostcrashdump instance with the specified UUID.

Signature:

```
host_crashdump ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host_crashdump ref`

reference to the object

RPC name: gethost

Overview:

Get the host field of the given hostcrashdump.

Signature:

```
host ref get_host (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given hostcrashdump.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_other_config (session ref session_ref,
host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given hostcrashdump.

Signature:

```
host_crashdump record get_record (session ref session_ref, host_crashdump
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_crashdump record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsize

Overview:

Get the size field of the given hostcrashdump.

Signature:

```
int get_size (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gettimestamp

Overview:

Get the timestamp field of the given hostcrashdump.

Signature:

```
datetime get_timestamp (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `datetime`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given hostcrashdump.

Signature:

```
string get_uuid (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given hostcrashdump. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host_crashdump ref
self, string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given hostcrashdump.

Signature:

```
void set_other_config (session ref session_ref, host_crashdump ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: upload

Overview:

Upload the specified host crash dump to a specified URL

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void upload (session ref session_ref, host_crashdump ref self, string url,
(string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	The host crashdump to upload
`string`	`url`	The URL to upload to
`(string -> string) map`	`options`	Extra configuration operations

Minimum Role: pool-operator

Return Type: `void`

Class: Hostdriver

UNSUPPORTED. A multiversion driver on a host

Fields for class: Hostdriver

FIELD	TYPE	QUALIFIER	DESCRIPTION
`active_variant`	`Driver_variant ref`	*RO/runtime*	Currently active variant of this driver, if any, or Null.
`description`	`string`	*RO/runtime*	Description of the driver
`friendly_name`	`string`	*RO/runtime*	Descriptive name, not used for identification
`host`	`host ref`	*RO/constructor*	Host where this driver is installed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`info`	`string`	*RO/runtime*	Information about the driver
`name`	`string`	*RO/runtime*	Name identifying the driver uniquely
`selected_variant`	`Driver_variant ref`	*RO/runtime*	Variant (if any) selected to become active after reboot. Or Null
`type`	`string`	*RO/runtime*	Device type this driver supports, like network or storage
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`variants`	`Driver_variant ref set`	*RO/runtime*	Variants of this driver available for selection

RPCs associated with class: Hostdriver

RPC name: deselect

Overview:

UNSUPPORTED. Deselect the currently active variant of this driver after reboot. No action will be taken if no variant is currently active.

Signature:

```
void deselect (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Host_driver ref`	`self`	Driver to become inactive (after reboot).

Minimum Role: pool-admin

Return Type: `void`

RPC name: getactivevariant

Overview:

Get the activevariant field of the given Hostdriver.

Signature:

```
Driver_variant ref get_active_variant (session ref session_ref,
Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant ref`

value of the field

RPC name: getall

Overview:

Return a list of all the Hostdrivers known to the system.

Signature:

```
Host_driver ref set get_all (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Host_driver ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Hostdriver references to Hostdriver records for all Hostdrivers known to the system.

Signature:

```
(Host_driver ref -> Host_driver record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Host_driver ref -> Host_driver record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the Hostdriver instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Host_driver ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Host_driver ref`

reference to the object

RPC name: getdescription

Overview:

Get the description field of the given Hostdriver.

Signature:

```
string get_description (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfriendlyname

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the friendlyname field of the given Hostdriver.

Signature:

```
string get_friendly_name (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given Hostdriver.

Signature:

```
host ref get_host (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getinfo

Overview:

Get the info field of the given Hostdriver.

Signature:

```
string get_info (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getname

Overview:

Get the name field of the given Hostdriver.

Signature:

```
string get_name (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Hostdriver.

Signature:

```
Host_driver record get_record (session ref session_ref, Host_driver ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Host_driver record`

all fields from the object

RPC name: getselectedvariant

Overview:

Get the selectedvariant field of the given Hostdriver.

Signature:

```
Driver_variant ref get_selected_variant (session ref session_ref,
Host_driver ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant ref`

value of the field

RPC name: gettype

Overview:

Get the type field of the given Hostdriver.

Signature:

```
string get_type (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Hostdriver.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvariants

Overview:

Get the variants field of the given Hostdriver.

Signature:

```
Driver_variant ref set get_variants (session ref session_ref, Host_driver
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: rescan

Overview:

UNSUPPORTED. Rescan a host's drivers and update information about them. This is mostly for trouble shooting.

Signature:

```
void rescan (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Update driver information of this host.

Minimum Role: pool-admin

Return Type: `void`

RPC name: select

Overview:

UNSUPPORTED. Select a variant of the driver to become active after reboot or immediately if currently no version is active

Signature:

```
void select (session ref session_ref, Host_driver ref self, Driver_variant
ref variant)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Host_driver ref`	`self`	Driver to become active (after reboot).
`Driver_variant ref`	`variant`	Driver version to become active (after reboot).

Minimum Role: pool-admin

Return Type: `void`

Class: hostmetrics

The metrics associated with a host

Fields for class: hostmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated
`live`	`bool`	*RO/runtime*	Pool master thinks this host is live
`memory_free`	`int`	*RO/runtime*	**Removed.** Free host memory (bytes)
`memory_total`	`int`	*RO/runtime*	Total host memory (bytes)
`other_config`	`(string -> string) map`	*RW*	additional configuration
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: hostmetrics

RPC name: addtootherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Add the given keyvalue pair to the otherconfig field of the given hostmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, host_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the hostmetrics instances known to the system.

Signature:

```
host_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host_metrics ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

references to all objects

RPC name: getallrecords

Overview:

Return a map of hostmetrics references to hostmetrics records for all hostmetrics instances known to the system.

Signature:

```
(host_metrics ref -> host_metrics record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(host_metrics ref -> host_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the hostmetrics instance with the specified UUID.

Signature:

```
host_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `host_metrics ref`

reference to the object

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given hostmetrics.

Signature:

```
datetime get_last_updated (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlive

Overview:

Get the live field of the given hostmetrics.

Signature:

```
bool get_live (session ref session_ref, host_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmemoryfree

This message is removed.

Overview:

Get the memory/free field of the given hostmetrics.

Signature:

```
int get_memory_free (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorytotal

Overview:

Get the memory/total field of the given hostmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int get_memory_total (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given hostmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

Overview:

Get a record containing the current state of the given hostmetrics.

Signature:

```
host_metrics record get_record (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_metrics record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given hostmetrics.

Signature:

```
string get_uuid (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given hostmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given hostmetrics.

Signature:

```
void set_other_config (session ref session_ref, host_metrics ref self,
(string -> string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: hostpatch

This class is deprecated.

Represents a patch stored on a server

Fields for class: hostpatch

FIELD	TYPE	QUALIFIER	DESCRIPTION
`applied`	`bool`	*RO/runtime*	**Deprecated.** True if the patch has been applied
`host`	`host ref`	*RO/constructor*	**Deprecated.** Host the patch relates to
`name_description`	`string`	*RO/constructor*	**Deprecated.** a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	**Deprecated.** a human-readable name
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`pool_patch`	`pool_patch ref`	*RO/constructor*	**Deprecated.** The patch applied
`size`	`int`	*RO/runtime*	**Deprecated.** Size of the patch
`timestamp_applied`	`datetime`	*RO/runtime*	**Deprecated.** Time the patch was applied
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference
`version`	`string`	*RO/constructor*	**Deprecated.** Patch version number

RPCs associated with class: hostpatch

RPC name: addtootherconfig

This message is deprecated.

Overview:

Add the given keyvalue pair to the otherconfig field of the given hostpatch.

Signature:

```
void add_to_other_config (session ref session_ref, host_patch ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: apply

This message is deprecated.

Overview:

Apply the selected patch and return its output

Signature:

```
string apply (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	The patch to apply

Minimum Role: pool-operator

Return Type: `string`

the output of the patch application process

RPC name: destroy

This message is deprecated.

Overview:

Destroy the specified host patch, removing it from the disk. This does NOT reverse the patch

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void destroy (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	The patch to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the hostpatchs known to the system.

Signature:

```
host_patch ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host_patch ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return a map of hostpatch references to hostpatch records for all hostpatchs known to the system.

Signature:

```
(host_patch ref -> host_patch record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(host_patch ref -> host_patch record) map`

records of all objects

RPC name: getapplied

This message is deprecated.

Overview:

Get the applied field of the given hostpatch.

Signature:

```
bool get_applied (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getbynamelabel

This message is deprecated.

Overview:

Get all the hostpatch instances with the given label.

Signature:

```
host_patch ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `host_patch ref set`

references to objects with matching names

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the hostpatch instance with the specified UUID.

Signature:

```
host_patch ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host_patch ref`

reference to the object

RPC name: gethost

This message is deprecated.

Overview:

Get the host field of the given hostpatch.

Signature:

```
host ref get_host (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getnamedescription

This message is deprecated.

Overview:

Get the name/description field of the given hostpatch.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string get_name_description (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

This message is deprecated.

Overview:

Get the name/label field of the given hostpatch.

Signature:

```
string get_name_label (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given hostpatch.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpoolpatch

This message is deprecated.

Overview:

Get the poolpatch field of the given hostpatch.

Signature:

```
pool_patch ref get_pool_patch (session ref session_ref, host_patch ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_patch ref`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given hostpatch.

Signature:

```
host_patch record get_record (session ref session_ref, host_patch ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_patch record`

all fields from the object

RPC name: getsize

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the size field of the given hostpatch.

Signature:

```
int get_size (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gettimestampapplied

This message is deprecated.

Overview:

Get the timestampapplied field of the given hostpatch.

Signature:

```
datetime get_timestamp_applied (session ref session_ref, host_patch ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `datetime`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given hostpatch.

Signature:

```
string get_uuid (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

This message is deprecated.

Overview:

Get the version field of the given hostpatch.

Signature:

```
string get_version (session ref session_ref, host_patch ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given hostpatch. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host_patch ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the otherconfig field of the given hostpatch.

Signature:

```
void set_other_config (session ref session_ref, host_patch ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: LVHD

LVHD SR specific operations

Fields for class: LVHD

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: LVHD

RPC name: enablethinprovisioning

Overview:

Upgrades an LVHD SR to enable thinprovisioning. Future VDIs created in this SR will be thinlyprovisioned, although existing VDIs will be left alone. Note that the SR must be attached to the SRmaster for upgrade to work.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string enable_thin_provisioning (session ref session_ref, host ref host,
SR ref SR, int initial_allocation, int allocation_quantum)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The LVHD Host to upgrade to being thin-provisioned.
`SR ref`	`SR`	The LVHD SR to upgrade to being thin-provisioned.
`int`	`initial_allocation`	The initial amount of space to allocate to a newly-created VDI in bytes
`int`	`allocation_quantum`	The amount of space to allocate to a VDI when it needs to be enlarged in bytes

Minimum Role: pool-admin

Return Type: `string`

Message from LVHD.enablethinprovisioning extension

RPC name: getbyuuid

Overview:

Get a reference to the LVHD instance with the specified UUID.

Signature:

```
LVHD ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `LVHD ref`

reference to the object

RPC name: getrecord

Overview:

Get a record containing the current state of the given LVHD.

Signature:

```
LVHD record get_record (session ref session_ref, LVHD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`LVHD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `LVHD record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given LVHD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, LVHD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`LVHD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: message

An message for the attention of the administrator

Fields for class: message

FIELD	TYPE	QUALIFIER	DESCRIPTION
`body`	`string`	*RO/runtime*	The body of the message
`cls`	`cls`	*RO/runtime*	The class of the object this message is associated with
`name`	`string`	*RO/runtime*	The name of the message
`obj_uuid`	`string`	*RO/runtime*	The uuid of the object this message is associated with
`priority`	`int`	*RO/runtime*	The message priority, 0 being low priority

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`timestamp`	`datetime`	*RO/runtime*	The time at which the message was created
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: message

RPC name: create

Overview:

Signature:

```
message ref create (session ref session_ref, string name, int priority,
cls cls, string obj_uuid, string body)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	The name of the message
`int`	`priority`	The priority of the message
`cls`	`cls`	The class of object this message is associated with
`string`	`obj_uuid`	The uuid of the object this message is associated with
`string`	`body`	The body of the message

Minimum Role: pool-operator

Return Type: `message ref`

The reference of the created message

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, message ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`message ref`	`self`	The reference of the message to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroyall

Overview:

Signature:

```
void destroy_all (session ref session_ref, (string -> string) map filters)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`(string -> string) map`	`filters`	Optional filters identifying messages to destroy: before (RFC3339 DateTime, destroy messages dated before this timestamp), after (RFC3339 DateTime, destroy messages dated after this timestamp), and priority (int, only destroy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
		messages with this priority). All provided conditions must be met (logical AND). If no filters are provided, all messages are destroyed. If no timezone is specified in a timestamp, UTC is assumed.

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroymany

Overview:

Signature:

```
void destroy_many (session ref session_ref, message ref set messages)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`message ref set`	`messages`	Messages to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: get

Overview:

Signature:

```
(message ref -> message record) map get (session ref session_ref, cls cls,
string obj_uuid, datetime since)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`cls`	`cls`	The class of object
`string`	`obj_uuid`	The uuid of the object
`datetime`	`since`	The cutoff time. When the timezone is missing, UTC is assumed

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The relevant messages

RPC name: getall

Overview:

Signature:

```
message ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `message ref set`

The references to the messages

RPC name: getallrecords

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(message ref -> message record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The messages

RPC name: getallrecordswhere

Overview:

Signature:

```
(message ref -> message record) map get_all_records_where (session ref
session_ref, string expr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`expr`	The expression to match

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The messages

RPC name: getbyuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
message ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	The uuid of the message

Minimum Role: read-only

Return Type: `message ref`

The message reference

RPC name: getrecord

Overview:

Signature:

```
message record get_record (session ref session_ref, message ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`message ref`	`self`	The reference to the message

Minimum Role: read-only

Return Type: `message record`

The message record

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsince

Overview:

Signature:

```
(message ref -> message record) map get_since (session ref session_ref,
datetime since)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`datetime`	`since`	The cutoff time. When the timezone is missing, UTC is assumed

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The relevant messages

Class: network

A virtual network

Fields for class: network

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`network_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`assigned_ips`	`(VIF ref -> string) map`	*RO/runtime*	The IP addresses assigned to VIFs on networks that have active xapi-managed DHCP
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this network
`bridge`	`string`	*RO/constructor*	name of the bridge corresponding to this network on the local host
`current_operations`	`(string -> network_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`default_locking_mode`	`network_default_locking_mode`	*RO/runtime*	The network will use this value to determine the behaviour of all VIFs where locking_mode = default
`managed`	`bool`	*RO/constructor*	true if the bridge is managed by xapi
`MTU`	`int`	*RW*	MTU in octets
`name_description`	`string`	*RW*	a notes field containing human-readable description

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_label`	`string`	*RW*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`PIFs`	`PIF ref set`	*RO/runtime*	list of connected pifs
`purpose`	`network_purpose set`	*RO/runtime*	Set of purposes for which the server will use this network
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VIFs`	`VIF ref set`	*RO/runtime*	list of connected vifs

RPCs associated with class: network

RPC name: addpurpose

Overview:

Give a network a new purpose if not present already

Signature:

```
void add_purpose (session ref session_ref, network ref self,
network_purpose value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	The network

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`network_purpose`	`value`	The purpose to add

Minimum Role: pool-admin

Return Type: `void`

Possible Error Codes: `NETWORK_INCOMPATIBLE_PURPOSES`

RPC name: addtags

Overview:

Add the given value to the tags field of the given network. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, network ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_other_config (session ref session_ref, network ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new network instance, and return its handle.

Signature:

```
network ref create (session ref session_ref, network record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `network ref`

reference to the newly created object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this pool

Signature:

```
blob ref create_new_blob (session ref session_ref, network ref network,
string name, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	The network
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: destroy

Overview:

Destroy the specified network instance.

Signature:

```
void destroy (session ref session_ref, network ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the networks known to the system.

Signature:

```
network ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `network ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of network references to network records for all networks known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(network ref -> network record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(network ref -> network record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given network.

Signature:

```
network_operations set get_allowed_operations (session ref session_ref,
network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_operations set`

value of the field

RPC name: getassignedips

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the assignedips field of the given network.

Signature:

```
(VIF ref -> string) map get_assigned_ips (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(VIF ref -> string) map`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given network.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getbridge

Overview:

Get the bridge field of the given network.

Signature:

```
string get_bridge (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbynamelabel

Overview:

Get all the network instances with the given label.

Signature:

```
network ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `network ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the network instance with the specified UUID.

Signature:

```
network ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `network ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given network.

Signature:

```
(string -> network_operations) map get_current_operations (session ref
session_ref, network ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> network_operations) map`

value of the field

RPC name: getdefaultlockingmode

Overview:

Get the defaultlockingmode field of the given network.

Signature:

```
network_default_locking_mode get_default_locking_mode (session ref
session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_default_locking_mode`

value of the field

RPC name: getmanaged

Overview:

Get the managed field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_managed (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getMTU

Overview:

Get the MTU field of the given network.

Signature:

```
int get_MTU (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnamedescription

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the name/description field of the given network.

Signature:

```
string get_name_description (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given network.

Signature:

```
string get_name_label (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given network.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPIFs

Overview:

Get the PIFs field of the given network.

Signature:

```
PIF ref set get_PIFs (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `PIF ref set`

value of the field

RPC name: getpurpose

Overview:

Get the purpose field of the given network.

Signature:

```
network_purpose set get_purpose (session ref session_ref, network ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_purpose set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given network.

Signature:

```
network record get_record (session ref session_ref, network ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network record`

all fields from the object

RPC name: gettags

Overview:

Get the tags field of the given network.

Signature:

```
string set get_tags (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVIFs

Overview:

Get the VIFs field of the given network.

Signature:

```
VIF ref set get_VIFs (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF ref set`

value of the field

RPC name: removefromotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given network. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, network ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removepurpose

Overview:

Remove a purpose from a network if present

Signature:

```
void remove_purpose (session ref session_ref, network ref self,
network_purpose value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	The network
`network_purpose`	`value`	The purpose to remove

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given network. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, network ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: setdefaultlockingmode

Overview:

Set the default locking mode for VIFs attached to this network

Signature:

```
void set_default_locking_mode (session ref session_ref, network ref
network, network_default_locking_mode value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	The network
`network_default_locking_mode`	`value`	The default locking mode for VIFs attached to this network.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setMTU

Overview:

Set the MTU field of the given network.

Signature:

```
void set_MTU (session ref session_ref, network ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_name_description (session ref session_ref, network ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given network.

Signature:

```
void set_name_label (session ref session_ref, network ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given network.

Signature:

```
void set_other_config (session ref session_ref, network ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given network.

Signature:

```
void set_tags (session ref session_ref, network ref self, string set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

Class: networksriov

networksriov which connects logical pif and physical pif

Fields for class: networksriov

FIELD	TYPE	QUALIFIER	DESCRIPTION
`configuration_mode`	`sriov_configuration_mode`	*RO/runtime*	The mode for configure network sriov
`logical_PIF`	`PIF ref`	*RO/constructor*	The logical PIF to connect to the SR-IOV network after enable SR-IOV on the physical PIF
`physical_PIF`	`PIF ref`	*RO/constructor*	The PIF that has SR-IOV enabled
`requires_reboot`	`bool`	*RO/runtime*	Indicates whether the host need to be rebooted before SR-IOV is enabled on the physical PIF
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: networksriov

RPC name: create

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable SRIOV on the specific PIF. It will create a networksriov based on the specific PIF and automatically create a logical PIF to connect the specific network.

Signature:

```
network_sriov ref create (session ref session_ref, PIF ref pif, network ref network)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`pif`	PIF on which to enable SR-IOV
`network ref`	`network`	Network to connect SR-IOV virtual functions with VM VIFs

Minimum Role: pool-operator

Return Type: `network_sriov ref`

The reference of the created networksriov object

RPC name: destroy

Overview:

Disable SRIOV on the specific PIF. It will destroy the networksriov and the logical PIF accordingly.

Signature:

```
void destroy (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`network_sriov ref`	`self`	SRIOV to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the networksriovs known to the system.

Signature:

```
network_sriov ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `network_sriov ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of networksriov references to networksriov records for all networksriovs known to the system.

Signature:

```
(network_sriov ref -> network_sriov record) map get_all_records (session
ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(network_sriov ref -> network_sriov record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the networksriov instance with the specified UUID.

Signature:

```
network_sriov ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `network_sriov ref`

reference to the object

RPC name: getconfigurationmode

Overview:

Get the configurationmode field of the given networksriov.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
sriov_configuration_mode get_configuration_mode (session ref session_ref,
network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `sriov_configuration_mode`

value of the field

RPC name: getlogicalPIF

Overview:

Get the logicalPIF field of the given networksriov.

Signature:

```
PIF ref get_logical_PIF (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getphysicalPIF

Overview:

Get the physicalPIF field of the given networksriov.

Signature:

```
PIF ref get_physical_PIF (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given networksriov.

Signature:

```
network_sriov record get_record (session ref session_ref, network_sriov
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `network_sriov record`

all fields from the object

RPC name: getremainingcapacity

Overview:

Get the number of free SRIOV VFs on the associated PIF

Signature:

```
int get_remaining_capacity (session ref session_ref, network_sriov ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	the NETWORK_SRIOV object

Minimum Role: read-only

Return Type: `int`

The number of free SRIOV VFs on the associated PIF

RPC name: getrequiresreboot

Overview:

Get the requiresreboot field of the given networksriov.

Signature:

```
bool get_requires_reboot (session ref session_ref, network_sriov ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given networksriov.

Signature:

```
string get_uuid (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: Observer

Describes an observer which will control observability activity in the Toolstack

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Observer

FIELD	TYPE	QUALIFIER	DESCRIPTION
`attributes`	`(string -> string) map`	*RO/constructor*	Attributes that observer will add to the data they produce
`components`	`string set`	*RO/constructor*	The list of xenserver components the observer will broadcast. An empty list means all components
`enabled`	`bool`	*RO/constructor*	This denotes if the observer is enabled. true if it is enabled and false if it is disabled
`endpoints`	`string set`	*RO/constructor*	The list of endpoints where data is exported to. Each endpoint is a URL or the string 'bugtool' refering to the internal logs
`hosts`	`host ref set`	*RO/constructor*	The list of hosts the observer is active on. An empty list means all hosts
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: Observer

RPC name: create

Overview:

Create a new Observer instance, and return its handle.

Signature:

```
Observer ref create (session ref session_ref, Observer record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer record`	`args`	All constructor arguments

Minimum Role: pool-admin

Return Type: `Observer ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified Observer instance.

Signature:

```
void destroy (session ref session_ref, Observer ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the Observers known to the system.

Signature:

```
Observer ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Observer ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Observer references to Observer records for all Observers known to the system.

Signature:

```
(Observer ref -> Observer record) map get_all_records (session ref
session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Observer ref -> Observer record) map`

records of all objects

RPC name: getattributes

Overview:

Get the attributes field of the given Observer.

Signature:

```
(string -> string) map get_attributes (session ref session_ref, Observer
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getbynamelabel

Overview:

Get all the Observer instances with the given label.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Observer ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `Observer ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the Observer instance with the specified UUID.

Signature:

```
Observer ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Observer ref`

reference to the object

RPC name: getcomponents

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the components field of the given Observer.

Signature:

```
string set get_components (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getenabled

Overview:

Get the enabled field of the given Observer.

Signature:

```
bool get_enabled (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getendpoints

Overview:

Get the endpoints field of the given Observer.

Signature:

```
string set get_endpoints (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gethosts

Overview:

Get the hosts field of the given Observer.

Signature:

```
host ref set get_hosts (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `host ref set`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given Observer.

Signature:

```
string get_name_description (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given Observer.

Signature:

```
string get_name_label (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Observer.

Signature:

```
Observer record get_record (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Observer record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given Observer.

Signature:

```
string get_uuid (session ref session_ref, Observer ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: setattributes

Overview:

Set the attributes of an observer. These are used to emit metadata by the observer

Signature:

```
void set_attributes (session ref session_ref, Observer ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`(string -> string) map`	`value`	The attributes that the observer emits as part of the data

Minimum Role: pool-admin

Return Type: `void`

RPC name: setcomponents

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the components on which the observer will broadcast to. i.e. xapi, xenopsd, networkd, etc.

Signature:

```
void set_components (session ref session_ref, Observer ref self, string
set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`string set`	`value`	The components the observer will broadcast to

Minimum Role: pool-admin

Return Type: `void`

RPC name: setenabled

Overview:

Enable / disable this observer which will stop the observer from producing observability information

Signature:

```
void set_enabled (session ref session_ref, Observer ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	If the observer is to be enabled (true) or disabled (false)

Minimum Role: pool-admin

Return Type: `void`

RPC name: setendpoints

Overview:

Set the file/HTTP endpoints the observer sends data to

Signature:

```
void set_endpoints (session ref session_ref, Observer ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`string set`	`value`	The endpoints that the observer will export data to. A URL or the string 'bugtool'. This can refer to an enpoint to the local file system

Minimum Role: pool-admin

Return Type: `void`

RPC name: sethosts

Overview:

Sets the hosts that the observer is to be registered on

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_hosts (session ref session_ref, Observer ref self, host ref set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`host ref set`	`value`	Hosts the observer is registered on

Minimum Role: pool-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given Observer.

Signature:

```
void set_name_description (session ref session_ref, Observer ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given Observer.

Signature:

```
void set_name_label (session ref session_ref, Observer ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

Class: PBD

The physical block devices through which hosts access SRs

Fields for class: PBD

FIELD	TYPE	QUALIFIER	DESCRIPTION
`currently_attached`	`bool`	*RO/runtime*	is the SR currently attached on this host?
`device_config`	`(string -> string) map`	*RO/constructor*	a config string to string map that is provided to the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			host's SR-backend-driver
`host`	`host ref`	*RO/constructor*	physical machine on which the pbd is available
`other_config`	`(string -> string) map`	*RW*	additional configuration
`SR`	`SR ref`	*RO/constructor*	the storage repository that the pbd realises
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PBD

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PBD.

Signature:

```
void add_to_other_config (session ref session_ref, PBD ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new PBD instance, and return its handle.

Signature:

```
PBD ref create (session ref session_ref, PBD record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `PBD ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified PBD instance.

Signature:

```
void destroy (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PBD ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PBDs known to the system.

Signature:

```
PBD ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PBD ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PBD references to PBD records for all PBDs known to the system.

Signature:

```
(PBD ref -> PBD record) map get_all_records (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PBD ref -> PBD record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PBD instance with the specified UUID.

Signature:

```
PBD ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PBD ref`

reference to the object

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given PBD.

Signature:

```
bool get_currently_attached (session ref session_ref, PBD ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdeviceconfig

Overview:

Get the deviceconfig field of the given PBD.

Signature:

```
(string -> string) map get_device_config (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethost

Overview:

Get the host field of the given PBD.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
host ref get_host (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PBD.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PBD ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

Overview:

Get a record containing the current state of the given PBD.

Signature:

```
PBD record get_record (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PBD record`

all fields from the object

RPC name: getSR

Overview:

Get the SR field of the given PBD.

Signature:

```
SR ref get_SR (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `SR ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PBD.

Signature:

```
string get_uuid (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: plug

Overview:

Activate the specified PBD, causing the referenced SR to be attached and scanned

Signature:

```
void plug (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PBD ref`	`self`	The PBD to activate

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `SR_UNKNOWN_DRIVER`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PBD. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PBD ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setdeviceconfig

Overview:

Sets the PBD's deviceconfig field

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_device_config (session ref session_ref, PBD ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	The PBD to modify
`(string -> string) map`	`value`	The new value of the PBD's device_config

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PBD.

Signature:

```
void set_other_config (session ref session_ref, PBD ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: unplug

Overview:

Deactivate the specified PBD, causing the referenced SR to be detached and nolonger scanned

Signature:

```
void unplug (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	The PBD to deactivate

Minimum Role: pool-operator

Return Type: `void`

Class: PCI

A PCI device

Fields for class: PCI

FIELD	TYPE	QUALIFIER	DESCRIPTION
`class_id`	`string`	*RO/constructor*	PCI class ID
`class_name`	`string`	*RO/constructor*	PCI class name
`dependencies`	`PCI ref set`	*RO/runtime*	List of dependent PCI devices
`device_id`	`string`	*RO/constructor*	Device ID
`device_name`	`string`	*RO/constructor*	Device name
`driver_name`	`string`	*RO/constructor*	Driver name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	Physical machine that owns the PCI device
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`pci_id`	`string`	*RO/constructor*	PCI ID of the physical device
`subsystem_device_id`	`string`	*RO/constructor*	Subsystem device ID
`subsystem_device_name`	`string`	*RO/constructor*	Subsystem device name
`subsystem_vendor_id`	`string`	*RO/constructor*	Subsystem vendor ID
`subsystem_vendor_name`	`string`	*RO/constructor*	Subsystem vendor name
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_id`	`string`	*RO/constructor*	Vendor ID
`vendor_name`	`string`	*RO/constructor*	Vendor name

RPCs associated with class: PCI

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PCI.

Signature:

```
void add_to_other_config (session ref session_ref, PCI ref self, string key, string value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: disabledom0access

Overview:

Hide a PCI device from the dom0 kernel. Takes affect after next boot.

Signature:

```
pci_dom0_access disable_dom0_access (session ref session_ref, PCI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	The PCI to hide

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PCI from dom0

RPC name: enabledom0access

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Unhide a PCI device from the dom0 kernel. Takes affect after next boot.

Signature:

```
pci_dom0_access enable_dom0_access (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	The PCI to unhide

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PCI from dom0

RPC name: getall

Overview:

Return a list of all the PCIs known to the system.

Signature:

```
PCI ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PCI ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of PCI references to PCI records for all PCIs known to the system.

Signature:

```
(PCI ref -> PCI record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PCI ref -> PCI record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PCI instance with the specified UUID.

Signature:

```
PCI ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PCI ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the object

RPC name: getclassid

Overview:

Get the classid field of the given PCI.

Signature:

```
string get_class_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getclassname

Overview:

Get the classname field of the given PCI.

Signature:

```
string get_class_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdependencies

Overview:

Get the dependencies field of the given PCI.

Signature:

```
PCI ref set get_dependencies (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref set`

value of the field

RPC name: getdeviceid

Overview:

Get the deviceid field of the given PCI.

Signature:

```
string get_device_id (session ref session_ref, PCI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdevicename

Overview:

Get the devicename field of the given PCI.

Signature:

```
string get_device_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdom0accessstatus

Overview:

Return a PCI device dom0 access status.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
pci_dom0_access get_dom0_access_status (session ref session_ref, PCI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	The PCI

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PCI from dom0

RPC name: getdrivername

Overview:

Get the drivername field of the given PCI.

Signature:

```
string get_driver_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given PCI.

Signature:

```
host ref get_host (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PCI.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PCI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpciid

Overview:

Get the pciid field of the given PCI.

Signature:

```
string get_pci_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PCI.

Signature:

```
PCI record get_record (session ref session_ref, PCI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI record`

all fields from the object

RPC name: getsubsystemdeviceid

Overview:

Get the subsystemdeviceid field of the given PCI.

Signature:

```
string get_subsystem_device_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsubsystemdevicename

Overview:

Get the subsystemdevicename field of the given PCI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_subsystem_device_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsubsystemvendorid

Overview:

Get the subsystemvendorid field of the given PCI.

Signature:

```
string get_subsystem_vendor_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsubsystemvendorname

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the subsystemvendorname field of the given PCI.

Signature:

```
string get_subsystem_vendor_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PCI.

Signature:

```
string get_uuid (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getvendorid

Overview:

Get the vendorid field of the given PCI.

Signature:

```
string get_vendor_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorname

Overview:

Get the vendorname field of the given PCI.

Signature:

```
string get_vendor_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PCI. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PCI ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PCI.

Signature:

```
void set_other_config (session ref session_ref, PCI ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: PGPU

A physical GPU pGPU

Fields for class: PGPU

FIELD	TYPE	QUALIFIER	DESCRIPTION
`compatibility_metadata`	`(string -> string) map`	*RO/runtime*	PGPU metadata to determine whether a VGPU can migrate between two PGPUs
`dom0_access`	`pci_dom0_access`	*RO/runtime*	**Deprecated.** The accessibility of this device from dom0
`enabled_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	List of VGPU types which have been enabled for this PGPU
`GPU_group`	`GPU_group ref`	*RO/constructor*	GPU group the pGPU is contained in
`host`	`host ref`	*RO/runtime*	Host that owns the GPU
`is_system_display_device`	`bool`	*RO/runtime*	Is this device the system display device

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PCI`	`PCI ref`	*RO/constructor*	Link to underlying PCI device
`resident_VGPUs`	`VGPU ref set`	*RO/runtime*	List of VGPUs running on this PGPU
`supported_VGPU_max_capacities`	`(VGPU_type ref -> int) map`	*RO/runtime*	A map relating each VGPU type supported on this GPU to the maximum number of VGPUs of that type which can run simultaneously on this GPU
`supported_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	List of VGPU types supported by the underlying hardware
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PGPU

RPC name: addenabledVGPUtypes

Overview:

Signature:

```
void add_enabled_VGPU_types (session ref session_ref, PGPU ref self,
VGPU_type ref value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to which we are adding an enabled VGPU type
`VGPU_type ref`	`value`	The VGPU type to enable

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PGPU.

Signature:

```
void add_to_other_config (session ref session_ref, PGPU ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: disabledom0access

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Signature:

```
pci_dom0_access disable_dom0_access (session ref session_ref, PGPU ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to which dom0 will be denied access

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PGPU from dom0

RPC name: enabledom0access

This message is deprecated.

Overview:

Signature:

```
pci_dom0_access enable_dom0_access (session ref session_ref, PGPU ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to which dom0 will be granted access

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `pci_dom0_access`

The accessibility of this PGPU from dom0

RPC name: getall

Overview:

Return a list of all the PGPUs known to the system.

Signature:

```
PGPU ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PGPU ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PGPU references to PGPU records for all PGPUs known to the system.

Signature:

```
(PGPU ref -> PGPU record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(PGPU ref -> PGPU record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PGPU instance with the specified UUID.

Signature:

```
PGPU ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PGPU ref`

reference to the object

RPC name: getcompatibilitymetadata

Overview:

Get the compatibilitymetadata field of the given PGPU.

Signature:

```
(string -> string) map get_compatibility_metadata (session ref
session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getdom0access

This message is deprecated.

Overview:

Get the dom0access field of the given PGPU.

Signature:

```
pci_dom0_access get_dom0_access (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pci_dom0_access`

value of the field

RPC name: getenabledVGPUtypes

Overview:

Get the enabledVGPUtypes field of the given PGPU.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VGPU_type ref set get_enabled_VGPU_types (session ref session_ref, PGPU
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getGPUgroup

Overview:

Get the GPUgroup field of the given PGPU.

Signature:

```
GPU_group ref get_GPU_group (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given PGPU.

Signature:

```
host ref get_host (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getissystemdisplaydevice

Overview:

Get the issystemdisplaydevice field of the given PGPU.

Signature:

```
bool get_is_system_display_device (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PGPU.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPCI

Overview:

Get the PCI field of the given PGPU.

Signature:

```
PCI ref get_PCI (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PGPU.

Signature:

```
PGPU record get_record (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU record`

all fields from the object

RPC name: getremainingcapacity

Overview:

Signature:

```
int get_remaining_capacity (session ref session_ref, PGPU ref self,
VGPU_type ref vgpu_type)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to query
`VGPU_type ref`	`vgpu_type`	The VGPU type for which we want to find the number of VGPUs which can still be started on this PGPU

Minimum Role: read-only

Return Type: `int`

The number of VGPUs of the specified type which can still be started on this PGPU

RPC name: getresidentVGPUs

Overview:

Get the residentVGPUs field of the given PGPU.

Signature:

```
VGPU ref set get_resident_VGPUs (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsupportedVGPUmaxcapacities

Overview:

Get the supportedVGPUmaxcapacities field of the given PGPU.

Signature:

```
(VGPU_type ref -> int) map get_supported_VGPU_max_capacities (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(VGPU_type ref -> int) map`

value of the field

RPC name: getsupportedVGPUtypes

Overview:

Get the supportedVGPUtypes field of the given PGPU.

Signature:

```
VGPU_type ref set get_supported_VGPU_types (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PGPU.

Signature:

```
string get_uuid (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removeenabledVGPUtypes

Overview:

Signature:

```
void remove_enabled_VGPU_types (session ref session_ref, PGPU ref self,
VGPU_type ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PGPU ref`	`self`	The PGPU from which we are removing an enabled VGPU type
`VGPU_type ref`	`value`	The VGPU type to disable

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PGPU. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PGPU ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setenabledVGPUtypes

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_enabled_VGPU_types (session ref session_ref, PGPU ref self,
VGPU_type ref set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU on which we are enabling a set of VGPU types
`VGPU_type ref set`	`value`	The VGPU types to enable

Minimum Role: pool-operator

Return Type: `void`

RPC name: setGPUgroup

Overview:

Signature:

```
void set_GPU_group (session ref session_ref, PGPU ref self, GPU_group ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to move to a new group
`GPU_group ref`	`value`	The group to which the PGPU will be moved

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PGPU.

Signature:

```
void set_other_config (session ref session_ref, PGPU ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: PIF

A physical network interface note separate VLANs are represented as several PIFs

Fields for class: PIF

FIELD	TYPE	QUALIFIER	DESCRIPTION
`bond_master_of`	`Bond ref set`	*RO/runtime*	Indicates this PIF represents the results of a bond
`bond_slave_of`	`Bond ref`	*RO/runtime*	Indicates which bond this interface is part of

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`capabilities`	`string set`	*RO/runtime*	Additional capabilities on the interface.
`currently_attached`	`bool`	*RO/runtime*	true if this interface is online
`device`	`string`	*RO/constructor*	machine-readable name of the interface (for example, eth0)
`disallow_unplug`	`bool`	*RO/runtime*	Prevent this PIF from being unplugged; set this to notify the management tool-stack that the PIF has a special use and should not be unplugged under any circumstances (for example, because you're running storage traffic over it)
`DNS`	`string`	*RO/runtime*	Comma separated list of the IP addresses of the DNS servers to use
`gateway`	`string`	*RO/runtime*	IP gateway
`host`	`host ref`	*RO/constructor*	physical machine to which this pif is connected
`igmp_snooping_status`	`pif_igmp_status`	*RO/runtime*	The IGMP snooping status of the corresponding network bridge
`IP`	`string`	*RO/runtime*	IP address

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`ip_configuration_mode`	`ip_configuration_mode`	*RO/runtime*	Sets if and how this interface gets an IP address
`IPv6`	`string set`	*RO/runtime*	IPv6 address
`ipv6_configuration_mode`	`ipv6_configuration_mode`	*RO/runtime*	Sets if and how this interface gets an IPv6 address
`ipv6_gateway`	`string`	*RO/runtime*	IPv6 gateway
`MAC`	`string`	*RO/constructor*	ethernet MAC address of physical interface
`managed`	`bool`	*RO/constructor*	Indicates whether the interface is managed by xapi. If it is not, then xapi will not configure the interface, the commands PIF.plug/ unplug/ reconfigure_ip(v6) cannot be used, nor can the interface be bonded or have VLANs based on top through xapi.
`management`	`bool`	*RO/runtime*	Indicates whether the control software is listening for connections on this interface
`metrics`	`PIF_metrics ref`	*RO/runtime*	metrics associated with this PIF
`MTU`	`int`	*RO/constructor*	MTU in octets

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`netmask`	`string`	*RO/runtime*	IP netmask
`network`	`network ref`	*RO/constructor*	virtual network to which this pif is connected
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PCI`	`PCI ref`	*RO/runtime*	Link to underlying PCI device
`physical`	`bool`	*RO/runtime*	true if this represents a physical network interface
`primary_address_type`	`primary_address_type`	*RO/runtime*	Which protocol should define the primary address of this interface
`properties`	`(string -> string) map`	*RO/runtime*	Additional configuration properties for the interface.
`sriov_logical_PIF_of`	`network_sriov ref set`	*RO/runtime*	Indicates which network_sriov this interface is logical of
`sriov_physical_PIF_of`	`network_sriov ref set`	*RO/runtime*	Indicates which network_sriov this interface is physical of
`tunnel_access_PIF_of`	`tunnel ref set`	*RO/runtime*	Indicates to which tunnel this PIF gives access

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`tunnel_transport_PIF_of`	`tunnel ref set`	*RO/runtime*	Indicates to which tunnel this PIF provides transport
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VLAN`	`int`	*RO/constructor*	VLAN tag for all traffic passing through this interface
`VLAN_master_of`	`VLAN ref`	*RO/runtime*	Indicates which VLAN this interface receives untagged traffic from
`VLAN_slave_of`	`VLAN ref set`	*RO/runtime*	Indicates which VLANs this interface transmits tagged traffic to

RPCs associated with class: PIF

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PIF.

Signature:

```
void add_to_other_config (session ref session_ref, PIF ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: createVLAN

This message is deprecated.

Overview:

Create a VLAN interface from an existing physical interface. This call is deprecated: use VLAN.create instead

Signature:

```
PIF ref create_VLAN (session ref session_ref, string device, network ref
network, host ref host, int VLAN)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`device`	physical interface on which to create the VLAN interface
`network ref`	`network`	network to which this interface should be connected
`host ref`	`host`	physical machine to which this PIF is connected
`int`	`VLAN`	VLAN tag for the new interface

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `PIF ref`

The reference of the created PIF object

Possible Error Codes: `VLAN_TAG_INVALID`

RPC name: dbforget

Overview:

Destroy a PIF database record.

Signature:

```
void db_forget (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	The ref of the PIF whose database record should be destroyed

Minimum Role: pool-operator

Return Type: `void`

RPC name: dbintroduce

Overview:

Create a new PIF record in the database only

Signature:

```
PIF ref db_introduce (session ref session_ref, string device, network ref
network, host ref host, string MAC, int MTU, int VLAN, bool physical,
ip_configuration_mode ip_configuration_mode, string IP, string netmask,
string gateway, string DNS, Bond ref bond_slave_of, VLAN ref
VLAN_master_of, bool management, (string -> string) map other_config, bool
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
disallow_unplug, ipv6_configuration_mode ipv6_configuration_mode, string
set IPv6, string ipv6_gateway, primary_address_type primary_address_type,
bool managed, (string -> string) map properties)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`device`	
`network ref`	`network`	
`host ref`	`host`	
`string`	`MAC`	
`int`	`MTU`	
`int`	`VLAN`	
`bool`	`physical`	
`ip_configuration_mode`	`ip_configuration_mode`	
`string`	`IP`	
`string`	`netmask`	
`string`	`gateway`	
`string`	`DNS`	
`Bond ref`	`bond_slave_of`	
`VLAN ref`	`VLAN_master_of`	
`bool`	`management`	
`(string -> string) map`	`other_config`	
`bool`	`disallow_unplug`	

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`ipv6_configuration_mode`	`ipv6_configuration_mode`	
`string set`	`IPv6`	
`string`	`ipv6_gateway`	
`primary_address_type`	`primary_address_type`	
`bool`	`managed`	
`(string -> string) map`	`properties`	

Minimum Role: pool-operator

Return Type: `PIF ref`

The ref of the newly created PIF record.

RPC name: destroy

This message is deprecated.

Overview:

Destroy the PIF object provided it is a VLAN interface. This call is deprecated: use VLAN.destroy or Bond.destroy instead

Signature:

```
void destroy (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to destroy

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `PIF_IS_PHYSICAL`

RPC name: forget

Overview:

Destroy the PIF object matching a particular network interface

Signature:

```
void forget (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	The PIF object to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `PIF_TUNNEL_STILL_EXISTS`, `CLUSTERING_ENABLED`

RPC name: getall

Overview:

Return a list of all the PIFs known to the system.

Signature:

```
PIF ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `PIF ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PIF references to PIF records for all PIFs known to the system.

Signature:

```
(PIF ref -> PIF record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PIF ref -> PIF record) map`

records of all objects

RPC name: getbondmasterof

Overview:

Get the bondmasterof field of the given PIF.

Signature:

```
Bond ref set get_bond_master_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Bond ref set`

value of the field

RPC name: getbondslaveof

Overview:

Get the bondslaveof field of the given PIF.

Signature:

```
Bond ref get_bond_slave_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Bond ref`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the PIF instance with the specified UUID.

Signature:

```
PIF ref get_by_uuid (session ref session_ref, string uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PIF ref`

reference to the object

RPC name: getcapabilities

Overview:

Get the capabilities field of the given PIF.

Signature:

```
string set get_capabilities (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_currently_attached (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdevice

Overview:

Get the device field of the given PIF.

Signature:

```
string get_device (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdisallowunplug

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the disallowunplug field of the given PIF.

Signature:

```
bool get_disallow_unplug (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getDNS

Overview:

Get the DNS field of the given PIF.

Signature:

```
string get_DNS (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getgateway

Overview:

Get the gateway field of the given PIF.

Signature:

```
string get_gateway (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given PIF.

Signature:

```
host ref get_host (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `host ref`

value of the field

RPC name: getigmpsnoopingstatus

Overview:

Get the igmpsnoopingstatus field of the given PIF.

Signature:

```
pif_igmp_status get_igmp_snooping_status (session ref session_ref, PIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pif_igmp_status`

value of the field

RPC name: getIP

Overview:

Get the IP field of the given PIF.

Signature:

```
string get_IP (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getipconfigurationmode

Overview:

Get the ipconfigurationmode field of the given PIF.

Signature:

```
ip_configuration_mode get_ip_configuration_mode (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `ip_configuration_mode`

value of the field

RPC name: getIPv6

Overview:

Get the IPv6 field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set get_IPv6 (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv6configurationmode

Overview:

Get the ipv6configurationmode field of the given PIF.

Signature:

```
ipv6_configuration_mode get_ipv6_configuration_mode (session ref
session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `ipv6_configuration_mode`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getipv6gateway

Overview:

Get the ipv6gateway field of the given PIF.

Signature:

```
string get_ipv6_gateway (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getMAC

Overview:

Get the MAC field of the given PIF.

Signature:

```
string get_MAC (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getmanaged

Overview:

Get the managed field of the given PIF.

Signature:

```
bool get_managed (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmanagement

Overview:

Get the management field of the given PIF.

Signature:

```
bool get_management (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetrics

Overview:

Get the metrics field of the given PIF.

Signature:

```
PIF_metrics ref get_metrics (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF_metrics ref`

value of the field

RPC name: getMTU

Overview:

Get the MTU field of the given PIF.

Signature:

```
int get_MTU (session ref session_ref, PIF ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnetmask

Overview:

Get the netmask field of the given PIF.

Signature:

```
string get_netmask (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnetwork

Overview:

Get the network field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
network ref get_network (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PIF.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getPCI

Overview:

Get the PCI field of the given PIF.

Signature:

```
PCI ref get_PCI (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref`

value of the field

RPC name: getphysical

Overview:

Get the physical field of the given PIF.

Signature:

```
bool get_physical (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getprimaryaddresstype

Overview:

Get the primaryaddresstype field of the given PIF.

Signature:

```
primary_address_type get_primary_address_type (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `primary_address_type`

value of the field

RPC name: getproperties

Overview:

Get the properties field of the given PIF.

Signature:

```
(string -> string) map get_properties (session ref session_ref, PIF ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PIF.

Signature:

```
PIF record get_record (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF record`

all fields from the object

RPC name: getsriovlogicalPIFof

Overview:

Get the sriovlogicalPIFof field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
network_sriov ref set get_sriov_logical_PIF_of (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_sriov ref set`

value of the field

RPC name: getsriovphysicalPIFof

Overview:

Get the sriovphysicalPIFof field of the given PIF.

Signature:

```
network_sriov ref set get_sriov_physical_PIF_of (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_sriov ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettunnelaccessPIFof

Overview:

Get the tunnelaccessPIFof field of the given PIF.

Signature:

```
tunnel ref set get_tunnel_access_PIF_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tunnel ref set`

value of the field

RPC name: gettunneltransportPIFof

Overview:

Get the tunneltransportPIFof field of the given PIF.

Signature:

```
tunnel ref set get_tunnel_transport_PIF_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `tunnel ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PIF.

Signature:

```
string get_uuid (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVLAN

Overview:

Get the VLAN field of the given PIF.

Signature:

```
int get_VLAN (session ref session_ref, PIF ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVLANmasterof

Overview:

Get the VLANmasterof field of the given PIF.

Signature:

```
VLAN ref get_VLAN_master_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VLAN ref`

value of the field

RPC name: getVLANslaveof

Overview:

Get the VLANslaveof field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VLAN ref set get_VLAN_slave_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VLAN ref set`

value of the field

RPC name: introduce

Overview:

Create a PIF object matching a particular network interface

Signature:

```
PIF ref introduce (session ref session_ref, host ref host, string MAC,
string device, bool managed)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host on which the interface exists
`string`	`MAC`	The MAC address of the interface
`string`	`device`	The device name to use for the interface

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`managed`	Indicates whether the interface is managed by xapi (defaults to "true")

Minimum Role: pool-operator

Return Type: `PIF ref`

The reference of the created PIF object

RPC name: plug

Overview:

Attempt to bring up a physical interface

Signature:

```
void plug (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to plug

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `TRANSPORT_PIF_NOT_CONFIGURED`

RPC name: reconfigureip

Overview:

Reconfigure the IP address settings for this interface

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void reconfigure_ip (session ref session_ref, PIF ref self,
ip_configuration_mode mode, string IP, string netmask, string gateway,
string DNS)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to reconfigure
`ip_configuration_mode`	`mode`	whether to use dynamic/ static/no-assignment
`string`	`IP`	the new IP address
`string`	`netmask`	the new netmask
`string`	`gateway`	the new gateway
`string`	`DNS`	the new DNS settings

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTERING_ENABLED`

RPC name: reconfigureipv6

Overview:

Reconfigure the IPv6 address settings for this interface

Signature:

```
void reconfigure_ipv6 (session ref session_ref, PIF ref self,
ipv6_configuration_mode mode, string IPv6, string gateway, string DNS)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to reconfigure
`ipv6_configuration_mode`	`mode`	whether to use dynamic/ static/no-assignment
`string`	`IPv6`	the new IPv6 address (in <addr>/<prefix length> format)
`string`	`gateway`	the new gateway
`string`	`DNS`	the new DNS settings

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTERING_ENABLED`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PIF. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PIF ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: scan

Overview:

Scan for physical interfaces on a host and create PIF objects to represent them

Signature:

```
void scan (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host on which to scan

Minimum Role: pool-operator

Return Type: `void`

RPC name: setdisallowunplug

Overview:

Set whether unplugging the PIF is allowed

Signature:

```
void set_disallow_unplug (session ref session_ref, PIF ref self, bool
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	Reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OTHER_OPERATION_IN_PROGRESS`, `CLUSTERING_ENABLED`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PIF.

Signature:

```
void set_other_config (session ref session_ref, PIF ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setprimaryaddresstype

Overview:

Change the primary address type used by this PIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_primary_address_type (session ref session_ref, PIF ref self,
primary_address_type primary_address_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to reconfigure
`primary_address_type`	`primary_address_type`	Whether to prefer IPv4 or IPv6 connections

Minimum Role: pool-operator

Return Type: `void`

RPC name: setproperty

Overview:

Set the value of a property of the PIF

Signature:

```
void set_property (session ref session_ref, PIF ref self, string name,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	The PIF
`string`	`name`	The property name
`string`	`value`	The property value

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: unplug

Overview:

Attempt to bring down a physical interface

Signature:

```
void unplug (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to unplug

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN`, `VIF_IN_USE`, `PIF_DOES_NOT_ALLOW_UNPLUG`, `PIF_HAS_FCOE_SR_IN_USE`

Class: PIFmetrics

The metrics associated with a physical network interface

Fields for class: PIFmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`carrier`	`bool`	*RO/runtime*	Report if the PIF got a carrier or not
`device_id`	`string`	*RO/runtime*	Report device ID
`device_name`	`string`	*RO/runtime*	Report device name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`duplex`	`bool`	*RO/runtime*	Full duplex capability of the link (if available)
`io_read_kbs`	`float`	*RO/runtime*	**Removed.** Read bandwidth (KiB/s)
`io_write_kbs`	`float`	*RO/runtime*	**Removed.** Write bandwidth (KiB/s)
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated
`other_config`	`(string -> string) map`	*RW*	additional configuration
`pci_bus_path`	`string`	*RO/runtime*	PCI bus path of the pif (if available)
`speed`	`int`	*RO/runtime*	Speed of the link in Mbit/s (if available)
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_id`	`string`	*RO/runtime*	Report vendor ID
`vendor_name`	`string`	*RO/runtime*	Report vendor name

RPCs associated with class: PIFmetrics

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PIFmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_other_config (session ref session_ref, PIF_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PIFmetrics instances known to the system.

Signature:

```
PIF_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PIF_metrics ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of PIFmetrics references to PIFmetrics records for all PIFmetrics instances known to the system.

Signature:

```
(PIF_metrics ref -> PIF_metrics record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PIF_metrics ref -> PIF_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PIFmetrics instance with the specified UUID.

Signature:

```
PIF_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `PIF_metrics ref`

reference to the object

RPC name: getcarrier

Overview:

Get the carrier field of the given PIFmetrics.

Signature:

```
bool get_carrier (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdeviceid

Overview:

Get the deviceid field of the given PIFmetrics.

Signature:

```
string get_device_id (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdevicename

Overview:

Get the devicename field of the given PIFmetrics.

Signature:

```
string get_device_name (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getduplex

Overview:

Get the duplex field of the given PIFmetrics.

Signature:

```
bool get_duplex (session ref session_ref, PIF_metrics ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getioreadkbs

This message is removed.

Overview:

Get the io/readkbs field of the given PIFmetrics.

Signature:

```
float get_io_read_kbs (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getiowritekbs

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the io/writekbs field of the given PIFmetrics.

Signature:

```
float get_io_write_kbs (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given PIFmetrics.

Signature:

```
datetime get_last_updated (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PIFmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpcibuspath

Overview:

Get the pcibuspath field of the given PIFmetrics.

Signature:

```
string get_pci_bus_path (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PIFmetrics.

Signature:

```
PIF_metrics record get_record (session ref session_ref, PIF_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF_metrics record`

all fields from the object

RPC name: getspeed

Overview:

Get the speed field of the given PIFmetrics.

Signature:

```
int get_speed (session ref session_ref, PIF_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PIFmetrics.

Signature:

```
string get_uuid (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorid

Overview:

Get the vendorid field of the given PIFmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_vendor_id (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorname

Overview:

Get the vendorname field of the given PIFmetrics.

Signature:

```
string get_vendor_name (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given PIFmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PIF_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PIFmetrics.

Signature:

```
void set_other_config (session ref session_ref, PIF_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

Class: pool

Poolwide information

Fields for class: pool

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`pool_allowed_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`auto_update_vm_secureboot_certificates`	`bool`	*RW*	When true, at VM.create time the pool automatically marks a newly created VM whose Secure Boot certificates are due to expire by setting VM.secureboot_certificates_state to update_on_boot (the same effect as VM.update_secureboot_certificates_on_boot), so the certificates are updated on the VM's next boot. This applies to VMs created by VM.create, including those created by operations that

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			internally call it: the HTTP(s) PUT /import and / import_metadata calls and disaster recovery (VM.recover and VM_appliance.recover); it does not apply to VM.clone or VM.copy, which inherit the source VM's state.
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this pool
`client_certificate_auth_enabled`	`bool`	*RO/runtime*	True if authentication by TLS client certificates is enabled
`client_certificate_auth_name`	`string`	*RO/runtime*	The name (CN/SAN) that an incoming client certificate must have to allow authentication
`coordinator_bias`	`bool`	*RW*	true if bias against pool master when scheduling vms is enabled, false otherwise
`cpu_info`	`(string -> string) map`	*RO/runtime*	Details about the physical CPUs on the pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`crash_dump_SR`	`SR ref`	*RW*	The SR in which VDIs for crash dumps are created
`current_operations`	`(string -> pool_allowed_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`custom_uefi_certificates`	`string`	*RO/constructor*	Custom UEFI certificates allowing Secure Boot
`default_SR`	`SR ref`	*RW*	Default SR for VDIs
`ext_auth_cache_enabled`	`bool`	*RO/runtime*	Specifies whether external authentication caching is enabled for this pool or not
`ext_auth_cache_expiry`	`int`	*RO/runtime*	Specifies how long external authentication entries should be cached for (seconds)
`ext_auth_cache_size`	`int`	*RO/runtime*	Maximum capacity of external authentication cache
`ext_auth_max_threads`	`int`	*RO/constructor*	Maximum number of threads to use for external (AD) authentication

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`guest_agent_config`	`(string -> string) map`	*RO/runtime*	Pool-wide guest agent configuration information
`gui_config`	`(string -> string) map`	*RW*	gui-specific configuration for pool
`ha_allow_overcommit`	`bool`	*RW*	If set to false then operations which would cause the Pool to become overcommitted will be blocked.
`ha_cluster_stack`	`string`	*RO/runtime*	The HA cluster stack that is currently in use. Only valid when HA is enabled.
`ha_configuration`	`(string -> string) map`	*RO/runtime*	The current HA configuration
`ha_enabled`	`bool`	*RO/runtime*	true if HA is enabled on the pool, false otherwise
`ha_host_failures_to_tolerate`	`int`	*RO/runtime*	Number of host failures to tolerate before the Pool is declared to be overcommitted
`ha_overcommitted`	`bool`	*RO/runtime*	True if the Pool is considered to be overcommitted i.e. if there exist insufficient physical resources to tolerate the configured number of host failures

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`ha_plan_exists_for`	`int`	*RO/runtime*	Number of future host failures we have managed to find a plan for. Once this reaches zero any future host failures will cause the failure of protected VMs.
`ha_reboot_vm_on_internal_shutdown`	`bool`	*RW*	Indicates whether an HA-protected VM that is shut down from inside (not through the API) should be automatically rebooted when HA is enabled
`ha_statefiles`	`string set`	*RO/runtime*	HA statefile VDIs in use
`health_check_config`	`(string -> string) map`	*RW*	Configuration for the automatic health check feature
`igmp_snooping_enabled`	`bool`	*RO/runtime*	true if IGMP snooping is enabled in the pool, false otherwise.
`is_psr_pending`	`bool`	*RW*	True if either a PSR is running or we are waiting for a PSR to be re-run
`last_update_sync`	`datetime`	*RO/runtime*	time of the last update sychronization

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`license_server`	`(string -> string) map`	*RW*	Licensing data shared within the whole pool
`limit_console_sessions`	`bool`	*RW*	When true, only one console connection per VM/host in the pool is accepted. Otherwise every connection for a VM/host's console is accepted. Note: when true, connection attempts via websocket will be rejected.
`live_patching_disabled`	`bool`	*RW*	The pool-wide flag to show if the live patching feauture is disabled or not.
`local_auth_max_threads`	`int`	*RO/constructor*	Maximum number of threads to use for PAM authentication
`master`	`host ref`	*RO/runtime*	The host that is pool master
`metadata_VDIs`	`VDI ref set`	*RO/runtime*	The set of currently known metadata VDIs for this pool
`migration_compression`	`bool`	*RW*	Default behaviour during migration, True if stream compression should be used
`name_description`	`string`	*RW*	Description

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_label`	`string`	*RW*	Short name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`policy_no_vendor_device`	`bool`	*RW*	**Deprecated.** This field was consulted when VM.create did not specify a value for 'has_vendor_device'; VM.create now uses a simple default and no longer consults this value.
`recommendations`	`(string -> string) map`	*RO/runtime*	The recommended pool properties for clients to respect for optimal performance. For example, max-vm-group=5
`redo_log_enabled`	`bool`	*RO/runtime*	true a redo-log is to be used other than when HA is enabled, false otherwise
`redo_log_vdi`	`VDI ref`	*RO/runtime*	indicates the VDI to use for the redo-log other than when HA is enabled
`repositories`	`Repository ref set`	*RO/runtime*	The set of currently enabled repositories
`repository_proxy_password`	`secret ref`	*RO/runtime*	Password for the authentication of the proxy used in syncing with the enabled repositories

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`repository_proxy_url`	`string`	*RO/runtime*	Url of the proxy used in syncing with the enabled repositories
`repository_proxy_username`	`string`	*RO/runtime*	Username for the authentication of the proxy used in syncing with the enabled repositories
`restrictions`	`(string -> string) map`	*RO/runtime*	Pool-wide restrictions currently in effect
`suspend_image_SR`	`SR ref`	*RW*	The SR in which VDIs for suspend images are created
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`telemetry_frequency`	`telemetry_frequency`	*RO/runtime*	How often the telemetry collection will be carried out
`telemetry_next_collection`	`datetime`	*RO/runtime*	The earliest timestamp (in UTC) when the next round of telemetry collection can be carried out
`telemetry_uuid`	`secret ref`	*RO/runtime*	The UUID of the pool for identification of telemetry data
`tls_verification_enabled`	`bool`	*RO/runtime*	True iff TLS certificate verification is enabled

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uefi_certificates`	`string`	*RO/constructor*	The UEFI certificates allowing Secure Boot
`update_sync_day`	`int`	*RO/runtime*	The day of the week the update synchronizations will be scheduled, based on pool's local timezone. Ignored when update_sync_frequency is daily
`update_sync_enabled`	`bool`	*RO/runtime*	Whether periodic update synchronization is enabled or not
`update_sync_frequency`	`update_sync_frequency`	*RO/runtime*	The frequency at which updates are synchronized from a remote CDN: daily or weekly.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vm_console_idle_timeout`	`int`	*RW*	The maximum time (in seconds) that a VM's console can be idle before it is automatically disconnected. The default value 0 means never timeout. This setting applies only to VM consoles; for host consoles, use the separate parameter

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			'host.console_idle_timeout'.
`vswitch_controller`	`string`	*RO/runtime*	**Deprecated.** address of the vswitch controller
`wlb_enabled`	`bool`	*RW*	true if workload balancing is enabled on the pool, false otherwise
`wlb_url`	`string`	*RO/runtime*	Url for the configured workload balancing host
`wlb_username`	`string`	*RO/runtime*	Username for accessing the workload balancing host
`wlb_verify_cert`	`bool`	*RW*	**Deprecated.** true if communication with the WLB server should enforce TLS certificate verification.

RPCs associated with class: pool

RPC name: addrepository

Overview:

Add a repository to the enabled set

Signature:

```
void add_repository (session ref session_ref, pool ref self, Repository ref value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Repository ref`	`value`	The repository to be added to the enabled set

Minimum Role: client-cert

Return Type: `void`

RPC name: addtags

Overview:

Add the given value to the tags field of the given pool. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtoguestagentconfig

Overview:

Add a keyvalue pair to the poolwide guest agent configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void add_to_guest_agent_config (session ref session_ref, pool ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`key`	The key to add
`string`	`value`	The value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: addtoguiconfig

Overview:

Add the given keyvalue pair to the guiconfig field of the given pool.

Signature:

```
void add_to_gui_config (session ref session_ref, pool ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtohealthcheckconfig

Overview:

Add the given keyvalue pair to the healthcheckconfig field of the given pool.

Signature:

```
void add_to_health_check_config (session ref session_ref, pool ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtolicenseserver

Overview:

Add the given keyvalue pair to the licenseserver field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_license_server (session ref session_ref, pool ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given pool.

Signature:

```
void add_to_other_config (session ref session_ref, pool ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: applyedition

Overview:

Apply an edition to all hosts in the pool

Signature:

```
void apply_edition (session ref session_ref, pool ref self, string
edition)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool
`string`	`edition`	The requested edition

Minimum Role: pool-operator

Return Type: `void`

RPC name: certificateinstall

This message is deprecated.

Overview:

Install a TLS CA certificate, poolwide.

Signature:

```
void certificate_install (session ref session_ref, string name, string
cert)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	A name to give the certificate
`string`	`cert`	The certificate in PEM format

Minimum Role: pool-operator

Return Type: `void`

RPC name: certificatelist

This message is deprecated.

Overview:

List the names of all installed TLS CA certificates.

Signature:

```
string set certificate_list (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `string set`

All installed certificates

RPC name: certificatesync

Overview:

Copy the TLS CA certificates and CRLs of the master to all slaves.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void certificate_sync (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: certificateuninstall

This message is deprecated.

Overview:

Remove a poolwide TLS CA certificate.

Signature:

```
void certificate_uninstall (session ref session_ref, string name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	The certificate name

Minimum Role: pool-operator

Return Type: `void`

RPC name: checkupdatereadiness

Overview:

Check if the pool is ready to be updated. If not, report the reasons.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set set check_update_readiness (session ref session_ref, pool ref
self, bool requires_reboot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`requires_reboot`	Assume that the update will require host reboots

Minimum Role: client-cert

Return Type: `string set set`

A set of error codes with arguments, if the pool is not ready to update. An empty list means the pool can be updated.

RPC name: configurerepositoryproxy

Overview:

Configure proxy for RPM package repositories.

Signature:

```
void configure_repository_proxy (session ref session_ref, pool ref self,
string url, string username, string password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`url`	The URL of the proxy server

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`username`	The username used to authenticate with the proxy server
`string`	`password`	The password used to authenticate with the proxy server

Minimum Role: client-cert

Return Type: `void`

RPC name: configureupdatesync

Overview:

Configure periodic update synchronization to synchronize updates from a remote CDN

Signature:

```
void configure_update_sync (session ref session_ref, pool ref self,
update_sync_frequency update_sync_frequency, int update_sync_day)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`update_sync_frequency`	`update_sync_frequency`	The frequency at which updates are synchronized from a remote CDN: daily or weekly.
`int`	`update_sync_day`	The day of the week the update synchronization will happen, based on pool's local timezone. Valid values are 0 to 6, 0 being Sunday. For

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
		'daily' schedule, the value is ignored.

Minimum Role: pool-operator

Return Type: `void`

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this pool

Signature:

```
blob ref create_new_blob (session ref session_ref, pool ref pool, string name, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: createVLAN

Overview:

Create PIFs, mapping a network to the same physical interface/VLAN on each host. This call is deprecated: use Pool.createVLANfromPIF instead.

Signature:

```
PIF ref set create_VLAN (session ref session_ref, string device, network ref network, int VLAN)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`device`	physical interface on which to create the VLAN interface
`network ref`	`network`	network to which this interface should be connected
`int`	`VLAN`	VLAN tag for the new interface

Minimum Role: pool-operator

Return Type: `PIF ref set`

The references of the created PIF objects

Possible Error Codes: `VLAN_TAG_INVALID`

RPC name: createVLANfromPIF

Overview:

Create a poolwide VLAN by taking the PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PIF ref set create_VLAN_from_PIF (session ref session_ref, PIF ref pif,
network ref network, int VLAN)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`pif`	physical interface on any particular host, that identifies the PIF on which to create the (pool-wide) VLAN interface
`network ref`	`network`	network to which this interface should be connected
`int`	`VLAN`	VLAN tag for the new interface

Minimum Role: pool-operator

Return Type: `PIF ref set`

The references of the created PIF objects

Possible Error Codes: `VLAN_TAG_INVALID`

RPC name: crlinstall

Overview:

Install a TLS CAissued Certificate Revocation List, poolwide.

Signature:

```
void crl_install (session ref session_ref, string name, string cert)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	A name to give the CRL
`string`	`cert`	The CRL

Minimum Role: pool-operator

Return Type: `void`

RPC name: crllist

Overview:

List the names of all installed TLS CAissued Certificate Revocation Lists.

Signature:

```
string set crl_list (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `string set`

The names of all installed CRLs

RPC name: crluninstall

Overview:

Remove a poolwide TLS CAissued Certificate Revocation List.

Signature:

```
void crl_uninstall (session ref session_ref, string name)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	The CRL name

Minimum Role: pool-operator

Return Type: `void`

RPC name: deconfigurewlb

Overview:

Permanently deconfigures workload balancing monitoring on this pool

Signature:

```
void deconfigure_wlb (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: designatenewmaster

Overview:

Perform an orderly handover of the role of master to the referenced host.

Signature:

```
void designate_new_master (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host who should become the new master

Minimum Role: pool-operator

Return Type: `void`

RPC name: detectnonhomogeneousexternalauth

Overview:

This call asynchronously detects if the external authentication configuration in any slave is different from that in the master and raises appropriate alerts

Signature:

```
void detect_nonhomogeneous_external_auth (session ref session_ref, pool ref pool)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool where to detect non-homogeneous external authentication configuration

Minimum Role: pool-operator

Return Type: `void`

RPC name: disableclientcertificateauth

Overview:

Disable client certificate authentication on the pool

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void disable_client_certificate_auth (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: client-cert

Return Type: `void`

RPC name: disableexternalauth

Overview:

This call disables external authentication on all the hosts of the pool

Signature:

```
void disable_external_auth (session ref session_ref, pool ref pool,
(string -> string) map config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool whose external authentication should be disabled
`(string -> string) map`	`config`	Optional parameters as a list of key-values containing the configuration data

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: disableha

Overview:

Turn off High Availability mode

Signature:

```
void disable_ha (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: client-cert

Return Type: `void`

RPC name: disablelocalstoragecaching

Overview:

This call disables poolwide local storage caching

Signature:

```
void disable_local_storage_caching (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: disableredolog

Overview:

Disable the redo log if in use, unless HA is enabled.

Signature:

```
void disable_redo_log (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: disablerepositoryproxy

Overview:

Disable the proxy for RPM package repositories.

Signature:

```
void disable_repository_proxy (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: client-cert

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: disablessh

Overview:

Disable SSH access on all hosts in the pool. It's a helper which calls host.disablessh for all the hosts in the pool.

Signature:

```
void disable_ssh (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: pool-admin

Return Type: `void`

RPC name: disablessllegacy

This message is deprecated.

Overview:

Sets ssllegacy false on each host, poolmaster last. See Host.ssllegacy and Host.setssllegacy.

Signature:

```
void disable_ssl_legacy (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	(ignored)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: eject

Overview:

Instruct a pool master to eject a host from the pool

Signature:

```
void eject (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to eject

Minimum Role: pool-operator

Return Type: `void`

RPC name: emergencyresetmaster

Overview:

Instruct a slave already in a pool that the master has changed

Signature:

```
void emergency_reset_master (session ref session_ref, string
master_address)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`master_address`	The hostname of the master

Minimum Role: pool-operator

Return Type: `void`

RPC name: emergencytransitiontomaster

Overview:

Instruct host that's currently a slave to transition to being master

Signature:

```
void emergency_transition_to_master (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: enableclientcertificateauth

Overview:

Enable client certificate authentication on the pool

Signature:

```
void enable_client_certificate_auth (session ref session_ref, pool ref
self, string name)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`name`	The name (CN/SAN) that an incoming client certificate must have to allow authentication

Minimum Role: client-cert

Return Type: `void`

RPC name: enableexternalauth

Overview:

This call enables external authentication on all the hosts of the pool

Signature:

```
void enable_external_auth (session ref session_ref, pool ref pool, (string -> string) map config, string service_name, string auth_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool whose external authentication should be enabled
`(string -> string) map`	`config`	A list of key-values containing the configuration data
`string`	`service_name`	The name of the service

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`auth_type`	The type of authentication (for example, AD for Active Directory)

Minimum Role: pool-admin

Return Type: `void`

RPC name: enableha

Overview:

Turn on High Availability mode

Signature:

```
void enable_ha (session ref session_ref, SR ref set heartbeat_srs, (string
-> string) map configuration)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref set`	`heartbeat_srs`	Set of SRs to use for storage heartbeating
`(string -> string) map`	`configuration`	Detailed HA configuration to apply

Minimum Role: client-cert

Return Type: `void`

RPC name: enablelocalstoragecaching

Overview:

This call attempts to enable poolwide local storage caching

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void enable_local_storage_caching (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool

Minimum Role: pool-operator

Return Type: `void`

RPC name: enableredolog

Overview:

Enable the redo log on the given SR and start using it, unless HA is enabled.

Signature:

```
void enable_redo_log (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	SR to hold the redo log.

Minimum Role: pool-operator

Return Type: `void`

RPC name: enablessh

Overview:

Enable SSH access on all hosts in the pool. It's a helper which calls host.enablessh for all the hosts in the pool.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void enable_ssh (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: pool-admin

Return Type: `void`

RPC name: enablessllegacy

This message is removed.

Overview:

Sets ssllegacy true on each host, poolmaster last. See Host.ssllegacy and Host.setssllegacy.

Signature:

```
void enable_ssl_legacy (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	(ignored)

Minimum Role: pool-operator

Return Type: `void`

RPC name: enabletlsverification

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable TLS server certificate verification

Signature:

```
void enable_tls_verification (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-admin

Return Type: `void`

RPC name: externalauthsetldaps

Overview:

This call enables or disables LDAPS for external authentication on all hosts in the pool

Signature:

```
void external_auth_set_ldaps (session ref session_ref, pool ref pool, bool
ldaps, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool whose LDAPS configuration should be set
`bool`	`ldaps`	Whether to enable or disable LDAPS
`bool`	`force`	Force the operation even if already in the desired state

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the pools known to the system.

Signature:

```
pool ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `pool ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of pool references to pool records for all pools known to the system.

Signature:

```
(pool ref -> pool record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(pool ref -> pool record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given pool.

Signature:

```
pool_allowed_operations set get_allowed_operations (session ref
session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_allowed_operations set`

value of the field

RPC name: getautoupdatevmsecurebootcertificates

Overview:

Get the autoupdatevmsecurebootcertificates field of the given pool.

Signature:

```
bool get_auto_update_vm_secureboot_certificates (session ref session_ref,
pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given pool.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the pool instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
pool ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `pool ref`

reference to the object

RPC name: getclientcertificateauthenabled

Overview:

Get the clientcertificateauthenabled field of the given pool.

Signature:

```
bool get_client_certificate_auth_enabled (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getclientcertificateauthname

Overview:

Get the clientcertificateauthname field of the given pool.

Signature:

```
string get_client_certificate_auth_name (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getcoordinatorbias

Overview:

Get the coordinatorbias field of the given pool.

Signature:

```
bool get_coordinator_bias (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getcpuinfo

Overview:

Get the cpuinfo field of the given pool.

Signature:

```
(string -> string) map get_cpu_info (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcrashdumpSR

Overview:

Get the crashdumpSR field of the given pool.

Signature:

```
SR ref get_crash_dump_SR (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given pool.

Signature:

```
(string -> pool_allowed_operations) map get_current_operations (session
ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> pool_allowed_operations) map`

value of the field

RPC name: getcustomueficertificates

Overview:

Get the customueficertificates field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_custom_uefi_certificates (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdefaultSR

Overview:

Get the defaultSR field of the given pool.

Signature:

```
SR ref get_default_SR (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getextauthcacheenabled

Overview:

Get the extauthcacheenabled field of the given pool.

Signature:

```
bool get_ext_auth_cache_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getextauthcacheexpiry

Overview:

Get the extauthcacheexpiry field of the given pool.

Signature:

```
int get_ext_auth_cache_expiry (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getextauthcachesize

Overview:

Get the extauthcachesize field of the given pool.

Signature:

```
int get_ext_auth_cache_size (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getextauthmaxthreads

Overview:

Get the extauthmaxthreads field of the given pool.

Signature:

```
int get_ext_auth_max_threads (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getguestagentconfig

Overview:

Get the guestagentconfig field of the given pool.

Signature:

```
(string -> string) map get_guest_agent_config (session ref session_ref,
pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getguestsecurebootreadiness

Overview:

Signature:

```
pool_guest_secureboot_readiness get_guest_secureboot_readiness (session
ref session_ref, pool ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: pool-operator

Return Type: `pool_guest_secureboot_readiness`

The readiness of the pool

RPC name: getguiconfig

Overview:

Get the guiconfig field of the given pool.

Signature:

```
(string -> string) map get_gui_config (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethaallowovercommit

Overview:

Get the haallowovercommit field of the given pool.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
bool get_ha_allow_overcommit (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethaclusterstack

Overview:

Get the haclusterstack field of the given pool.

Signature:

```
string get_ha_cluster_stack (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethaconfiguration

Overview:

Get the haconfiguration field of the given pool.

Signature:

```
(string -> string) map get_ha_configuration (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethaenabled

Overview:

Get the haenabled field of the given pool.

Signature:

```
bool get_ha_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethahostfailurestotolerate

Overview:

Get the hahostfailurestotolerate field of the given pool.

Signature:

```
int get_ha_host_failures_to_tolerate (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gethaovercommitted

Overview:

Get the haovercommitted field of the given pool.

Signature:

```
bool get_ha_overcommitted (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethaplanexistsfor

Overview:

Get the haplanexistsfor field of the given pool.

Signature:

```
int get_ha_plan_exists_for (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getharebootvmoninternalshutdown

Overview:

Get the harebootvmoninternalshutdown field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_ha_reboot_vm_on_internal_shutdown (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethastatefiles

Overview:

Get the hastatefiles field of the given pool.

Signature:

```
string set get_ha_statefiles (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethealthcheckconfig

Overview:

Get the healthcheckconfig field of the given pool.

Signature:

```
(string -> string) map get_health_check_config (session ref session_ref,
pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getigmpsnoopingenabled

Overview:

Get the igmpsnoopingenabled field of the given pool.

Signature:

```
bool get_igmp_snooping_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getispsrpending

Overview:

Get the ispsrpending field of the given pool.

Signature:

```
bool get_is_psr_pending (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlastupdatesync

Overview:

Get the lastupdatesync field of the given pool.

Signature:

```
datetime get_last_update_sync (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlicenseserver

Overview:

Get the licenseserver field of the given pool.

Signature:

```
(string -> string) map get_license_server (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlicensestate

Overview:

This call returns the license state for the pool

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_license_state (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool

Minimum Role: read-only

Return Type: `(string -> string) map`

The pool's license state

RPC name: getlimitconsolesessions

Overview:

Get the limitconsolesessions field of the given pool.

Signature:

```
bool get_limit_console_sessions (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getlivepatchingdisabled

Overview:

Get the livepatchingdisabled field of the given pool.

Signature:

```
bool get_live_patching_disabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlocalauthmaxthreads

Overview:

Get the localauthmaxthreads field of the given pool.

Signature:

```
int get_local_auth_max_threads (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getmaster

Overview:

Get the master field of the given pool.

Signature:

```
host ref get_master (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getmetadataVDIs

Overview:

Get the metadataVDIs field of the given pool.

Signature:

```
VDI ref set get_metadata_VDIs (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref set`

value of the field

RPC name: getmigrationcompression

Overview:

Get the migrationcompression field of the given pool.

Signature:

```
bool get_migration_compression (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the namedescription field of the given pool.

Signature:

```
string get_name_description (session ref session_ref, pool ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the namelabel field of the given pool.

Signature:

```
string get_name_label (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_other_config (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpolicynovendordevice

This message is deprecated.

Overview:

Get the policynovendordevice field of the given pool.

Signature:

```
bool get_policy_no_vendor_device (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecommendations

Overview:

Get the recommendations field of the given pool.

Signature:

```
(string -> string) map get_recommendations (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given pool.

Signature:

```
pool record get_record (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `pool record`

all fields from the object

RPC name: getredologenabled

Overview:

Get the redologenabled field of the given pool.

Signature:

```
bool get_redo_log_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getredologvdi

Overview:

Get the redologvdi field of the given pool.

Signature:

```
VDI ref get_redo_log_vdi (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getrepositories

Overview:

Get the repositories field of the given pool.

Signature:

```
Repository ref set get_repositories (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Repository ref set`

value of the field

RPC name: getrepositoryproxypassword

Overview:

Get the repositoryproxypassword field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
secret ref get_repository_proxy_password (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `secret ref`

value of the field

RPC name: getrepositoryproxyurl

Overview:

Get the repositoryproxyurl field of the given pool.

Signature:

```
string get_repository_proxy_url (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrepositoryproxyusername

Overview:

Get the repositoryproxyusername field of the given pool.

Signature:

```
string get_repository_proxy_username (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrestrictions

Overview:

Get the restrictions field of the given pool.

Signature:

```
(string -> string) map get_restrictions (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsuspendimageSR

Overview:

Get the suspendimageSR field of the given pool.

Signature:

```
SR ref get_suspend_image_SR (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: gettags

Overview:

Get the tags field of the given pool.

Signature:

```
string set get_tags (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettelemetryfrequency

Overview:

Get the telemetryfrequency field of the given pool.

Signature:

```
telemetry_frequency get_telemetry_frequency (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `telemetry_frequency`

value of the field

RPC name: gettelemetrynextcollection

Overview:

Get the telemetrynextcollection field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
datetime get_telemetry_next_collection (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: gettelemetryuuid

Overview:

Get the telemetryuuid field of the given pool.

Signature:

```
secret ref get_telemetry_uuid (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `secret ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettlsverificationenabled

Overview:

Get the tlsverificationenabled field of the given pool.

Signature:

```
bool get_tls_verification_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getueficertificates

Overview:

Get the ueficertificates field of the given pool.

Signature:

```
string get_uefi_certificates (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getupdatesyncday

Overview:

Get the updatesyncday field of the given pool.

Signature:

```
int get_update_sync_day (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getupdatesyncenabled

Overview:

Get the updatesyncenabled field of the given pool.

Signature:

```
bool get_update_sync_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getupdatesyncfrequency

Overview:

Get the updatesyncfrequency field of the given pool.

Signature:

```
update_sync_frequency get_update_sync_frequency (session ref session_ref,
pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_sync_frequency`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvmconsoleidletimeout

Overview:

Get the vmconsoleidletimeout field of the given pool.

Signature:

```
int get_vm_console_idle_timeout (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getvswitchcontroller

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Get the vswitchcontroller field of the given pool.

Signature:

```
string get_vswitch_controller (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getwlbenabled

Overview:

Get the wlbenabled field of the given pool.

Signature:

```
bool get_wlb_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getwlburl

Overview:

Get the wlburl field of the given pool.

Signature:

```
string get_wlb_url (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getwlbusername

Overview:

Get the wlbusername field of the given pool.

Signature:

```
string get_wlb_username (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getwlbverifycert

This message is deprecated.

Overview:

Get the wlbverifycert field of the given pool.

Signature:

```
bool get_wlb_verify_cert (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: hacomputehypotheticalmaxhostfailurestotolerate

Overview:

Returns the maximum number of host failures we could tolerate before we would be unable to restart the provided VMs

Signature:

```
int ha_compute_hypothetical_max_host_failures_to_tolerate (session ref
session_ref, (VM ref -> string) map configuration)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`(VM ref -> string) map`	`configuration`	Map of protected VM reference to restart priority

Minimum Role: read-only

Return Type: `int`

maximum value for hahostfailurestotolerate given provided configuration

RPC name: hacomputemaxhostfailurestotolerate

Overview:

Returns the maximum number of host failures we could tolerate before we would be unable to restart configured VMs

Signature:

```
int ha_compute_max_host_failures_to_tolerate (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `int`

maximum value for hahostfailurestotolerate given current configuration

RPC name: hacomputevmfailoverplan

Overview:

Return a VM failover plan assuming a given subset of hosts fail

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(VM ref -> (string -> string) map) map ha_compute_vm_failover_plan
(session ref session_ref, host ref set failed_hosts, VM ref set
failed_vms)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref set`	`failed_hosts`	The set of hosts to assume have failed
`VM ref set`	`failed_vms`	The set of VMs to restart

Minimum Role: pool-operator

Return Type: `(VM ref -> (string -> string) map) map`

VM failover plan: a map of VM to host to restart the host on

RPC name: hafailoverplanexists

Overview:

Returns true if a VM failover plan exists for up to 'n' host failures

Signature:

```
bool ha_failover_plan_exists (session ref session_ref, int n)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`int`	`n`	The number of host failures to plan for

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `bool`

true if a failover plan exists for the supplied number of host failures

RPC name: hapreventrestartsfor

Overview:

When this call returns the VM restart logic will not run for the requested number of seconds. If the argument is zero then the restart thread is immediately unblocked

Signature:

```
void ha_prevent_restarts_for (session ref session_ref, int seconds)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`int`	`seconds`	The number of seconds to block the restart thread for

Minimum Role: pool-operator

Return Type: `void`

RPC name: hasextension

Overview:

Return true if the extension is available on the pool

Signature:

```
bool has_extension (session ref session_ref, pool ref self, string name)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`name`	The name of the API call

Minimum Role: pool-admin

Return Type: `bool`

True if the extension exists, false otherwise

RPC name: initializewlb

Overview:

Initializes workload balancing monitoring on this pool with the specified wlb server

Signature:

```
void initialize_wlb (session ref session_ref, string wlb_url, string wlb_username, string wlb_password, string xenserver_username, string xenserver_password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`wlb_url`	The ip address and port to use when accessing the wlb server
`string`	`wlb_username`	The username used to authenticate with the wlb server
`string`	`wlb_password`	The password used to authenticate with the wlb server

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`xenserver_username`	The username used by the wlb server to authenticate with the xenserver
`string`	`xenserver_password`	The password used by the wlb server to authenticate with the xenserver

Minimum Role: pool-operator

Return Type: `void`

RPC name: installcacertificate

This message is deprecated.

Overview:

Install a TLS CA certificate, poolwide.

Signature:

```
void install_ca_certificate (session ref session_ref, string name, string
cert)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	A name to give the certificate
`string`	`cert`	The certificate in PEM format

Minimum Role: client-cert

Return Type: `void`

RPC name: installtrustedcertificate

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Install a trusted TLS certificate, poolwide.

Signature:

```
void install_trusted_certificate (session ref session_ref, pool ref self,
bool ca, string cert, certificate_purpose set purpose)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`ca`	The trusted certificate is a root CA certificate used to verify a chain (true), or a leaf certificate used for certificate pinning (false)
`string`	`cert`	The certificate in PEM format
`certificate_purpose set`	`purpose`	The purpose of the certificate

Minimum Role: client-cert

Return Type: `void`

RPC name: join

Overview:

Instruct host to join a new pool

Signature:

```
void join (session ref session_ref, string master_address, string
master_username, string master_password)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`master_address`	The hostname of the master of the pool to join
`string`	`master_username`	The username of the master (for initial authentication)
`string`	`master_password`	The password for the master (for initial authentication)

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `JOINING_HOST_CANNOT_CONTAIN_SHARED_SRS`

RPC name: joinforce

Overview:

Instruct host to join a new pool

Signature:

```
void join_force (session ref session_ref, string master_address, string
master_username, string master_password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`master_address`	The hostname of the master of the pool to join
`string`	`master_username`	The username of the master (for initial authentication)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`master_password`	The password for the master (for initial authentication)

Minimum Role: pool-operator

Return Type: `void`

RPC name: managementreconfigure

Overview:

Reconfigure the management network interface for all Hosts in the Pool

Signature:

```
void management_reconfigure (session ref session_ref, network ref network)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	The network

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `HA_IS_ENABLED`, `PIF_NOT_PRESENT`, `CANNOT_PLUG_BOND_SLAVE`, `PIF_INCOMPATIBLE_PRIMARY_ADDRESS_TYPE`, `PIF_HAS_NO_NETWORK_CONFIGURATION`, `PIF_HAS_NO_V6_NETWORK_CONFIGURATION`

RPC name: recoverslaves

Overview:

Instruct a pool master, M, to try and contact its slaves and, if slaves are in emergency mode, reset their master address to M.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref set recover_slaves (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `host ref set`

list of hosts whose master address were successfully reset

RPC name: removefromguestagentconfig

Overview:

Remove a keyvalue pair from the poolwide guest agent configuration

Signature:

```
void remove_from_guest_agent_config (session ref session_ref, pool ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`key`	The key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: removefromguiconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the guiconfig field of the given pool. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_gui_config (session ref session_ref, pool ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: removefromhealthcheckconfig

Overview:

Remove the given key and its corresponding value from the healthcheckconfig field of the given pool. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_health_check_config (session ref session_ref, pool ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromlicenseserver

Overview:

Remove the given key and its corresponding value from the licenseserver field of the given pool. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_license_server (session ref session_ref, pool ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given pool. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_other_config (session ref session_ref, pool ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removerepository

Overview:

Remove a repository from the enabled set

Signature:

```
void remove_repository (session ref session_ref, pool ref self, Repository
ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Repository ref`	`value`	The repository to be removed

Minimum Role: client-cert

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removetags

Overview:

Remove the given value from the tags field of the given pool. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: resettelemetryuuid

Overview:

Assign a new UUID to telemetry data.

Signature:

```
void reset_telemetry_uuid (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: retrievewlbconfiguration

Overview:

Retrieves the pool optimization criteria from the workload balancing server

Signature:

```
(string -> string) map retrieve_wlb_configuration (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(string -> string) map`

The configuration used in optimizing this pool

RPC name: retrievewlbrecommendations

Overview:

Retrieves vm migrate recommendations for the pool from the workload balancing server

Signature:

```
(VM ref -> string set) map retrieve_wlb_recommendations (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(VM ref -> string set) map`

The list of vm migration recommendations

RPC name: rotatesecret

Overview:

Signature:

```
void rotate_secret (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-admin

Return Type: `void`

Possible Error Codes: `INTERNAL_ERROR`, `HOST_IS_SLAVE`, `CANNOT_CONTACT_HOST`, `HA_IS_ENABLED`, `NOT_SUPPORTED_DURING_UPGRADE`

RPC name: sendtestpost

Overview:

Send the given body to the given host and port, using HTTPS, and print the response. This is used for debugging the SSL layer.

Signature:

```
string send_test_post (session ref session_ref, string host, int port,
string body)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`host`	
`int`	`port`	
`string`	`body`	

Minimum Role: pool-admin

Return Type: `string`

The response

RPC name: sendwlbconfiguration

Overview:

Sets the pool optimization criteria for the workload balancing server

Signature:

```
void send_wlb_configuration (session ref session_ref, (string -> string)
map config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`(string -> string) map`	`config`	The configuration to use in optimizing this pool

Minimum Role: pool-operator

Return Type: `void`

RPC name: setautoupdatevmsecurebootcertificates

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the autoupdatevmsecurebootcertificates field of the given pool.

Signature:

```
void set_auto_update_vm_secureboot_certificates (session ref session_ref,
pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setconsoleidletimeout

Overview:

Set the console idle timeout for all hosts in the pool

Signature:

```
void set_console_idle_timeout (session ref session_ref, pool ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`int`	`value`	The idle SSH/VNC session timeout in seconds. A value of 0 means no timeout.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setcoordinatorbias

Overview:

Set the coordinatorbias field of the given pool.

Signature:

```
void set_coordinator_bias (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setcrashdumpSR

Overview:

Set the crashdumpSR field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_crash_dump_SR (session ref session_ref, pool ref self, SR ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setcustomueficertificates

Overview:

Set custom UEFI certificates for a pool and all its hosts. Need `allow-custom-uefi-certs` set to true in conf. If empty: default back to Pool.ueficertificates

Signature:

```
void set_custom_uefi_certificates (session ref session_ref, pool ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`value`	The certificates to apply to the pool and its hosts

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setdefaultSR

Overview:

Set the defaultSR field of the given pool.

Signature:

```
void set_default_SR (session ref session_ref, pool ref self, SR ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthcacheenabled

Overview:

Signature:

```
void set_ext_auth_cache_enabled (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	Specifies whether caching is enabled for external authentication

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthcacheexpiry

Overview:

Signature:

```
void set_ext_auth_cache_expiry (session ref session_ref, pool ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The expiry time of entries in the external authentication cache (in seconds - 300 seconds, i.e. 5 minutes, is the default value)

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthcachesize

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_ext_auth_cache_size (session ref session_ref, pool ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The capacity of the external authentication cache

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthmaxthreads

Overview:

Signature:

```
void set_ext_auth_max_threads (session ref session_ref, pool ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The new maximum

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setguiconfig

Overview:

Set the guiconfig field of the given pool.

Signature:

```
void set_gui_config (session ref session_ref, pool ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: sethaallowovercommit

Overview:

Set the haallowovercommit field of the given pool.

Signature:

```
void set_ha_allow_overcommit (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethahostfailurestotolerate

Overview:

Set the maximum number of host failures to consider in the HA VM restart planner

Signature:

```
void set_ha_host_failures_to_tolerate (session ref session_ref, pool ref
self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	New number of host failures to consider

Minimum Role: pool-operator

Return Type: `void`

RPC name: setharebootvmoninternalshutdown

Overview:

Set the harebootvmoninternalshutdown field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_ha_reboot_vm_on_internal_shutdown (session ref session_ref, pool
ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethealthcheckconfig

Overview:

Set the healthcheckconfig field of the given pool.

Signature:

```
void set_health_check_config (session ref session_ref, pool ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: sethttpsonly

Overview:

updates all the host firewalls in the pool to open or close port 80 depending on the value

Signature:

```
void set_https_only (session ref session_ref, pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`value`	true - http port 80 will be blocked, false - http port 80 will be open for the hosts in the pool

Minimum Role: pool-operator

Return Type: `void`

RPC name: setigmpsnoopingenabled

Overview:

Enable or disable IGMP Snooping on the pool.

Signature:

```
void set_igmp_snooping_enabled (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	The pool
`bool`	`value`	Enable or disable IGMP Snooping on the pool

Minimum Role: pool-operator

Return Type: `void`

RPC name: setispsrpending

Overview:

Set the ispsrpending field of the given pool.

Signature:

```
void set_is_psr_pending (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlicenseserver

Overview:

Set the licenseserver field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_license_server (session ref session_ref, pool ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlimitconsolesessions

Overview:

Set the limitconsolesessions field of the given pool.

Signature:

```
void set_limit_console_sessions (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setlivepatchingdisabled

Overview:

Set the livepatchingdisabled field of the given pool.

Signature:

```
void set_live_patching_disabled (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlocalauthmaxthreads

Overview:

Signature:

```
void set_local_auth_max_threads (session ref session_ref, pool ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The new maximum

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setmigrationcompression

Overview:

Set the migrationcompression field of the given pool.

Signature:

```
void set_migration_compression (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the namedescription field of the given pool.

Signature:

```
void set_name_description (session ref session_ref, pool ref self, string
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the namelabel field of the given pool.

Signature:

```
void set_name_label (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_other_config (session ref session_ref, pool ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setpolicynovendordevice

This message is deprecated.

Overview:

Set the policynovendordevice field of the given pool.

Signature:

```
void set_policy_no_vendor_device (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setrepositories

Overview:

Set enabled set of repositories

Signature:

```
void set_repositories (session ref session_ref, pool ref self, Repository ref set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Repository ref set`	`value`	The set of repositories to be enabled

Minimum Role: client-cert

Return Type: `void`

RPC name: setsshautomode

Overview:

Set the SSH auto mode for all hosts in the pool

Signature:

```
void set_ssh_auto_mode (session ref session_ref, pool ref self, bool value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`value`	The SSH auto mode for all hosts in the pool. When true, the SSH port is closed by default and it's open only in case the API is unavailable. When false, the SSH port is always open.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setsshenabledtimeout

Overview:

Set the SSH enabled timeout for all hosts in the pool

Signature:

```
void set_ssh_enabled_timeout (session ref session_ref, pool ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The SSH enabled timeout in seconds. (0 means no timeout, max 2 days)

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setsuspendimageSR

Overview:

Set the suspendimageSR field of the given pool.

Signature:

```
void set_suspend_image_SR (session ref session_ref, pool ref self, SR ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given pool.

Signature:

```
void set_tags (session ref session_ref, pool ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: settelemetrynextcollection

Overview:

Set the timestamp for the next telemetry data collection.

Signature:

```
void set_telemetry_next_collection (session ref session_ref, pool ref
self, datetime value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`datetime`	`value`	The earliest timestamp (in UTC) when the next round of telemetry collection can be carried out.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setueficertificates

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the UEFI certificates for a pool and all its hosts. Deprecated: use setcustomueficertificates instead

Signature:

```
void set_uefi_certificates (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`value`	The certificates to apply to the pool and its hosts

Minimum Role: pool-admin

Return Type: `void`

RPC name: setupdatesyncenabled

Overview:

enable or disable periodic update synchronization depending on the value

Signature:

```
void set_update_sync_enabled (session ref session_ref, pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	true - enable periodic update synchronization, false - disable it

Minimum Role: pool-operator

Return Type: `void`

RPC name: setvmconsoleidletimeout

Overview:

Set the vmconsoleidletimeout field of the given pool.

Signature:

```
void set_vm_console_idle_timeout (session ref session_ref, pool ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setvswitchcontroller

This message is deprecated.

Overview:

Set the IP address of the vswitch controller.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_vswitch_controller (session ref session_ref, string address)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`address`	IP address of the vswitch controller.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setwlbenabled

Overview:

Set the wlbenabled field of the given pool.

Signature:

```
void set_wlb_enabled (session ref session_ref, pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: client-cert

Return Type: `void`

RPC name: setwlbverifycert

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the wlbverifycert field of the given pool.

Signature:

```
void set_wlb_verify_cert (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: syncdatabase

Overview:

Forcibly synchronise the database now

Signature:

```
void sync_database (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: synctrustedcertificatesfrom

Overview:

Download trusted TLS certificates from a remote pool and install them in this pool. Certificates already present locally matched by fingerprint and purpose are skipped.

Signature:

```
Certificate ref set sync_trusted_certificates_from (session ref
session_ref, pool ref self, string remote_pool, session ref
remote_session, string remote_certificate, bool ca)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`remote_pool`	The hostname or IP address of the coordinator of the remote pool from which the certificates are downloaded
`session ref`	`remote_session`	A session obtained from the remote pool, used to authenticate the download
`string`	`remote_certificate`	The PEM-encoded TLS certificate of the remote pool's coordinator, used to verify the TLS connection to the remote pool.
`bool`	`ca`	true for 'ca' or false for 'pinned'

Minimum Role: client-cert

Return Type: `Certificate ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The references of certificates synced.

RPC name: syncupdates

Overview:

Sync with the enabled repository

Signature:

```
string sync_updates (session ref session_ref, pool ref self, bool force,
string token, string token_id, string username, string password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`force`	If true local mirroring repo will be removed before syncing
`string`	`token`	The token for repository client authentication
`string`	`token_id`	The ID of the token
`string`	`username`	The username of the remote pool
`string`	`password`	The password of the remote pool

Minimum Role: client-cert

Return Type: `string`

The SHA256 hash of updateinfo.xml.gz

RPC name: testarchivetarget

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

This call tests if a location is valid

Signature:

```
string test_archive_target (session ref session_ref, pool ref self,
(string -> string) map config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool
`(string -> string) map`	`config`	Location config settings to test

Minimum Role: pool-operator

Return Type: `string`

An XMLRPC result

RPC name: uninstallcacertificate

This message is deprecated.

Overview:

Remove a poolwide TLS CA certificate.

Signature:

```
void uninstall_ca_certificate (session ref session_ref, string name, bool
force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`name`	The certificate name
`bool`	`force`	Remove the DB entry even if the file is non-existent

Minimum Role: client-cert

Return Type: `void`

RPC name: uninstalltrustedcertificate

Overview:

Uninstall a trusted TLS certificate, poolwide.

Signature:

```
void uninstall_trusted_certificate (session ref session_ref, pool ref
self, Certificate ref certificate)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Certificate ref`	`certificate`	The reference of the trusted certificate to be uninstalled

Minimum Role: client-cert

Return Type: `void`

Class: poolpatch

This class is deprecated.

Poolwide patches

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: poolpatch

FIELD	TYPE	QUALIFIER	DESCRIPTION
`after_apply_guidance`	`after_apply_guidance set`	*RO/runtime*	**Deprecated.** What the client should do after this patch has been applied.
`host_patches`	`host_patch ref set`	*RO/runtime*	**Deprecated.** This hosts this patch is applied to.
`name_description`	`string`	*RO/constructor*	**Deprecated.** a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	**Deprecated.** a human-readable name
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`pool_applied`	`bool`	*RO/runtime*	**Deprecated.** This patch should be applied across the entire pool
`pool_update`	`pool_update ref`	*RO/constructor*	**Deprecated.** A reference to the associated pool_update object
`size`	`int`	*RO/runtime*	**Deprecated.** Size of the patch
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`version`	`string`	*RO/constructor*	**Deprecated.** Patch version number

RPCs associated with class: poolpatch

RPC name: addtootherconfig

This message is deprecated.

Overview:

Add the given keyvalue pair to the otherconfig field of the given poolpatch.

Signature:

```
void add_to_other_config (session ref session_ref, pool_patch ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: apply

This message is deprecated.

Overview:

Apply the selected patch to a host and return its output

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string apply (session ref session_ref, pool_patch ref self, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to apply
`host ref`	`host`	The host to apply the patch too

Minimum Role: pool-operator

Return Type: `string`

the output of the patch application process

RPC name: clean

This message is deprecated.

Overview:

Removes the patch's files from the server

Signature:

```
void clean (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to clean up

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: cleanonhost

This message is deprecated.

Overview:

Removes the patch's files from the specified host

Signature:

```
void clean_on_host (session ref session_ref, pool_patch ref self, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to clean up
`host ref`	`host`	The host on which to clean the patch

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

This message is deprecated.

Overview:

Removes the patch's files from all hosts in the pool, and removes the database entries. Only works on unapplied patches.

Signature:

```
void destroy (session ref session_ref, pool_patch ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getafterapplyguidance

This message is deprecated.

Overview:

Get the afterapplyguidance field of the given poolpatch.

Signature:

```
after_apply_guidance set get_after_apply_guidance (session ref
session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `after_apply_guidance set`

value of the field

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the poolpatchs known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
pool_patch ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `pool_patch ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

Return a map of poolpatch references to poolpatch records for all poolpatchs known to the system.

Signature:

```
(pool_patch ref -> pool_patch record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(pool_patch ref -> pool_patch record) map`

records of all objects

RPC name: getbynamelabel

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Get all the poolpatch instances with the given label.

Signature:

```
pool_patch ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `pool_patch ref set`

references to objects with matching names

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the poolpatch instance with the specified UUID.

Signature:

```
pool_patch ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `pool_patch ref`

reference to the object

RPC name: gethostpatches

This message is deprecated.

Overview:

Get the hostpatches field of the given poolpatch.

Signature:

```
host_patch ref set get_host_patches (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_patch ref set`

value of the field

RPC name: getnamedescription

This message is deprecated.

Overview:

Get the name/description field of the given poolpatch.

Signature:

```
string get_name_description (session ref session_ref, pool_patch ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

This message is deprecated.

Overview:

Get the name/label field of the given poolpatch.

Signature:

```
string get_name_label (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the otherconfig field of the given poolpatch.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpoolapplied

This message is deprecated.

Overview:

Get the poolapplied field of the given poolpatch.

Signature:

```
bool get_pool_applied (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getpoolupdate

This message is deprecated.

Overview:

Get the poolupdate field of the given poolpatch.

Signature:

```
pool_update ref get_pool_update (session ref session_ref, pool_patch ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update ref`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given poolpatch.

Signature:

```
pool_patch record get_record (session ref session_ref, pool_patch ref
self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_patch record`

all fields from the object

RPC name: getsize

This message is deprecated.

Overview:

Get the size field of the given poolpatch.

Signature:

```
int get_size (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the uuid field of the given poolpatch.

Signature:

```
string get_uuid (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

This message is deprecated.

Overview:

Get the version field of the given poolpatch.

Signature:

```
string get_version (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: poolapply

This message is deprecated.

Overview:

Apply the selected patch to all hosts in the pool and return a map of hostref > patch output

Signature:

```
void pool_apply (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to apply

Minimum Role: pool-operator

Return Type: `void`

RPC name: poolclean

This message is deprecated.

Overview:

Removes the patch's files from all hosts in the pool, but does not remove the database entries

Signature:

```
void pool_clean (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool_patch ref`	`self`	The patch to clean up

Minimum Role: pool-operator

Return Type: `void`

RPC name: precheck

This message is deprecated.

Overview:

Execute the precheck stage of the selected patch on a host and return its output

Signature:

```
string precheck (session ref session_ref, pool_patch ref self, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch whose prechecks will be run
`host ref`	`host`	The host to run the prechecks on

Minimum Role: pool-operator

Return Type: `string`

the output of the patch prechecks

RPC name: removefromotherconfig

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given poolpatch. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, pool_patch ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given poolpatch.

Signature:

```
void set_other_config (session ref session_ref, pool_patch ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: poolupdate

Poolwide updates to the host software

Fields for class: poolupdate

FIELD	TYPE	QUALIFIER	DESCRIPTION
`after_apply_guidance`	`update_after_apply_guidance set`	*RO/constructor*	What the client should do after this update has been applied.
`enforce_homogeneity`	`bool`	*RO/constructor*	Flag - if true, all hosts in a pool must apply this update
`hosts`	`host ref set`	*RO/runtime*	The hosts that have applied this update.
`installation_size`	`int`	*RO/constructor*	Size of the update in bytes
`key`	`string`	*RO/constructor*	GPG key of the update
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vdi`	`VDI ref`	*RO/constructor*	VDI the update was uploaded to
`version`	`string`	*RO/constructor*	Update version number

RPCs associated with class: poolupdate

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given poolupdate.

Signature:

```
void add_to_other_config (session ref session_ref, pool_update ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: apply

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Apply the selected update to a host

Signature:

```
void apply (session ref session_ref, pool_update ref self, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update to apply
`host ref`	`host`	The host to apply the update to.

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

Overview:

Removes the database entry. Only works on unapplied update.

Signature:

```
void destroy (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update to destroy

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getafterapplyguidance

Overview:

Get the afterapplyguidance field of the given poolupdate.

Signature:

```
update_after_apply_guidance set get_after_apply_guidance (session ref
session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_after_apply_guidance set`

value of the field

RPC name: getall

Overview:

Return a list of all the poolupdates known to the system.

Signature:

```
pool_update ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `pool_update ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of poolupdate references to poolupdate records for all poolupdates known to the system.

Signature:

```
(pool_update ref -> pool_update record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(pool_update ref -> pool_update record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the poolupdate instances with the given label.

Signature:

```
pool_update ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `pool_update ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the poolupdate instance with the specified UUID.

Signature:

```
pool_update ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `pool_update ref`

reference to the object

RPC name: getenforcehomogeneity

Overview:

Get the enforcehomogeneity field of the given poolupdate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_enforce_homogeneity (session ref session_ref, pool_update ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethosts

Overview:

Get the hosts field of the given poolupdate.

Signature:

```
host ref set get_hosts (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getinstallationsize

Overview:

Get the installationsize field of the given poolupdate.

Signature:

```
int get_installation_size (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getkey

Overview:

Get the key field of the given poolupdate.

Signature:

```
string get_key (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given poolupdate.

Signature:

```
string get_name_description (session ref session_ref, pool_update ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given poolupdate.

Signature:

```
string get_name_label (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given poolupdate.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given poolupdate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
pool_update record get_record (session ref session_ref, pool_update ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given poolupdate.

Signature:

```
string get_uuid (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getvdi

Overview:

Get the vdi field of the given poolupdate.

Signature:

```
VDI ref get_vdi (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getversion

Overview:

Get the version field of the given poolupdate.

Signature:

```
string get_version (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: introduce

Overview:

Introduce update VDI

Signature:

```
pool_update ref introduce (session ref session_ref, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI which contains a software update.

Minimum Role: pool-operator

Return Type: `pool_update ref`

the introduced pool update

RPC name: poolapply

Overview:

Apply the selected update to all hosts in the pool

Signature:

```
void pool_apply (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool_update ref`	`self`	The update to apply

Minimum Role: pool-operator

Return Type: `void`

RPC name: poolclean

Overview:

Removes the update's files from all hosts in the pool, but does not revert the update

Signature:

```
void pool_clean (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update to clean up

Minimum Role: pool-operator

Return Type: `void`

RPC name: precheck

Overview:

Execute the precheck stage of the selected update on a host

Signature:

```
livepatch_status precheck (session ref session_ref, pool_update ref self,
host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update whose prechecks will be run
`host ref`	`host`	The host to run the prechecks on.

Minimum Role: pool-operator

Return Type: `livepatch_status`

The precheck pool update

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given poolupdate. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, pool_update ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given poolupdate.

Signature:

```
void set_other_config (session ref session_ref, pool_update ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: proberesult

A set of properties that describe one result element of SR.probe. Result elements and properties can change dynamically based on changes to the the SR.probe inputparameters or the target.

Fields for class: proberesult

FIELD	TYPE	QUALIFIER	DESCRIPTION
`complete`	`bool`	*RO/runtime*	True if this configuration is complete and can be used to call SR.create. False if it requires further iterative calls to SR.probe, to potentially narrow down on a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			configuration that can be used.
`configuration`	`(string -> string) map`	*RO/runtime*	Plugin-specific configuration which describes where and how to locate the storage repository. This may include the physical block device name, a remote NFS server and path or an RBD storage pool.
`extra_info`	`(string -> string) map`	*RO/runtime*	Additional plugin-specific information about this configuration, that might be of use for an API user. This can, for example, include the LUN or the WWPN.
`sr`	`sr_stat record option`	*RO/runtime*	Existing SR found for this configuration

RPCs associated with class: proberesult

Class proberesult has no RPCs associated with it.

Class: PUSB

A physical USB device

Fields for class: PUSB

FIELD	TYPE	QUALIFIER	DESCRIPTION
`description`	`string`	*RO/constructor*	USB device description

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	Physical machine that owns the USB device
`other_config`	`(string -> string) map`	*RW*	additional configuration
`passthrough_enabled`	`bool`	*RO/runtime*	enabled for passthrough
`path`	`string`	*RO/constructor*	port path of USB device
`product_desc`	`string`	*RO/constructor*	product description of the USB device
`product_id`	`string`	*RO/constructor*	product id of the USB device
`serial`	`string`	*RO/constructor*	serial of the USB device
`speed`	`float`	*RO/constructor*	USB device speed
`USB_group`	`USB_group ref`	*RO/constructor*	USB group the PUSB is contained in
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_desc`	`string`	*RO/constructor*	vendor description of the USB device
`vendor_id`	`string`	*RO/constructor*	vendor id of the USB device
`version`	`string`	*RO/constructor*	USB device version

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: PUSB

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PUSB.

Signature:

```
void add_to_other_config (session ref session_ref, PUSB ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PUSBs known to the system.

Signature:

```
PUSB ref set get_all (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PUSB ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PUSB references to PUSB records for all PUSBs known to the system.

Signature:

```
(PUSB ref -> PUSB record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PUSB ref -> PUSB record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PUSB instance with the specified UUID.

Signature:

```
PUSB ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PUSB ref`

reference to the object

RPC name: getdescription

Overview:

Get the description field of the given PUSB.

Signature:

```
string get_description (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given PUSB.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref get_host (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PUSB.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PUSB ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpassthroughenabled

Overview:

Get the passthroughenabled field of the given PUSB.

Signature:

```
bool get_passthrough_enabled (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getpath

Overview:

Get the path field of the given PUSB.

Signature:

```
string get_path (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getproductdesc

Overview:

Get the productdesc field of the given PUSB.

Signature:

```
string get_product_desc (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getproductid

Overview:

Get the productid field of the given PUSB.

Signature:

```
string get_product_id (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PUSB.

Signature:

```
PUSB record get_record (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PUSB record`

all fields from the object

RPC name: getserial

Overview:

Get the serial field of the given PUSB.

Signature:

```
string get_serial (session ref session_ref, PUSB ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getspeed

Overview:

Get the speed field of the given PUSB.

Signature:

```
float get_speed (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getUSBgroup

Overview:

Get the USBgroup field of the given PUSB.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
USB_group ref get_USB_group (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `USB_group ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PUSB.

Signature:

```
string get_uuid (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendordesc

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the vendordesc field of the given PUSB.

Signature:

```
string get_vendor_desc (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorid

Overview:

Get the vendorid field of the given PUSB.

Signature:

```
string get_vendor_id (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getversion

Overview:

Get the version field of the given PUSB.

Signature:

```
string get_version (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PUSB. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PUSB ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: scan

Overview:

Signature:

```
void scan (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PUSB.

Signature:

```
void set_other_config (session ref session_ref, PUSB ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: setpassthroughenabled

Overview:

Signature:

```
void set_passthrough_enabled (session ref session_ref, PUSB ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	this PUSB
`bool`	`value`	passthrough is enabled when true and disabled with false

Minimum Role: pool-admin

Return Type: `void`

Class: PVScachestorage

Describes the storage that is available to a PVS site for caching purposes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: PVScachestorage

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	The host on which this object defines PVS cache storage
`site`	`PVS_site ref`	*RO/constructor*	The PVS_site for which this object defines the storage
`size`	`int`	*RO/constructor*	The size of the cache VDI (in bytes)
`SR`	`SR ref`	*RO/constructor*	SR providing storage for the PVS cache
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VDI`	`VDI ref`	*RO/runtime*	The VDI used for caching

RPCs associated with class: PVScachestorage

RPC name: create

Overview:

Create a new PVScachestorage instance, and return its handle.

Signature:

```
PVS_cache_storage ref create (session ref session_ref, PVS_cache_storage record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PVS_cache_storage record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `PVS_cache_storage ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified PVScachestorage instance.

Signature:

```
void destroy (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PVScachestorages known to the system.

Signature:

```
PVS_cache_storage ref set get_all (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PVS_cache_storage ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVScachestorage references to PVScachestorage records for all PVScachestorages known to the system.

Signature:

```
(PVS_cache_storage ref -> PVS_cache_storage record) map get_all_records
(session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_cache_storage ref -> PVS_cache_storage record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PVScachestorage instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_cache_storage ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_cache_storage ref`

reference to the object

RPC name: gethost

Overview:

Get the host field of the given PVScachestorage.

Signature:

```
host ref get_host (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getrecord

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a record containing the current state of the given PVScachestorage.

Signature:

```
PVS_cache_storage record get_record (session ref session_ref,
PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_cache_storage record`

all fields from the object

RPC name: getsite

Overview:

Get the site field of the given PVScachestorage.

Signature:

```
PVS_site ref get_site (session ref session_ref, PVS_cache_storage ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getsize

Overview:

Get the size field of the given PVScachestorage.

Signature:

```
int get_size (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getSR

Overview:

Get the SR field of the given PVScachestorage.

Signature:

```
SR ref get_SR (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVScachestorage.

Signature:

```
string get_uuid (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDI

Overview:

Get the VDI field of the given PVScachestorage.

Signature:

```
VDI ref get_VDI (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

Class: PVSproxy

a proxy connects a VM/VIF with a PVS site

Fields for class: PVSproxy

FIELD	TYPE	QUALIFIER	DESCRIPTION
`currently_attached`	`bool`	*RO/runtime*	true = VM is currently proxied
`site`	`PVS_site ref`	*RO/constructor*	PVS site this proxy is part of
`status`	`pvs_proxy_status`	*RO/runtime*	The run-time status of the proxy
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VIF`	`VIF ref`	*RO/constructor*	VIF of the VM using the proxy

RPCs associated with class: PVSproxy

RPC name: create

Overview:

Configure a VM/VIF to use a PVS proxy

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_proxy ref create (session ref session_ref, PVS_site ref site, VIF ref
VIF)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`site`	PVS site that we proxy for
`VIF ref`	`VIF`	VIF for the VM that needs to be proxied

Minimum Role: pool-operator

Return Type: `PVS_proxy ref`

The reference of the created PVS proxy

RPC name: destroy

Overview:

remove or switch off a PVS proxy for this VM

Signature:

```
void destroy (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	this PVS proxy

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getall

Overview:

Return a list of all the PVSproxys known to the system.

Signature:

```
PVS_proxy ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PVS_proxy ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVSproxy references to PVSproxy records for all PVSproxys known to the system.

Signature:

```
(PVS_proxy ref -> PVS_proxy record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_proxy ref -> PVS_proxy record) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PVSproxy instance with the specified UUID.

Signature:

```
PVS_proxy ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_proxy ref`

reference to the object

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given PVSproxy.

Signature:

```
bool get_currently_attached (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PVSproxy.

Signature:

```
PVS_proxy record get_record (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_proxy record`

all fields from the object

RPC name: getsite

Overview:

Get the site field of the given PVSproxy.

Signature:

```
PVS_site ref get_site (session ref session_ref, PVS_proxy ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site ref`

value of the field

RPC name: getstatus

Overview:

Get the status field of the given PVSproxy.

Signature:

```
pvs_proxy_status get_status (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pvs_proxy_status`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVSproxy.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVIF

Overview:

Get the VIF field of the given PVSproxy.

Signature:

```
VIF ref get_VIF (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF ref`

value of the field

Class: PVSserver

individual machine serving provisioning block data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: PVSserver

FIELD	TYPE	QUALIFIER	DESCRIPTION
`addresses`	`string set`	*RO/constructor*	IPv4 addresses of this server
`first_port`	`int`	*RO/constructor*	First UDP port accepted by this server
`last_port`	`int`	*RO/constructor*	Last UDP port accepted by this server
`site`	`PVS_site ref`	*RO/constructor*	PVS site this server is part of
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PVSserver

RPC name: forget

Overview:

forget a PVS server

Signature:

```
void forget (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	this PVS server

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: getaddresses

Overview:

Get the addresses field of the given PVSserver.

Signature:

```
string set get_addresses (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getall

Overview:

Return a list of all the PVSservers known to the system.

Signature:

```
PVS_server ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `PVS_server ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVSserver references to PVSserver records for all PVSservers known to the system.

Signature:

```
(PVS_server ref -> PVS_server record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_server ref -> PVS_server record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PVSserver instance with the specified UUID.

Signature:

```
PVS_server ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_server ref`

reference to the object

RPC name: getfirstport

Overview:

Get the firstport field of the given PVSserver.

Signature:

```
int get_first_port (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getlastport

Overview:

Get the lastport field of the given PVSserver.

Signature:

```
int get_last_port (session ref session_ref, PVS_server ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PVSserver.

Signature:

```
PVS_server record get_record (session ref session_ref, PVS_server ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_server record`

all fields from the object

RPC name: getsite

Overview:

Get the site field of the given PVSserver.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
PVS_site ref get_site (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVSserver.

Signature:

```
string get_uuid (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: introduce

Overview:

introduce new PVS server

Signature:

```
PVS_server ref introduce (session ref session_ref, string set addresses,
int first_port, int last_port, PVS_site ref site)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`addresses`	IPv4/IPv6 addresses of the server
`int`	`first_port`	first UDP port accepted by this server
`int`	`last_port`	last UDP port accepted by this server
`PVS_site ref`	`site`	PVS site this server is a part of

Minimum Role: pool-operator

Return Type: `PVS_server ref`

the new PVS server

Class: PVSsite

machines serving blocks of data for provisioning VMs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: PVSsite

FIELD	TYPE	QUALIFIER	DESCRIPTION
`cache_storage`	`PVS_cache_storage ref set`	*RO/runtime*	The SR used by PVS proxy for the cache
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`proxies`	`PVS_proxy ref set`	*RO/runtime*	The set of proxies associated with the site
`PVS_uuid`	`string`	*RO/constructor*	Unique identifier of the PVS site, as configured in PVS
`servers`	`PVS_server ref set`	*RO/runtime*	The set of PVS servers in the site
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PVSsite

RPC name: forget

Overview:

Remove a site's meta data

Signature:

```
void forget (session ref session_ref, PVS_site ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	this PVS site

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `PVS_SITE_CONTAINS_RUNNING_PROXIES`, `PVS_SITE_CONTAINS_SERVERS`

RPC name: getall

Overview:

Return a list of all the PVSsites known to the system.

Signature:

```
PVS_site ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PVS_site ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVSsite references to PVSsite records for all PVSsites known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(PVS_site ref -> PVS_site record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_site ref -> PVS_site record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the PVSsite instances with the given label.

Signature:

```
PVS_site ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `PVS_site ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a reference to the PVSsite instance with the specified UUID.

Signature:

```
PVS_site ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_site ref`

reference to the object

RPC name: getcachestorage

Overview:

Get the cachestorage field of the given PVSsite.

Signature:

```
PVS_cache_storage ref set get_cache_storage (session ref session_ref,
PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_cache_storage ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given PVSsite.

Signature:

```
string get_name_description (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given PVSsite.

Signature:

```
string get_name_label (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getproxies

Overview:

Get the proxies field of the given PVSsite.

Signature:

```
PVS_proxy ref set get_proxies (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_proxy ref set`

value of the field

RPC name: getPVSuuid

Overview:

Get the PVSuuid field of the given PVSsite.

Signature:

```
string get_PVS_uuid (session ref session_ref, PVS_site ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PVSsite.

Signature:

```
PVS_site record get_record (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site record`

all fields from the object

RPC name: getservers

Overview:

Get the servers field of the given PVSsite.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_server ref set get_servers (session ref session_ref, PVS_site ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_server ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVSsite.

Signature:

```
string get_uuid (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: introduce

Overview:

Introduce new PVS site

Signature:

```
PVS_site ref introduce (session ref session_ref, string name_label, string name_description, string PVS_uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	name of the PVS site
`string`	`name_description`	description of the PVS site
`string`	`PVS_uuid`	unique identifier of the PVS site

Minimum Role: pool-operator

Return Type: `PVS_site ref`

the new PVS site

RPC name: setnamedescription

Overview:

Set the name/description field of the given PVSsite.

Signature:

```
void set_name_description (session ref session_ref, PVS_site ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given PVSsite.

Signature:

```
void set_name_label (session ref session_ref, PVS_site ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setPVSuuid

Overview:

Update the PVS UUID of the PVS site

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_PVS_uuid (session ref session_ref, PVS_site ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	this PVS site
`string`	`value`	PVS UUID to be used

Minimum Role: pool-operator

Return Type: `void`

Class: Repository

Repository for updates

Fields for class: Repository

FIELD	TYPE	QUALIFIER	DESCRIPTION
`binary_url`	`string`	*RO/constructor*	Base URL of binary packages in this repository
`certificate`	`string`	*RO/constructor*	The certificate of the host which hosts this repository
`gpgkey_path`	`string`	*RO/constructor*	The file name of the GPG public key of this repository
`hash`	`string`	*RO/runtime*	SHA256 checksum of latest updateinfo.xml.gz in this repository if its 'update' is true

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`origin`	`origin`	*RO/constructor*	The origin of this repository. 'remote' if the origin of the repository is a remote one, 'bundle' if the origin of the repository is a local bundle file, 'remote_pool' if the origin of the repository is a remote pool
`source_url`	`string`	*RO/constructor*	Base URL of source packages in this repository
`up_to_date`	`bool`	*RO/runtime*	**Removed.** True if all hosts in pool is up to date with this repository
`update`	`bool`	*RO/constructor*	True if updateinfo.xml in this repository needs to be parsed
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: Repository

RPC name: forget

Overview:

Remove the repository record from the database

Signature:

```
void forget (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	The repository to be removed from the database

Minimum Role: client-cert

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the Repositorys known to the system.

Signature:

```
Repository ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `Repository ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Repository references to Repository records for all Repositorys known to the system.

Signature:

```
(Repository ref -> Repository record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Repository ref -> Repository record) map`

records of all objects

RPC name: getbinaryurl

Overview:

Get the binaryurl field of the given Repository.

Signature:

```
string get_binary_url (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbynamelabel

Overview:

Get all the Repository instances with the given label.

Signature:

```
Repository ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `Repository ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the Repository instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Repository ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Repository ref`

reference to the object

RPC name: getcertificate

Overview:

Get the certificate field of the given Repository.

Signature:

```
string get_certificate (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getgpgkeypath

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the gpgkeypath field of the given Repository.

Signature:

```
string get_gpgkey_path (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethash

Overview:

Get the hash field of the given Repository.

Signature:

```
string get_hash (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnamedescription

Overview:

Get the name/description field of the given Repository.

Signature:

```
string get_name_description (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given Repository.

Signature:

```
string get_name_label (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getorigin

Overview:

Get the origin field of the given Repository.

Signature:

```
origin get_origin (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `origin`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Repository.

Signature:

```
Repository record get_record (session ref session_ref, Repository ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Repository record`

all fields from the object

RPC name: getsourceurl

Overview:

Get the sourceurl field of the given Repository.

Signature:

```
string get_source_url (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuptodate

This message is removed.

Overview:

Get the uptodate field of the given Repository.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_up_to_date (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getupdate

Overview:

Get the update field of the given Repository.

Signature:

```
bool get_update (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the uuid field of the given Repository.

Signature:

```
string get_uuid (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: introduce

Overview:

Add the configuration for a new remote repository

Signature:

```
Repository ref introduce (session ref session_ref, string name_label,
string name_description, string binary_url, string source_url, bool
update, string gpgkey_path)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	The name of the repository
`string`	`name_description`	The description of the repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`binary_url`	Base URL of binary packages in this repository
`string`	`source_url`	Base URL of source packages in this repository
`bool`	`update`	True if the repository is an update repository. This means that updateinfo.xml will be parsed
`string`	`gpgkey_path`	The GPG public key file name

Minimum Role: client-cert

Return Type: `Repository ref`

The ref of the created repository record.

RPC name: introducebundle

Overview:

Add the configuration for a new bundle repository

Signature:

```
Repository ref introduce_bundle (session ref session_ref, string
name_label, string name_description)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	The name of the repository
`string`	`name_description`	The description of the repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: client-cert

Return Type: `Repository ref`

The ref of the created repository record.

RPC name: introduceremotepool

Overview:

Add the configuration for a new remote pool repository

Signature:

```
Repository ref introduce_remote_pool (session ref session_ref, string name_label, string name_description, string binary_url, string certificate)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	The name of the repository
`string`	`name_description`	The description of the repository
`string`	`binary_url`	Base URL of binary packages in the local repository of this remote pool in https://<coordinator-ip>/repository/ enabled format
`string`	`certificate`	The host certificate of the coordinator of the remote pool

Minimum Role: client-cert

Return Type: `Repository ref`

The ref of the created repository record.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setgpgkeypath

Overview:

Set the file name of the GPG public key of the repository

Signature:

```
void set_gpgkey_path (session ref session_ref, Repository ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	The repository
`string`	`value`	The file name of the GPG public key of the repository

Minimum Role: client-cert

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given Repository.

Signature:

```
void set_name_description (session ref session_ref, Repository ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Repository ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: client-cert

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given Repository.

Signature:

```
void set_name_label (session ref session_ref, Repository ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: client-cert

Return Type: `void`

Class: role

A set of permissions associated with a subject

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: role

FIELD	TYPE	QUALIFIER	DESCRIPTION
`is_internal`	`bool`	*RO/runtime*	Indicates whether the role is only to be assigned internally by xapi, or can be used by clients
`name_description`	`string`	*RO/constructor*	what this role is for
`name_label`	`string`	*RO/constructor*	a short user-friendly name for the role
`subroles`	`role ref set`	*RO/constructor*	a list of pointers to other roles or permissions
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: role

RPC name: getall

Overview:

Return a list of all the roles known to the system.

Signature:

```
role ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `role ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

references to all objects

RPC name: getallrecords

Overview:

Return a map of role references to role records for all roles known to the system.

Signature:

```
(role ref -> role record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(role ref -> role record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the role instances with the given label.

Signature:

```
role ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `role ref set`

references to objects with matching names

RPC name: getbypermission

Overview:

This call returns a list of roles given a permission

Signature:

```
role ref set get_by_permission (session ref session_ref, role ref
permission)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`permission`	a reference to a permission

Minimum Role: read-only

Return Type: `role ref set`

a list of references to roles

RPC name: getbypermissionnamelabel

Overview:

This call returns a list of roles given a permission name

Signature:

```
role ref set get_by_permission_name_label (session ref session_ref, string
label)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	The short friendly name of the role

Minimum Role: read-only

Return Type: `role ref set`

a list of references to roles

RPC name: getbyuuid

Overview:

Get a reference to the role instance with the specified UUID.

Signature:

```
role ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `role ref`

reference to the object

RPC name: getisinternal

Overview:

Get the isinternal field of the given role.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_is_internal (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given role.

Signature:

```
string get_name_description (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the name/label field of the given role.

Signature:

```
string get_name_label (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpermissions

Overview:

This call returns a list of permissions given a role

Signature:

```
role ref set get_permissions (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	a reference to a role

Minimum Role: read-only

Return Type: `role ref set`

a list of permissions

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpermissionsnamelabel

Overview:

This call returns a list of permission names given a role

Signature:

```
string set get_permissions_name_label (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	a reference to a role

Minimum Role: read-only

Return Type: `string set`

a list of permission names

RPC name: getrecord

Overview:

Get a record containing the current state of the given role.

Signature:

```
role record get_record (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `role record`

all fields from the object

RPC name: getsubroles

Overview:

Get the subroles field of the given role.

Signature:

```
role ref set get_subroles (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `role ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given role.

Signature:

```
string get_uuid (session ref session_ref, role ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: SDNcontroller

Describes the SDN controller that is to connect with the pool

Fields for class: SDNcontroller

FIELD	TYPE	QUALIFIER	DESCRIPTION
`address`	`string`	*RO/constructor*	IP address of the controller
`port`	`int`	*RO/constructor*	TCP port of the controller
`protocol`	`sdn_controller_protocol`	*RO/constructor*	Protocol to connect with SDN controller
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: SDNcontroller

RPC name: forget

Overview:

Remove the OVS manager of the pool and destroy the db record.

Signature:

```
void forget (session ref session_ref, SDN_controller ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	this SDN controller

Minimum Role: pool-operator

Return Type: `void`

RPC name: getaddress

Overview:

Get the address field of the given SDNcontroller.

Signature:

```
string get_address (session ref session_ref, SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getall

Overview:

Return a list of all the SDNcontrollers known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SDN_controller ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `SDN_controller ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of SDNcontroller references to SDNcontroller records for all SDNcontrollers known to the system.

Signature:

```
(SDN_controller ref -> SDN_controller record) map get_all_records (session
ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(SDN_controller ref -> SDN_controller record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the SDNcontroller instance with the specified UUID.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
SDN_controller ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `SDN_controller ref`

reference to the object

RPC name: getport

Overview:

Get the port field of the given SDNcontroller.

Signature:

```
int get_port (session ref session_ref, SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getprotocol

Overview:

Get the protocol field of the given SDNcontroller.

Signature:

```
sdn_controller_protocol get_protocol (session ref session_ref,
SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `sdn_controller_protocol`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given SDNcontroller.

Signature:

```
SDN_controller record get_record (session ref session_ref, SDN_controller
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `SDN_controller record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given SDNcontroller.

Signature:

```
string get_uuid (session ref session_ref, SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: introduce

Overview:

Introduce an SDN controller to the pool.

Signature:

```
SDN_controller ref introduce (session ref session_ref,
sdn_controller_protocol protocol, string address, int port)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`sdn_controller_protocol`	`protocol`	Protocol to connect with the controller.
`string`	`address`	IP address of the controller.
`int`	`port`	TCP port of the controller.

Minimum Role: pool-operator

Return Type: `SDN_controller ref`

the introduced SDN controller

Class: secret

A secret

Fields for class: secret

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	other_config
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`value`	`string`	*RW*	the secret

RPCs associated with class: secret

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given secret.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_other_config (session ref session_ref, secret ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new secret instance, and return its handle.

Signature:

```
secret ref create (session ref session_ref, secret record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `secret ref`

reference to the newly created object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Destroy the specified secret instance.

Signature:

```
void destroy (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the secrets known to the system.

Signature:

```
secret ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `secret ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of secret references to secret records for all secrets known to the system.

Signature:

```
(secret ref -> secret record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `(secret ref -> secret record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the secret instance with the specified UUID.

Signature:

```
secret ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `secret ref`

reference to the object

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given secret.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given secret.

Signature:

```
secret record get_record (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `secret record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given secret.

Signature:

```
string get_uuid (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `string`

value of the field

RPC name: getvalue

Overview:

Get the value field of the given secret.

Signature:

```
string get_value (session ref session_ref, secret ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given secret. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, secret ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the otherconfig field of the given secret.

Signature:

```
void set_other_config (session ref session_ref, secret ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setvalue

Overview:

Set the value field of the given secret.

Signature:

```
void set_value (session ref session_ref, secret ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Class: session

A session

Fields for class: session

FIELD	TYPE	QUALIFIER	DESCRIPTION
`auth_user_name`	`string`	*RO/runtime*	the subject name of the user that was externally authenticated. If a session instance has is_local_superuser set, then the value of this field is undefined.
`auth_user_sid`	`string`	*RO/runtime*	the subject identifier of the user that was externally authenticated. If a session instance has is_local_superuser set, then the value of this field is undefined.
`client_certificate`	`bool`	*RO/runtime*	indicates whether this session was authenticated using a client certificate
`is_local_superuser`	`bool`	*RO/runtime*	true iff this session was created using local superuser credentials
`last_active`	`datetime`	*RO/runtime*	Timestamp for last time session was active

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`originator`	`string`	*RO/runtime*	a key string provided by a API user to distinguish itself from other users sharing the same login name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`parent`	`session ref`	*RO/constructor*	references the parent session that created this session
`pool`	`bool`	*RO/runtime*	True if this session relates to a intra-pool login, false otherwise
`rbac_permissions`	`string set`	*RO/constructor*	list with all RBAC permissions for this session
`subject`	`subject ref`	*RO/runtime*	references the subject instance that created the session. If a session instance has is_local_superuser set, then the value of this field is undefined.
`tasks`	`task ref set`	*RO/runtime*	list of tasks created using the current session
`this_host`	`host ref`	*RO/runtime*	Currently connected host
`this_user`	`user ref`	*RO/runtime*	Currently connected user

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`validation_time`	`datetime`	*RO/runtime*	time when session was last validated

RPCs associated with class: session

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given session.

Signature:

```
void add_to_other_config (session ref session_ref, session ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: changepassword

Overview:

Change the account password; if your session is authenticated with root privileges then the oldpwd is validated and the newpwd is set regardless

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void change_password (session ref session_ref, string old_pwd, string
new_pwd)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`old_pwd`	Old password for account
`string`	`new_pwd`	New password for account

Return Type: `void`

RPC name: createfromdbfile

Overview:

Signature:

```
session ref create_from_db_file (session ref session_ref, string filename)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`filename`	Database dump filename.

Return Type: `session ref`

ID of newly created session

RPC name: getallsubjectidentifiers

Overview:

Return a list of all the user subjectidentifiers of all existing sessions

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string set get_all_subject_identifiers (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `string set`

The list of user subjectidentifiers of all existing sessions

RPC name: getauthusername

Overview:

Get the authusername field of the given session.

Signature:

```
string get_auth_user_name (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getauthusersid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the authusersid field of the given session.

Signature:

```
string get_auth_user_sid (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the session instance with the specified UUID.

Signature:

```
session ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `session ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getclientcertificate

Overview:

Get the clientcertificate field of the given session.

Signature:

```
bool get_client_certificate (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getislocalsuperuser

Overview:

Get the islocalsuperuser field of the given session.

Signature:

```
bool get_is_local_superuser (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getlastactive

Overview:

Get the lastactive field of the given session.

Signature:

```
datetime get_last_active (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getoriginator

Overview:

Get the originator field of the given session.

Signature:

```
string get_originator (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given session.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, session
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getparent

Overview:

Get the parent field of the given session.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
session ref get_parent (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `session ref`

value of the field

RPC name: getpool

Overview:

Get the pool field of the given session.

Signature:

```
bool get_pool (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrbacpermissions

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the rbacpermissions field of the given session.

Signature:

```
string set get_rbac_permissions (session ref session_ref, session ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given session.

Signature:

```
session record get_record (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `session record`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

all fields from the object

RPC name: getsubject

Overview:

Get the subject field of the given session.

Signature:

```
subject ref get_subject (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `subject ref`

value of the field

RPC name: gettasks

Overview:

Get the tasks field of the given session.

Signature:

```
task ref set get_tasks (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `task ref set`

value of the field

RPC name: getthishost

Overview:

Get the thishost field of the given session.

Signature:

```
host ref get_this_host (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getthisuser

Overview:

Get the thisuser field of the given session.

Signature:

```
user ref get_this_user (session ref session_ref, session ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `user ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given session.

Signature:

```
string get_uuid (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvalidationtime

Overview:

Get the validationtime field of the given session.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
datetime get_validation_time (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: locallogout

Overview:

Log out of local session.

Signature:

```
void local_logout (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-admin

Return Type: `void`

RPC name: loginwithpassword

Overview:

Attempt to authenticate the user, returning a session reference if successful

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
session ref login_with_password (string uname, string pwd, string version,
string originator)
```

Arguments:

TYPE	NAME	DESCRIPTION
`string`	`uname`	Username for login.
`string`	`pwd`	Password for login.
`string`	`version`	Client API version.
`string`	`originator`	Key string for distinguishing different API users sharing the same login name.

Minimum Role: read-only

Return Type: `session ref`

reference of newly created session

Possible Error Codes: `SESSION_AUTHENTICATION_FAILED`, `HOST_IS_SLAVE`

RPC name: logout

Overview:

Log out of a session

Signature:

```
void logout (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: logoutsubjectidentifier

Overview:

Log out all sessions associated to a user subjectidentifier, except the session associated with the context calling this function

Signature:

```
void logout_subject_identifier (session ref session_ref, string
subject_identifier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`subject_identifier`	User subject-identifier of the sessions to be destroyed

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given session. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, session ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given session.

Signature:

```
void set_other_config (session ref session_ref, session ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: slavelocalloginwithpassword

Overview:

Authenticate locally against a slave in emergency mode. Note the resulting sessions are only good for use on this host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
session ref slave_local_login_with_password (string uname, string pwd)
```

Arguments:

TYPE	NAME	DESCRIPTION
`string`	`uname`	Username for login.
`string`	`pwd`	Password for login.

Minimum Role: pool-admin

Return Type: `session ref`

ID of newly created session

Class: SM

A storage manager plugin

Fields for class: SM

FIELD	TYPE	QUALIFIER	DESCRIPTION
`capabilities`	`string set`	*RO/runtime*	**Deprecated.** capabilities of the SM plugin
`configuration`	`(string -> string) map`	*RO/runtime*	names and descriptions of device config keys
`copyright`	`string`	*RO/runtime*	Entity which owns the copyright of this plugin
`driver_filename`	`string`	*RO/runtime*	filename of the storage driver
`features`	`(string -> int) map`	*RO/runtime*	capabilities of the SM plugin, with

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			capability version numbers
`name_description`	`string`	*RO/runtime*	a notes field containing human-readable description
`name_label`	`string`	*RO/runtime*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`required_api_version`	`string`	*RO/runtime*	Minimum SM API version required on the server
`required_cluster_stack`	`string set`	*RO/runtime*	The storage plugin requires that one of these cluster stacks is configured and running.
`supported_image_formats`	`image_format_type set`	*RO/runtime*	Image formats supported by the SR: raw, vhd, qcow2
`type`	`string`	*RO/runtime*	SR.type
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor`	`string`	*RO/runtime*	Vendor who created this plugin
`version`	`string`	*RO/runtime*	Version of the plugin

RPCs associated with class: SM

RPC name: addtootherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Add the given keyvalue pair to the otherconfig field of the given SM.

Signature:

```
void add_to_other_config (session ref session_ref, SM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the SMs known to the system.

Signature:

```
SM ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `SM ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

references to all objects

RPC name: getallrecords

Overview:

Return a map of SM references to SM records for all SMs known to the system.

Signature:

```
(SM ref -> SM record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(SM ref -> SM record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the SM instances with the given label.

Signature:

```
SM ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `SM ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the SM instance with the specified UUID.

Signature:

```
SM ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `SM ref`

reference to the object

RPC name: getcapabilities

This message is deprecated.

Overview:

Get the capabilities field of the given SM.

Signature:

```
string set get_capabilities (session ref session_ref, SM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getconfiguration

Overview:

Get the configuration field of the given SM.

Signature:

```
(string -> string) map get_configuration (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcopyright

Overview:

Get the copyright field of the given SM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_copyright (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdriverfilename

Overview:

Get the driverfilename field of the given SM.

Signature:

```
string get_driver_filename (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfeatures

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the features field of the given SM.

Signature:

```
(string -> int) map get_features (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> int) map`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given SM.

Signature:

```
string get_name_description (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnamelabel

Overview:

Get the name/label field of the given SM.

Signature:

```
string get_name_label (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given SM.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, SM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given SM.

Signature:

```
SM record get_record (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SM record`

all fields from the object

RPC name: getrequiredapiversion

Overview:

Get the requiredapiversion field of the given SM.

Signature:

```
string get_required_api_version (session ref session_ref, SM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrequiredclusterstack

Overview:

Get the requiredclusterstack field of the given SM.

Signature:

```
string set get_required_cluster_stack (session ref session_ref, SM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getsupportedimageformats

Overview:

Get the supportedimageformats field of the given SM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
image_format_type set get_supported_image_formats (session ref
session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `image_format_type set`

value of the field

RPC name: gettype

Overview:

Get the type field of the given SM.

Signature:

```
string get_type (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given SM.

Signature:

```
string get_uuid (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendor

Overview:

Get the vendor field of the given SM.

Signature:

```
string get_vendor (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getversion

Overview:

Get the version field of the given SM.

Signature:

```
string get_version (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given SM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, SM ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given SM.

Signature:

```
void set_other_config (session ref session_ref, SM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: SR

A storage repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: SR

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`storage_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this SR
`clustered`	`bool`	*RO/runtime*	True if the SR is using aggregated local storage
`content_type`	`string`	*RO/constructor*	the type of the SR's content, if required (for example, ISOs)
`current_operations`	`(string -> storage_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`introduced_by`	`DR_task ref`	*RO/runtime*	The disaster recovery task which introduced this SR
`is_tools_sr`	`bool`	*RO/runtime*	True if this is the SR that contains the Tools ISO VDIs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`local_cache_enabled`	`bool`	*RO/runtime*	True if this SR is assigned to be the local cache for its host
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`PBDs`	`PBD ref set`	*RO/runtime*	describes how particular hosts can see this storage repository
`physical_size`	`int`	*RO/constructor*	total physical size of the repository (in bytes)
`physical_utilisation`	`int`	*RO/runtime*	physical space currently utilised on this storage repository (in bytes). Note that for sparse disk formats, physical_utilisation may be less than virtual_allocation
`shared`	`bool`	*RO/runtime*	true if this SR is (capable of being) shared between multiple hosts
`sm_config`	`(string -> string) map`	*RW*	SM dependent data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`type`	`string`	*RO/constructor*	type of the storage repository
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VDIs`	`VDI ref set`	*RO/runtime*	all virtual disks known to this storage repository
`virtual_allocation`	`int`	*RO/runtime*	sum of virtual_sizes of all VDIs in this storage repository (in bytes)

RPCs associated with class: SR

RPC name: addtags

Overview:

Add the given value to the tags field of the given SR. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, SR ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`value`	New value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given SR.

Signature:

```
void add_to_other_config (session ref session_ref, SR ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtosmconfig

Overview:

Add the given keyvalue pair to the smconfig field of the given SR.

Signature:

```
void add_to_sm_config (session ref session_ref, SR ref self, string key,
string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: assertcanhosthastatefile

Overview:

Returns successfully if the given SR can host an HA statefile. Otherwise returns an error to explain why not

Signature:

```
void assert_can_host_ha_statefile (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to query

Minimum Role: pool-operator

Return Type: `void`

RPC name: assertsupportsdatabasereplication

Overview:

Returns successfully if the given SR supports database replication. Otherwise returns an error to explain why not.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void assert_supports_database_replication (session ref session_ref, SR ref
sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to query

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new Storage Repository and introduce it into the managed system, creating both SR record and PBD record to attach it to current host with specified deviceconfig parameters

Signature:

```
SR ref create (session ref session_ref, host ref host, (string -> string)
map device_config, int physical_size, string name_label, string
name_description, string type, string content_type, bool shared, (string
-> string) map sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`int`	`physical_size`	The physical size of the new storage repository
`string`	`name_label`	The name of the new storage repository
`string`	`name_description`	The description of the new storage repository
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`string`	`content_type`	The type of the new SRs content, if required (for example, ISOs)
`bool`	`shared`	True if the SR (is capable of) being shared by multiple hosts
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `SR ref`

The reference of the newly created Storage Repository.

Possible Error Codes: `SR_UNKNOWN_DRIVER`

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this SR

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
blob ref create_new_blob (session ref session_ref, SR ref sr, string name,
string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: destroy

Overview:

Destroy specified SR, removing SRrecord from database and remove SR from disk. In order to affect this operation the appropriate deviceconfig is read from the specified SR's PBD on current host

Signature:

```
void destroy (session ref session_ref, SR ref sr)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `SR_HAS_PBD`

RPC name: disabledatabasereplication

Overview:

Signature:

```
void disable_database_replication (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to which metadata should be no longer replicated

Minimum Role: pool-operator

Return Type: `void`

RPC name: enabledatabasereplication

Overview:

Signature:

```
void enable_database_replication (session ref session_ref, SR ref sr)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to which metadata should be replicated

Minimum Role: pool-operator

Return Type: `void`

RPC name: forget

Overview:

Removing specified SRrecord from database, without attempting to remove SR from disk

Signature:

```
void forget (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `SR_HAS_PBD`

RPC name: forgetdatasourcearchives

Overview:

Forget the recorded statistics related to the specified data source

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void forget_data_source_archives (session ref session_ref, SR ref sr,
string data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`data_source`	The data source whose archives are to be forgotten

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the SRs known to the system.

Signature:

```
SR ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `SR ref set`

references to all objects

RPC name: getallrecords

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return a map of SR references to SR records for all SRs known to the system.

Signature:

```
(SR ref -> SR record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(SR ref -> SR record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given SR.

Signature:

```
storage_operations set get_allowed_operations (session ref session_ref, SR
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `storage_operations set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getblobs

Overview:

Get the blobs field of the given SR.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getbynamelabel

Overview:

Get all the SR instances with the given label.

Signature:

```
SR ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `SR ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the SR instance with the specified UUID.

Signature:

```
SR ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `SR ref`

reference to the object

RPC name: getclustered

Overview:

Get the clustered field of the given SR.

Signature:

```
bool get_clustered (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getcontenttype

Overview:

Get the contenttype field of the given SR.

Signature:

```
string get_content_type (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given SR.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> storage_operations) map get_current_operations (session ref
session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> storage_operations) map`

value of the field

RPC name: getdatasources

Overview:

Signature:

```
data_source record set get_data_sources (session ref session_ref, SR ref
sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to interrogate

Minimum Role: read-only

Return Type: `data_source record set`

A set of data sources

RPC name: getintroducedby

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the introducedby field of the given SR.

Signature:

```
DR_task ref get_introduced_by (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `DR_task ref`

value of the field

RPC name: getistoolssr

Overview:

Get the istoolssr field of the given SR.

Signature:

```
bool get_is_tools_sr (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getlocalcacheenabled

Overview:

Get the localcacheenabled field of the given SR.

Signature:

```
bool get_local_cache_enabled (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given SR.

Signature:

```
string get_name_description (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given SR.

Signature:

```
string get_name_label (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given SR.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, SR ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPBDs

Overview:

Get the PBDs field of the given SR.

Signature:

```
PBD ref set get_PBDs (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PBD ref set`

value of the field

RPC name: getphysicalsize

Overview:

Get the physicalsize field of the given SR.

Signature:

```
int get_physical_size (session ref session_ref, SR ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getphysicalutilisation

Overview:

Get the physicalutilisation field of the given SR.

Signature:

```
int get_physical_utilisation (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given SR.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR record get_record (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR record`

all fields from the object

RPC name: getshared

Overview:

Get the shared field of the given SR.

Signature:

```
bool get_shared (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsmconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the smconfig field of the given SR.

Signature:

```
(string -> string) map get_sm_config (session ref session_ref, SR ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsupportedtypes

Overview:

Return a set of all the SR types supported by the system

Signature:

```
string set get_supported_types (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `string set`

the supported SR types

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettags

Overview:

Get the tags field of the given SR.

Signature:

```
string set get_tags (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettype

Overview:

Get the type field of the given SR.

Signature:

```
string get_type (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given SR.

Signature:

```
string get_uuid (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDIs

Overview:

Get the VDIs field of the given SR.

Signature:

```
VDI ref set get_VDIs (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref set`

value of the field

RPC name: getvirtualallocation

Overview:

Get the virtualallocation field of the given SR.

Signature:

```
int get_virtual_allocation (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: introduce

Overview:

Introduce a new Storage Repository into the managed system

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR ref introduce (session ref session_ref, string uuid, string name_label,
string name_description, string type, string content_type, bool shared,
(string -> string) map sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	The uuid assigned to the introduced SR
`string`	`name_label`	The name of the new storage repository
`string`	`name_description`	The description of the new storage repository
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`string`	`content_type`	The type of the new SRs content, if required (for example, ISOs)
`bool`	`shared`	True if the SR (is capable of) being shared by multiple hosts
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `SR ref`

The reference of the newly introduced Storage Repository.

RPC name: make

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Create a new Storage Repository on disk. This call is deprecated: use SR.create instead.

Signature:

```
string make (session ref session_ref, host ref host, (string -> string)
map device_config, int physical_size, string name_label, string
name_description, string type, string content_type, (string -> string) map
sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`int`	`physical_size`	The physical size of the new storage repository
`string`	`name_label`	The name of the new storage repository
`string`	`name_description`	The description of the new storage repository
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`string`	`content_type`	The type of the new SRs content, if required (for example, ISOs)
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `string`

The uuid of the newly created Storage Repository.

RPC name: probe

Overview:

Perform a backendspecific scan, using the given deviceconfig. If the deviceconfig is complete, then this will return a list of the SRs present of this type on the device, if any. If the deviceconfig is partial, then a backendspecific scan will be performed, returning results that will guide the user in improving the deviceconfig.

Signature:

```
string probe (session ref session_ref, host ref host, (string -> string)
map device_config, string type, (string -> string) map sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `string`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

An XML fragment containing the scan results. These are specific to the scan being performed, and the backend.

RPC name: probeext

Overview:

Perform a backendspecific scan, using the given deviceconfig. If the deviceconfig is complete, then this will return a list of the SRs present of this type on the device, if any. If the deviceconfig is partial, then a backendspecific scan will be performed, returning results that will guide the user in improving the deviceconfig.

Signature:

```
probe_result record set probe_ext (session ref session_ref, host ref host,
(string -> string) map device_config, string type, (string -> string) map
sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `probe_result record set`

A set of records containing the scan results.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: querydatasource

Overview:

Query the latest value of the specified data source

Signature:

```
float query_data_source (session ref session_ref, SR ref sr, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`data_source`	The data source to query

Minimum Role: read-only

Return Type: `float`

The latest value, averaged over the last 5 seconds

RPC name: recorddatasource

Overview:

Start recording the specified data source

Signature:

```
void record_data_source (session ref session_ref, SR ref sr, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`sr`	The SR
`string`	`data_source`	The data source to record

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given SR. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, SR ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromsmconfig

Overview:

Remove the given key and its corresponding value from the smconfig field of the given SR. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_sm_config (session ref session_ref, SR ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given SR. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, SR ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: scan

Overview:

Refreshes the list of VDIs associated with an SR

Signature:

```
void scan (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to scan

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name description of the SR

Signature:

```
void set_name_description (session ref session_ref, SR ref sr, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`value`	The name description for the SR

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name label of the SR

Signature:

```
void set_name_label (session ref session_ref, SR ref sr, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`value`	The name label for the SR

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given SR.

Signature:

```
void set_other_config (session ref session_ref, SR ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setphysicalsize

Overview:

Sets the SR's physicalsize field

Signature:

```
void set_physical_size (session ref session_ref, SR ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	The SR to modify
`int`	`value`	The new value of the SR's physical_size

Minimum Role: pool-operator

Return Type: `void`

RPC name: setshared

Overview:

Sets the shared flag on the SR

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_shared (session ref session_ref, SR ref sr, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`bool`	`value`	True if the SR is shared

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsmconfig

Overview:

Set the smconfig field of the given SR.

Signature:

```
void set_sm_config (session ref session_ref, SR ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: settags

Overview:

Set the tags field of the given SR.

Signature:

```
void set_tags (session ref session_ref, SR ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: update

Overview:

Refresh the fields on the SR object

Signature:

```
void update (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR whose fields should be refreshed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

Class: srstat

A set of highlevel properties associated with an SR.

Fields for class: srstat

FIELD	TYPE	QUALIFIER	DESCRIPTION
`clustered`	`bool`	*RO/runtime*	Indicates whether the SR uses clustered local storage.
`free_space`	`int`	*RO/runtime*	Number of bytes free on the backing storage (in bytes)
`health`	`sr_health`	*RO/runtime*	The health status of the SR.
`name_description`	`string`	*RO/runtime*	Longer, human-readable description of the SR. Descriptions are generally only displayed by clients when the user is examining SRs in detail.
`name_label`	`string`	*RO/runtime*	Short, human-readable label for the SR.
`total_space`	`int`	*RO/runtime*	Total physical size of the backing storage (in bytes)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string option`	*RO/runtime*	Uuid that uniquely identifies this SR, if one is available.

RPCs associated with class: srstat

Class srstat has no RPCs associated with it.

Class: subject

A user or group that can log in xapi

Fields for class: subject

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RO/constructor*	additional configuration
`roles`	`role ref set`	*RO/runtime*	the roles associated with this subject
`subject_identifier`	`string`	*RO/constructor*	the subject identifier, unique in the external directory service
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: subject

RPC name: addtoroles

Overview:

This call adds a new role to a subject

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_roles (session ref session_ref, subject ref self, role ref
role)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	The subject who we want to add the role to
`role ref`	`role`	The unique role reference

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

Overview:

Create a new subject instance, and return its handle.

Signature:

```
subject ref create (session ref session_ref, subject record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject record`	`args`	All constructor arguments

Minimum Role: pool-admin

Return Type: `subject ref`

reference to the newly created object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Destroy the specified subject instance.

Signature:

```
void destroy (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the subjects known to the system.

Signature:

```
subject ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `subject ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of subject references to subject records for all subjects known to the system.

Signature:

```
(subject ref -> subject record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(subject ref -> subject record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the subject instance with the specified UUID.

Signature:

```
subject ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `subject ref`

reference to the object

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given subject.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpermissionsnamelabel

Overview:

This call returns a list of permission names given a subject

Signature:

```
string set get_permissions_name_label (session ref session_ref, subject ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	The subject whose permissions will be retrieved

Minimum Role: read-only

Return Type: `string set`

a list of permission names

RPC name: getrecord

Overview:

Get a record containing the current state of the given subject.

Signature:

```
subject record get_record (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `subject record`

all fields from the object

RPC name: getroles

Overview:

Get the roles field of the given subject.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
role ref set get_roles (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `role ref set`

value of the field

RPC name: getsubjectidentifier

Overview:

Get the subjectidentifier field of the given subject.

Signature:

```
string get_subject_identifier (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the uuid field of the given subject.

Signature:

```
string get_uuid (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromroles

Overview:

This call removes a role from a subject

Signature:

```
void remove_from_roles (session ref session_ref, subject ref self, role
ref role)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	The subject from whom we want to remove the role
`role ref`	`role`	The unique role reference in the subject's roles field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

Class: task

A longrunning asynchronous task

Fields for class: task

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`task_allowed_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`backtrace`	`string`	*RO/runtime*	Function call trace for debugging.
`created`	`datetime`	*RO/runtime*	Time task was created
`current_operations`	`(string -> task_allowed_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`error_info`	`string set`	*RO/runtime*	if the task has failed, this field contains the set of associated error strings. Undefined otherwise.
`finished`	`datetime`	*RO/runtime*	Time task finished (i.e. succeeded or

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			failed). If task-status is pending, then the value of this field has no meaning
`name_description`	`string`	*RO/runtime*	a notes field containing human-readable description
`name_label`	`string`	*RO/runtime*	a human-readable name
`other_config`	`(string -> string) map`	*RO/runtime*	additional configuration
`progress`	`float`	*RO/runtime*	This field contains the estimated fraction of the task which is complete. This field should not be used to determine whether the task is complete - for this the status field of the task should be used.
`resident_on`	`host ref`	*RO/runtime*	the host on which the task is running
`result`	`string`	*RO/runtime*	if the task has completed successfully, this field contains the result value (either Void or an object reference). Undefined otherwise.
`status`	`task_status_type`	*RO/runtime*	current status of the task

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`subtask_of`	`task ref`	*RO/runtime*	Ref pointing to the task this is a substask of.
`subtasks`	`task ref set`	*RO/runtime*	List pointing to all the substasks.
`type`	`string`	*RO/runtime*	if the task has completed successfully, this field contains the type of the encoded result (i.e. name of the class whose reference is in the result field). Undefined otherwise.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: task

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given task.

Signature:

```
void add_to_other_config (session ref session_ref, task ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Task object to modify

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: read-only

Return Type: `void`

RPC name: cancel

Overview:

Request that a task be cancelled. Note that a task may fail to be cancelled and may complete or fail normally and note that, even when a task does cancel, it might take an arbitrary amount of time.

Signature:

```
void cancel (session ref session_ref, task ref task)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`task`	The task

Minimum Role: read-only

Return Type: `void`

Possible Error Codes: `OPERATION_NOT_ALLOWED`

RPC name: create

Overview:

Create a new task object which must be manually destroyed.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
task ref create (session ref session_ref, string label, string
description)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	short label for the new task
`string`	`description`	longer description for the new task

Minimum Role: read-only

Return Type: `task ref`

The reference of the created task object

RPC name: destroy

Overview:

Destroy the task object

Signature:

```
void destroy (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object

Minimum Role: read-only

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getall

Overview:

Return a list of all the tasks known to the system.

Signature:

```
task ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `task ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of task references to task records for all tasks known to the system.

Signature:

```
(task ref -> task record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(task ref -> task record) map`

records of all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given task.

Signature:

```
task_allowed_operations set get_allowed_operations (session ref
session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task_allowed_operations set`

value of the field

RPC name: getbacktrace

Overview:

Get the backtrace field of the given task.

Signature:

```
string get_backtrace (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbynamelabel

Overview:

Get all the task instances with the given label.

Signature:

```
task ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `task ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the task instance with the specified UUID.

Signature:

```
task ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `task ref`

reference to the object

RPC name: getcreated

Overview:

Get the created field of the given task.

Signature:

```
datetime get_created (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given task.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> task_allowed_operations) map get_current_operations (session
ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> task_allowed_operations) map`

value of the field

RPC name: geterrorinfo

Overview:

Get the errorinfo field of the given task.

Signature:

```
string set get_error_info (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getfinished

Overview:

Get the finished field of the given task.

Signature:

```
datetime get_finished (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given task.

Signature:

```
string get_name_description (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given task.

Signature:

```
string get_name_label (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given task.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, task ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprogress

Overview:

Get the progress field of the given task.

Signature:

```
float get_progress (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given task.

Signature:

```
task record get_record (session ref session_ref, task ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task record`

all fields from the object

RPC name: getresidenton

Overview:

Get the residenton field of the given task.

Signature:

```
host ref get_resident_on (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getresult

Overview:

Get the result field of the given task.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_result (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getstatus

Overview:

Get the status field of the given task.

Signature:

```
task_status_type get_status (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task_status_type`

value of the field

RPC name: getsubtaskof

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the subtaskof field of the given task.

Signature:

```
task ref get_subtask_of (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task ref`

value of the field

RPC name: getsubtasks

Overview:

Get the subtasks field of the given task.

Signature:

```
task ref set get_subtasks (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettype

Overview:

Get the type field of the given task.

Signature:

```
string get_type (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given task.

Signature:

```
string get_uuid (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given task. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, task ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Task object to modify
`string`	`key`	Key of entry to remove

Minimum Role: read-only

Return Type: `void`

RPC name: seterrorinfo

Overview:

Set the task error info

Signature:

```
void set_error_info (session ref session_ref, task ref self, string set
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`string set`	`value`	Task error info to be set

Minimum Role: read-only

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given task.

Signature:

```
void set_other_config (session ref session_ref, task ref self, (string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Task object to modify
`(string -> string) map`	`value`	New value to set

Minimum Role: read-only

Return Type: `void`

RPC name: setprogress

Overview:

Set the task progress

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_progress (session ref session_ref, task ref self, float value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`float`	`value`	Task progress value to be set

Minimum Role: read-only

Return Type: `void`

RPC name: setresult

Overview:

Set the task result

Signature:

```
void set_result (session ref session_ref, task ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`string`	`value`	Task result to be set

Minimum Role: read-only

Return Type: `void`

RPC name: setstatus

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the task status

Signature:

```
void set_status (session ref session_ref, task ref self, task_status_type
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`task_status_type`	`value`	task status value to be set

Minimum Role: read-only

Return Type: `void`

Class: tunnel

A tunnel for network traffic

Fields for class: tunnel

FIELD	TYPE	QUALIFIER	DESCRIPTION
`access_PIF`	`PIF ref`	*RO/constructor*	The interface through which the tunnel is accessed
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`protocol`	`tunnel_protocol`	*RW*	The protocol used for tunneling (either GRE or VxLAN)
`status`	`(string -> string) map`	*RW*	Status information about the tunnel

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`transport_PIF`	`PIF ref`	*RO/constructor*	The interface used by the tunnel
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: tunnel

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given tunnel.

Signature:

```
void add_to_other_config (session ref session_ref, tunnel ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtostatus

Overview:

Add the given keyvalue pair to the status field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_status (session ref session_ref, tunnel ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a tunnel

Signature:

```
tunnel ref create (session ref session_ref, PIF ref transport_PIF, network
ref network, tunnel_protocol protocol)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`transport_PIF`	PIF which receives the tagged traffic
`network ref`	`network`	Network to receive the tunnelled traffic

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`tunnel_protocol`	`protocol`	Protocol used for the tunnel (GRE or VxLAN)

Minimum Role: pool-operator

Return Type: `tunnel ref`

The reference of the created tunnel object

Possible Error Codes: `OPENVSWITCH_NOT_ACTIVE`, `TRANSPORT_PIF_NOT_CONFIGURED`, `IS_TUNNEL_ACCESS_PIF`

RPC name: destroy

Overview:

Destroy a tunnel

Signature:

```
void destroy (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	tunnel to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getaccessPIF

Overview:

Get the accessPIF field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PIF ref get_access_PIF (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getall

Overview:

Return a list of all the tunnels known to the system.

Signature:

```
tunnel ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `tunnel ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of tunnel references to tunnel records for all tunnels known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(tunnel ref -> tunnel record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(tunnel ref -> tunnel record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the tunnel instance with the specified UUID.

Signature:

```
tunnel ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `tunnel ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given tunnel.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprotocol

Overview:

Get the protocol field of the given tunnel.

Signature:

```
tunnel_protocol get_protocol (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `tunnel_protocol`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given tunnel.

Signature:

```
tunnel record get_record (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tunnel record`

all fields from the object

RPC name: getstatus

Overview:

Get the status field of the given tunnel.

Signature:

```
(string -> string) map get_status (session ref session_ref, tunnel ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gettransportPIF

Overview:

Get the transportPIF field of the given tunnel.

Signature:

```
PIF ref get_transport_PIF (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given tunnel. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, tunnel ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removefromstatus

Overview:

Remove the given key and its corresponding value from the status field of the given tunnel. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_status (session ref session_ref, tunnel ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given tunnel.

Signature:

```
void set_other_config (session ref session_ref, tunnel ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`tunnel ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setprotocol

Overview:

Set the protocol field of the given tunnel.

Signature:

```
void set_protocol (session ref session_ref, tunnel ref self,
tunnel_protocol value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`tunnel_protocol`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setstatus

Overview:

Set the status field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_status (session ref session_ref, tunnel ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: USBgroup

A group of compatible USBs across the resource pool

Fields for class: USBgroup

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PUSBs`	`PUSB ref set`	*RO/runtime*	List of PUSBs in the group
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VUSBs`	`VUSB ref set`	*RO/runtime*	List of VUSBs using the group

RPCs associated with class: USBgroup

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given USBgroup.

Signature:

```
void add_to_other_config (session ref session_ref, USB_group ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

Overview:

Signature:

```
USB_group ref create (session ref session_ref, string name_label, string
name_description, (string -> string) map other_config)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	
`string`	`name_description`	
`(string -> string) map`	`other_config`	

Minimum Role: pool-admin

Return Type: `USB_group ref`

The reference of the created USBgroup

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	The USB group to destroy

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the USBgroups known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
USB_group ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `USB_group ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of USBgroup references to USBgroup records for all USBgroups known to the system.

Signature:

```
(USB_group ref -> USB_group record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(USB_group ref -> USB_group record) map`

records of all objects

RPC name: getbynamelabel

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get all the USBgroup instances with the given label.

Signature:

```
USB_group ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `USB_group ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the USBgroup instance with the specified UUID.

Signature:

```
USB_group ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `USB_group ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the object

RPC name: getnamedescription

Overview:

Get the name/description field of the given USBgroup.

Signature:

```
string get_name_description (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given USBgroup.

Signature:

```
string get_name_label (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given USBgroup.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPUSBs

Overview:

Get the PUSBs field of the given USBgroup.

Signature:

```
PUSB ref set get_PUSBs (session ref session_ref, USB_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PUSB ref set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given USBgroup.

Signature:

```
USB_group record get_record (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `USB_group record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given USBgroup.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVUSBs

Overview:

Get the VUSBs field of the given USBgroup.

Signature:

```
VUSB ref set get_VUSBs (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VUSB ref set`

value of the field

RPC name: removefromotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given USBgroup. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, USB_group ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given USBgroup.

Signature:

```
void set_name_description (session ref session_ref, USB_group ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given USBgroup.

Signature:

```
void set_name_label (session ref session_ref, USB_group ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given USBgroup.

Signature:

```
void set_other_config (session ref session_ref, USB_group ref self,
(string -> string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

Class: user

This class is deprecated.

A user of the system

Fields for class: user

FIELD	TYPE	QUALIFIER	DESCRIPTION
`fullname`	`string`	*RW*	**Deprecated.** full name
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`short_name`	`string`	*RO/constructor*	**Deprecated.** short name (for example, userid)
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference

RPCs associated with class: user

RPC name: addtootherconfig

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Add the given keyvalue pair to the otherconfig field of the given user.

Signature:

```
void add_to_other_config (session ref session_ref, user ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

This message is deprecated.

Overview:

Create a new user instance, and return its handle.

Signature:

```
user ref create (session ref session_ref, user record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user record`	`args`	All constructor arguments

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `user ref`

reference to the newly created object

RPC name: destroy

This message is deprecated.

Overview:

Destroy the specified user instance.

Signature:

```
void destroy (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the user instance with the specified UUID.

Signature:

```
user ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `user ref`

reference to the object

RPC name: getfullname

This message is deprecated.

Overview:

Get the fullname field of the given user.

Signature:

```
string get_fullname (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given user.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, user ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given user.

Signature:

```
user record get_record (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `user record`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

all fields from the object

RPC name: getshortname

This message is deprecated.

Overview:

Get the shortname field of the given user.

Signature:

```
string get_short_name (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given user.

Signature:

```
string get_uuid (session ref session_ref, user ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given user. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, user ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setfullname

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the fullname field of the given user.

Signature:

```
void set_fullname (session ref session_ref, user ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given user.

Signature:

```
void set_other_config (session ref session_ref, user ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

Class: VBD

A virtual block device

Fields for class: VBD

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vbd_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`bootable`	`bool`	*RW*	true if this VBD is bootable
`current_operations`	`(string -> vbd_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`currently_attached`	`bool`	*RO/constructor*	is the device currently attached (erased on reboot)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`device`	`string`	*RO/constructor*	device seen by the guest, for example, hda1
`empty`	`bool`	*RO/constructor*	if true this represents an empty drive
`metrics`	`VBD_metrics ref`	*RO/runtime*	**Removed.** metrics associated with this VBD
`mode`	`vbd_mode`	*RO/constructor*	the mode the VBD should be mounted with
`other_config`	`(string -> string) map`	*RW*	additional configuration
`qos_algorithm_params`	`(string -> string) map`	*RW*	parameters for chosen QoS algorithm
`qos_algorithm_type`	`string`	*RW*	QoS algorithm to use
`qos_supported_algorithms`	`string set`	*RO/runtime*	supported QoS algorithms for this VBD
`runtime_properties`	`(string -> string) map`	*RO/runtime*	Device runtime properties
`status_code`	`int`	*RO/runtime*	error/success code associated with last attach-operation (erased on reboot)
`status_detail`	`string`	*RO/runtime*	error/success information associated with last attach-operation

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			status (erased on reboot)
`storage_lock`	`bool`	*RO/runtime*	true if a storage level lock was acquired
`type`	`vbd_type`	*RW*	how the VBD will appear to the guest (for example, disk or CD)
`unpluggable`	`bool`	*RW*	true if this VBD will support hot-unplug
`userdevice`	`string`	*RW*	user-friendly device name, for example, 0, 1, 2, etc.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VDI`	`VDI ref`	*RO/constructor*	the virtual disk
`VM`	`VM ref`	*RO/constructor*	the virtual machine

RPCs associated with class: VBD

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VBD.

Signature:

```
void add_to_other_config (session ref session_ref, VBD ref self, string key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoqosalgorithmparams

Overview:

Add the given keyvalue pair to the qos/algorithmparams field of the given VBD.

Signature:

```
void add_to_qos_algorithm_params (session ref session_ref, VBD ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: assertattachable

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Throws an error if this VBD could not be attached to this VM if the VM were running. Intended for debugging.

Signature:

```
void assert_attachable (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to query

Minimum Role: vm-admin

Return Type: `void`

RPC name: create

Overview:

Create a new VBD instance, and return its handle.

Signature:

```
VBD ref create (session ref session_ref, VBD record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VBD ref`

reference to the newly created object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Destroy the specified VBD instance.

Signature:

```
void destroy (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: eject

Overview:

Remove the media from the device and leave it empty

Signature:

```
void eject (session ref session_ref, VBD ref vbd)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`vbd`	The vbd representing the CDROM-like device

Minimum Role: vm-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `VBD_NOT_REMOVABLE_MEDIA`, `VBD_IS_EMPTY`

RPC name: getall

Overview:

Return a list of all the VBDs known to the system.

Signature:

```
VBD ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VBD ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VBD references to VBD records for all VBDs known to the system.

Signature:

```
(VBD ref -> VBD record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VBD ref -> VBD record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VBD.

Signature:

```
vbd_operations set get_allowed_operations (session ref session_ref, VBD
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vbd_operations set`

value of the field

RPC name: getbootable

Overview:

Get the bootable field of the given VBD.

Signature:

```
bool get_bootable (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VBD instance with the specified UUID.

Signature:

```
VBD ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VBD ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> vbd_operations) map get_current_operations (session ref
session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vbd_operations) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VBD.

Signature:

```
bool get_currently_attached (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getdevice

Overview:

Get the device field of the given VBD.

Signature:

```
string get_device (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getempty

Overview:

Get the empty field of the given VBD.

Signature:

```
bool get_empty (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getmetrics

This message is removed.

Overview:

Get the metrics field of the given VBD.

Signature:

```
VBD_metrics ref get_metrics (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VBD_metrics ref`

value of the field

RPC name: getmode

Overview:

Get the mode field of the given VBD.

Signature:

```
vbd_mode get_mode (session ref session_ref, VBD ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vbd_mode`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VBD.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmparams

Overview:

Get the qos/algorithmparams field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_qos_algorithm_params (session ref session_ref,
VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmtype

Overview:

Get the qos/algorithmtype field of the given VBD.

Signature:

```
string get_qos_algorithm_type (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getqossupportedalgorithms

Overview:

Get the qos/supportedalgorithms field of the given VBD.

Signature:

```
string set get_qos_supported_algorithms (session ref session_ref, VBD ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VBD.

Signature:

```
VBD record get_record (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VBD record`

all fields from the object

RPC name: getruntimeproperties

Overview:

Get the runtimeproperties field of the given VBD.

Signature:

```
(string -> string) map get_runtime_properties (session ref session_ref,
VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getstatuscode

Overview:

Get the statuscode field of the given VBD.

Signature:

```
int get_status_code (session ref session_ref, VBD ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getstatusdetail

Overview:

Get the statusdetail field of the given VBD.

Signature:

```
string get_status_detail (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getstoragelock

Overview:

Get the storagelock field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_storage_lock (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VBD.

Signature:

```
vbd_type get_type (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vbd_type`

value of the field

RPC name: getunpluggable

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the unpluggable field of the given VBD.

Signature:

```
bool get_unpluggable (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getuserdevice

Overview:

Get the userdevice field of the given VBD.

Signature:

```
string get_userdevice (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given VBD.

Signature:

```
string get_uuid (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDI

Overview:

Get the VDI field of the given VBD.

Signature:

```
VDI ref get_VDI (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VDI ref`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VBD.

Signature:

```
VM ref get_VM (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: insert

Overview:

Insert new media into the device

Signature:

```
void insert (session ref session_ref, VBD ref vbd, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VBD ref`	`vbd`	The vbd representing the CDROM-like device
`VDI ref`	`vdi`	The new VDI to 'insert'

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VBD_NOT_REMOVABLE_MEDIA`, `VBD_NOT_EMPTY`

RPC name: plug

Overview:

Hotplug the specified VBD, dynamically attaching it to the running VM

Signature:

```
void plug (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to hotplug

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VBD. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_other_config (session ref session_ref, VBD ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromqosalgorithmparams

Overview:

Remove the given key and its corresponding value from the qos/algorithmparams field of the given VBD. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_qos_algorithm_params (session ref session_ref, VBD ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setbootable

Overview:

Set the bootable field of the given VBD.

Signature:

```
void set_bootable (session ref session_ref, VBD ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setmode

Overview:

Sets the mode of the VBD. The powerstate of the VM must be halted.

Signature:

```
void set_mode (session ref session_ref, VBD ref self, vbd_mode value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	Reference to the object
`vbd_mode`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VBD.

Signature:

```
void set_other_config (session ref session_ref, VBD ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmparams

Overview:

Set the qos/algorithmparams field of the given VBD.

Signature:

```
void set_qos_algorithm_params (session ref session_ref, VBD ref self,
(string -> string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmtype

Overview:

Set the qos/algorithmtype field of the given VBD.

Signature:

```
void set_qos_algorithm_type (session ref session_ref, VBD ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: settype

Overview:

Set the type field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_type (session ref session_ref, VBD ref self, vbd_type value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`vbd_type`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setunpluggable

Overview:

Set the unpluggable field of the given VBD.

Signature:

```
void set_unpluggable (session ref session_ref, VBD ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setuserdevice

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the userdevice field of the given VBD.

Signature:

```
void set_userdevice (session ref session_ref, VBD ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: unplug

Overview:

Hotunplug the specified VBD, dynamically unattaching it from the running VM

Signature:

```
void unplug (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to hot-unplug

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `DEVICE_DETACH_REJECTED`, `DEVICE_ALREADY_DETACHED`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: unplugforce

Overview:

Forcibly unplug the specified VBD

Signature:

```
void unplug_force (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to forcibly unplug

Minimum Role: vm-admin

Return Type: `void`

Class: VBDmetrics

This class is removed.

The metrics associated with a virtual block device

Fields for class: VBDmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`io_read_kbs`	`float`	*RO/runtime*	**Removed.** Read bandwidth (KiB/s)
`io_write_kbs`	`float`	*RO/runtime*	**Removed.** Write bandwidth (KiB/s)
`last_updated`	`datetime`	*RO/runtime*	**Removed.** Time at which this information was last updated

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	**Removed.** additional configuration
`uuid`	`string`	*RO/runtime*	**Removed.** Unique identifier/object reference

RPCs associated with class: VBDmetrics

RPC name: addtootherconfig

This message is removed.

Overview:

Add the given keyvalue pair to the otherconfig field of the given VBDmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VBD_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Return a list of all the VBDmetrics instances known to the system.

Signature:

```
VBD_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VBD_metrics ref set`

references to all objects

RPC name: getallrecords

This message is removed.

Overview:

Return a map of VBDmetrics references to VBDmetrics records for all VBDmetrics instances known to the system.

Signature:

```
(VBD_metrics ref -> VBD_metrics record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VBD_metrics ref -> VBD_metrics record) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

records of all objects

RPC name: getbyuuid

This message is removed.

Overview:

Get a reference to the VBDmetrics instance with the specified UUID.

Signature:

```
VBD_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VBD_metrics ref`

reference to the object

RPC name: getioreadkbs

This message is removed.

Overview:

Get the io/readkbs field of the given VBDmetrics.

Signature:

```
float get_io_read_kbs (session ref session_ref, VBD_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getiowritekbs

This message is removed.

Overview:

Get the io/writekbs field of the given VBDmetrics.

Signature:

```
float get_io_write_kbs (session ref session_ref, VBD_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getlastupdated

This message is removed.

Overview:

Get the lastupdated field of the given VBDmetrics.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
datetime get_last_updated (session ref session_ref, VBD_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getotherconfig

This message is removed.

Overview:

Get the otherconfig field of the given VBDmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VBD_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getrecord

This message is removed.

Overview:

Get a record containing the current state of the given VBDmetrics.

Signature:

```
VBD_metrics record get_record (session ref session_ref, VBD_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VBD_metrics record`

all fields from the object

RPC name: getuuid

This message is removed.

Overview:

Get the uuid field of the given VBDmetrics.

Signature:

```
string get_uuid (session ref session_ref, VBD_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is removed.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VBDmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VBD_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the otherconfig field of the given VBDmetrics.

Signature:

```
void set_other_config (session ref session_ref, VBD_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VDI

A virtual disk image

Fields for class: VDI

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allow_caching`	`bool`	*RO/runtime*	true if this VDI is to be cached in the local cache SR
`allowed_operations`	`vdi_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`cbt_enabled`	`bool`	*RO/runtime*	True if changed blocks are tracked for this VDI
`crash_dumps`	`crashdump ref set`	*RO/runtime*	list of crash dumps that refer to this disk
`current_operations`	`(string -> vdi_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`is_a_snapshot`	`bool`	*RO/runtime*	true if this is a snapshot.
`is_tools_iso`	`bool`	*RO/runtime*	Whether this VDI is a Tools ISO
`location`	`string`	*RO/runtime*	location information
`managed`	`bool`	*RO/runtime*	
`metadata_latest`	`bool`	*RO/runtime*	Whether this VDI contains the latest known accessible metadata for the pool
`metadata_of_pool`	`pool ref`	*RO/runtime*	The pool whose metadata is contained in this VDI
`missing`	`bool`	*RO/runtime*	true if SR scan operation reported this VDI as not present on disk

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`on_boot`	`on_boot`	*RO/runtime*	The behaviour of this VDI on a VM boot
`other_config`	`(string -> string) map`	*RW*	additional configuration
`parent`	`VDI ref`	*RO/runtime*	**Deprecated.** This field is always null. Deprecated
`physical_utilisation`	`int`	*RO/runtime*	amount of physical space that the disk image is currently taking up on the storage repository (in bytes)
`read_only`	`bool`	*RO/constructor*	true if this disk may ONLY be mounted read-only
`sharable`	`bool`	*RO/constructor*	true if this disk may be shared
`sm_config`	`(string -> string) map`	*RW*	SM dependent data
`snapshot_of`	`VDI ref`	*RO/runtime*	Ref pointing to the VDI this snapshot is of.
`snapshot_time`	`datetime`	*RO/runtime*	Date/time when this snapshot was created.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`snapshots`	`VDI ref set`	*RO/runtime*	List pointing to all the VDIs snapshots.
`SR`	`SR ref`	*RO/constructor*	storage repository in which the VDI resides
`storage_lock`	`bool`	*RO/runtime*	true if this disk is locked at the storage level
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`type`	`vdi_type`	*RO/constructor*	type of the VDI
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VBDs`	`VBD ref set`	*RO/runtime*	list of vbds that refer to this disk
`virtual_size`	`int`	*RO/constructor*	size of disk as presented to the guest (in bytes). Note that, depending on storage backend type, requested size may not be respected exactly
`xenstore_data`	`(string -> string) map`	*RW*	data to be inserted into the xenstore tree (/local/domain/0/ backend/vbd/ <domid>/<device-id>/sm-data) after the VDI is attached. This is generally set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			by the SM backends on vdi_attach.

RPCs associated with class: VDI

RPC name: addtags

Overview:

Add the given value to the tags field of the given VDI. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, VDI ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VDI.

Signature:

```
void add_to_other_config (session ref session_ref, VDI ref self, string
key, string value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtosmconfig

Overview:

Add the given keyvalue pair to the smconfig field of the given VDI.

Signature:

```
void add_to_sm_config (session ref session_ref, VDI ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: addtoxenstoredata

Overview:

Add the given keyvalue pair to the xenstoredata field of the given VDI.

Signature:

```
void add_to_xenstore_data (session ref session_ref, VDI ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: clone

Overview:

Take an exact copy of the VDI and return a reference to the new disk. If any driverparams are specified then these are passed through to the storagespecific substrate driver that implements the clone operation. NB the clone lives in the same Storage Repository as its parent.

Signature:

```
VDI ref clone (session ref session_ref, VDI ref vdi, (string -> string)
map driver_params)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to clone
`(string -> string) map`	`driver_params`	Optional parameters that are passed through to the backend driver in order to specify storage-type-specific clone options

Minimum Role: vm-admin

Return Type: `VDI ref`

The ID of the newly created VDI.

RPC name: copy

Overview:

Copy either a full VDI or the block differences between two VDIs into either a fresh VDI or an existing VDI.

Signature:

```
VDI ref copy (session ref session_ref, VDI ref vdi, SR ref sr, VDI ref
base_vdi, VDI ref into_vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to copy
`SR ref`	`sr`	The destination SR (only required if the destination VDI is not specified

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`base_vdi`	The base VDI (only required if copying only changed blocks, by default all blocks will be copied)
`VDI ref`	`into_vdi`	The destination VDI to copy blocks into (if omitted then a destination SR must be provided and a fresh VDI will be created)

Minimum Role: vm-admin

Return Type: `VDI ref`

The reference of the VDI where the blocks were written.

Possible Error Codes: `VDI_READONLY`, `VDI_TOO_SMALL`, `VDI_NOT_SPARSE`

RPC name: create

Overview:

Create a new VDI instance, and return its handle.

Signature:

```
VDI ref create (session ref session_ref, VDI record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VDI ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the newly created object

RPC name: datadestroy

Overview:

Delete the data of the snapshot VDI, but keep its changed block tracking metadata. When successful, this call changes the type of the VDI to cbtmetadata. This operation is idempotent: calling it on a VDI of type cbtmetadata results in a noop, and no error will be thrown.

Signature:

```
void data_destroy (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI whose data should be deleted.

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `OPERATION_NOT_ALLOWED`, `VDI_INCOMPATIBLE_TYPE`, `VDI_NO_CBT_METADATA`, `VDI_IN_USE`, `VDI_IS_A_PHYSICAL_DEVICE`

RPC name: destroy

Overview:

Destroy the specified VDI instance.

Signature:

```
void destroy (session ref session_ref, VDI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: disablecbt

Overview:

Disable changed block tracking for the VDI. This call is only allowed on VDIs that support enabling CBT. It is an idempotent operation disabling CBT for a VDI for which CBT is not enabled results in a noop, and no error will be thrown.

Signature:

```
void disable_cbt (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI for which CBT should be disabled

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `OPERATION_NOT_ALLOWED`, `VDI_INCOMPATIBLE_TYPE`, `VDI_ON_BOOT_MODE_INCOMPATIBLE_WITH_OPERATION`

RPC name: enablecbt

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable changed block tracking for the VDI. This call is idempotent enabling CBT for a VDI for which CBT is already enabled results in a noop, and no error will be thrown.

Signature:

```
void enable_cbt (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI for which CBT should be enabled

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `OPERATION_NOT_ALLOWED`, `VDI_INCOMPATIBLE_TYPE`, `VDI_ON_BOOT_MODE_INCOMPATIBLE_WITH_OPERATION`

RPC name: forget

Overview:

Removes a VDI record from the database

Signature:

```
void forget (session ref session_ref, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to forget about

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VDIs known to the system.

Signature:

```
VDI ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VDI ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VDI references to VDI records for all VDIs known to the system.

Signature:

```
(VDI ref -> VDI record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VDI ref -> VDI record) map`

records of all objects

RPC name: getallowcaching

Overview:

Get the allowcaching field of the given VDI.

Signature:

```
bool get_allow_caching (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VDI.

Signature:

```
vdi_operations set get_allowed_operations (session ref session_ref, VDI
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vdi_operations set`

value of the field

RPC name: getbynamelabel

Overview:

Get all the VDI instances with the given label.

Signature:

```
VDI ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VDI ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VDI instance with the specified UUID.

Signature:

```
VDI ref get_by_uuid (session ref session_ref, string uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VDI ref`

reference to the object

RPC name: getcbtenabled

Overview:

Get the cbtenabled field of the given VDI.

Signature:

```
bool get_cbt_enabled (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getcrashdumps

Overview:

Get the crashdumps field of the given VDI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
crashdump ref set get_crash_dumps (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `crashdump ref set`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VDI.

Signature:

```
(string -> vdi_operations) map get_current_operations (session ref
session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vdi_operations) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getisasnapshot

Overview:

Get the isasnapshot field of the given VDI.

Signature:

```
bool get_is_a_snapshot (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getistoolsiso

Overview:

Get the istoolsiso field of the given VDI.

Signature:

```
bool get_is_tools_iso (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getlocation

Overview:

Get the location field of the given VDI.

Signature:

```
string get_location (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getmanaged

Overview:

Get the managed field of the given VDI.

Signature:

```
bool get_managed (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetadatalatest

Overview:

Get the metadatalatest field of the given VDI.

Signature:

```
bool get_metadata_latest (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetadataofpool

Overview:

Get the metadataofpool field of the given VDI.

Signature:

```
pool ref get_metadata_of_pool (session ref session_ref, VDI ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool ref`

value of the field

RPC name: getmissing

Overview:

Get the missing field of the given VDI.

Signature:

```
bool get_missing (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given VDI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_name_description (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VDI.

Signature:

```
string get_name_label (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnbdinfo

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get details specifying how to access this VDI via a Network Block Device server. For each of a set of NBD server addresses on which the VDI is available, the return value set contains a vdinbdserverinfo object that contains an exportname to request once the NBD connection is established, and connection details for the address. An empty list is returned if there is no network that has a PIF on a host with access to the relevant SR, or if no such network has been assigned an NBDrelated purpose in its purpose field. To access the given VDI, any of the vdinbdserverinfo objects can be used to make a connection to a server, and then the VDI will be available by requesting the exportname.

Signature:

```
vdi_nbd_server_info record set get_nbd_info (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to access via Network Block Device protocol

Minimum Role: vm-admin

Return Type: `vdi_nbd_server_info record set`

The details necessary for connecting to the VDI over NBD. This includes an authentication token, so must be treated as sensitive material and must not be sent over insecure networks.

Possible Error Codes: `VDI_INCOMPATIBLE_TYPE`

RPC name: getonboot

Overview:

Get the onboot field of the given VDI.

Signature:

```
on_boot get_on_boot (session ref session_ref, VDI ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_boot`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VDI.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getparent

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the parent field of the given VDI.

Signature:

```
VDI ref get_parent (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getphysicalutilisation

Overview:

Get the physicalutilisation field of the given VDI.

Signature:

```
int get_physical_utilisation (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getreadonly

Overview:

Get the readonly field of the given VDI.

Signature:

```
bool get_read_only (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VDI.

Signature:

```
VDI record get_record (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VDI record`

all fields from the object

RPC name: getsharable

Overview:

Get the sharable field of the given VDI.

Signature:

```
bool get_sharable (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsmconfig

Overview:

Get the smconfig field of the given VDI.

Signature:

```
(string -> string) map get_sm_config (session ref session_ref, VDI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsnapshotof

Overview:

Get the snapshotof field of the given VDI.

Signature:

```
VDI ref get_snapshot_of (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getsnapshottime

Overview:

Get the snapshottime field of the given VDI.

Signature:

```
datetime get_snapshot_time (session ref session_ref, VDI ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getsnapshots

Overview:

Get the snapshots field of the given VDI.

Signature:

```
VDI ref set get_snapshots (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref set`

value of the field

RPC name: getSR

Overview:

Get the SR field of the given VDI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR ref get_SR (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getstoragelock

Overview:

Get the storagelock field of the given VDI.

Signature:

```
bool get_storage_lock (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gettags

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the tags field of the given VDI.

Signature:

```
string set get_tags (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VDI.

Signature:

```
vdi_type get_type (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vdi_type`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given VDI.

Signature:

```
string get_uuid (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVBDs

Overview:

Get the VBDs field of the given VDI.

Signature:

```
VBD ref set get_VBDs (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VBD ref set`

value of the field

RPC name: getvirtualsize

Overview:

Get the virtualsize field of the given VDI.

Signature:

```
int get_virtual_size (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getxenstoredata

Overview:

Get the xenstoredata field of the given VDI.

Signature:

```
(string -> string) map get_xenstore_data (session ref session_ref, VDI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: introduce

Overview:

Create a new VDI record in the database only

Signature:

```
VDI ref introduce (session ref session_ref, string uuid, string
name_label, string name_description, SR ref SR, vdi_type type, bool
sharable, bool read_only, (string -> string) map other_config, string
location, (string -> string) map xenstore_data, (string -> string) map
sm_config, bool managed, int virtual_size, int physical_utilisation, pool
ref metadata_of_pool, bool is_a_snapshot, datetime snapshot_time, VDI ref
snapshot_of)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	The uuid of the disk to introduce
`string`	`name_label`	The name of the disk record
`string`	`name_description`	The description of the disk record
`SR ref`	`SR`	The SR that the VDI is in
`vdi_type`	`type`	The type of the VDI

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`sharable`	true if this disk may be shared
`bool`	`read_only`	true if this disk may ONLY be mounted read-only
`(string -> string) map`	`other_config`	additional configuration
`string`	`location`	location information
`(string -> string) map`	`xenstore_data`	Data to insert into xenstore
`(string -> string) map`	`sm_config`	Storage-specific config
`bool`	`managed`	Storage-specific config
`int`	`virtual_size`	Storage-specific config
`int`	`physical_utilisation`	Storage-specific config
`pool ref`	`metadata_of_pool`	Storage-specific config
`bool`	`is_a_snapshot`	Storage-specific config
`datetime`	`snapshot_time`	Storage-specific config. When the timezone is missing, UTC is assumed
`VDI ref`	`snapshot_of`	Storage-specific config

Minimum Role: vm-admin

Return Type: `VDI ref`

The ref of the newly created VDI record.

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`

RPC name: listchangedblocks

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Compare two VDIs in 64k block increments and report which blocks differ. This operation is not allowed when vdito is attached to a VM.

Signature:

```
string list_changed_blocks (session ref session_ref, VDI ref vdi_from, VDI ref vdi_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi_from`	The first VDI.
`VDI ref`	`vdi_to`	The second VDI.

Minimum Role: vm-operator

Return Type: `string`

A base64 stringencoding of the bitmap showing which blocks differ in the two VDIs.

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `VDI_IN_USE`

RPC name: opendatabase

Overview:

Load the metadata found on the supplied VDI and return a session reference which can be used in API calls to query its contents.

Signature:

```
session ref open_database (session ref session_ref, VDI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI which contains the database to open

Minimum Role: pool-operator

Return Type: `session ref`

A session which can be used to query the database

RPC name: poolmigrate

Overview:

Migrate a VDI, which may be attached to a running guest, to a different SR. The destination SR must be visible to the guest.

Signature:

```
VDI ref pool_migrate (session ref session_ref, VDI ref vdi, SR ref sr,
(string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION		
`session ref`	`session_ref`	Reference to a valid session		
`VDI ref`	`vdi`	The VDI to migrate		
`SR ref`	`sr`	The destination SR		
`(string -> string) map`	`options`	Extra parameters. Supports: "dest-img-format" (raw	vhd	qcow2) to specify the image format to use on the destination SR.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-power-admin

Return Type: `VDI ref`

The new reference of the migrated VDI.

RPC name: readdatabasepooluuid

Overview:

Check the VDI cache for the pool UUID of the database on this VDI.

Signature:

```
string read_database_pool_uuid (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The metadata VDI to look up in the cache.

Minimum Role: read-only

Return Type: `string`

The cached pool UUID of the database on the VDI.

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VDI. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VDI ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromsmconfig

Overview:

Remove the given key and its corresponding value from the smconfig field of the given VDI. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_sm_config (session ref session_ref, VDI ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromxenstoredata

Overview:

Remove the given key and its corresponding value from the xenstoredata field of the given VDI. If the key is not in that Map, then do nothing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void remove_from_xenstore_data (session ref session_ref, VDI ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given VDI. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, VDI ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: resize

Overview:

Resize the VDI.

Signature:

```
void resize (session ref session_ref, VDI ref vdi, int size)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to resize
`int`	`size`	The new size of the VDI

Minimum Role: vm-admin

Return Type: `void`

RPC name: resizeonline

This message is removed.

Overview:

Resize the VDI which may or may not be attached to running guests.

Signature:

```
void resize_online (session ref session_ref, VDI ref vdi, int size)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`vdi`	The VDI to resize
`int`	`size`	The new size of the VDI

Minimum Role: vm-admin

Return Type: `void`

RPC name: revert

Overview:

Copy the contents of a snapshot to the VDI it's related to. The original contents of the VDI are lost.

Signature:

```
void revert (session ref session_ref, VDI ref snapshot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`snapshot`	The snapshot to which we want to revert

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `UNIMPLEMENTED_IN_SM_BACKEND`

RPC name: setallowcaching

Overview:

Set the value of the allowcaching parameter. This value can only be changed when the VDI is not attached to a running VM. The caching behaviour is only affected by this flag for VHDbased VDIs that have one parent and no child VHDs. Moreover, caching only takes place when the host running the VM containing this VDI has a nominated SR for local caching.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_allow_caching (session ref session_ref, VDI ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`bool`	`value`	The value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name description of the VDI. This can only happen when its SR is currently attached.

Signature:

```
void set_name_description (session ref session_ref, VDI ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`string`	`value`	The name description for the VDI

Minimum Role: vm-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name label of the VDI. This can only happen when then its SR is currently attached.

Signature:

```
void set_name_label (session ref session_ref, VDI ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`string`	`value`	The name lable for the VDI

Minimum Role: vm-admin

Return Type: `void`

RPC name: setonboot

Overview:

Set the value of the onboot parameter. This value can only be changed when the VDI is not attached to a running VM.

Signature:

```
void set_on_boot (session ref session_ref, VDI ref self, on_boot value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	The VDI to modify
`on_boot`	`value`	The value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VDI.

Signature:

```
void set_other_config (session ref session_ref, VDI ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setreadonly

Overview:

Sets the VDI's readonly field

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_read_only (session ref session_ref, VDI ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`bool`	`value`	The new value of the VDI's read_only field

Minimum Role: vm-admin

Return Type: `void`

RPC name: setsharable

Overview:

Sets the VDI's sharable field

Signature:

```
void set_sharable (session ref session_ref, VDI ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`bool`	`value`	The new value of the VDI's sharable field

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setsmconfig

Overview:

Set the smconfig field of the given VDI.

Signature:

```
void set_sm_config (session ref session_ref, VDI ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given VDI.

Signature:

```
void set_tags (session ref session_ref, VDI ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: setxenstoredata

Overview:

Set the xenstoredata field of the given VDI.

Signature:

```
void set_xenstore_data (session ref session_ref, VDI ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: snapshot

Overview:

Take a readonly snapshot of the VDI, returning a reference to the snapshot. If any driverparams are specified then these are passed through to the storagespecific substrate driver that takes the snapshot. NB the snapshot lives in the same Storage Repository as its parent.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VDI ref snapshot (session ref session_ref, VDI ref vdi, (string -> string)
map driver_params)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to snapshot
`(string -> string) map`	`driver_params`	Optional parameters that can be passed through to backend driver in order to specify storage-type-specific snapshot options

Minimum Role: vm-admin

Return Type: `VDI ref`

The ID of the newly created VDI.

RPC name: update

Overview:

Ask the storage backend to refresh the fields in the VDI object

Signature:

```
void update (session ref session_ref, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI whose stats (eg size) should be updated

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`

Class: vdinbdserverinfo

Details for connecting to a VDI using the Network Block Device protocol

Fields for class: vdinbdserverinfo

FIELD	TYPE	QUALIFIER	DESCRIPTION
`address`	`string`	*RO/runtime*	An address on which the server can be reached; this can be IPv4, IPv6, or a DNS name.
`cert`	`string`	*RO/runtime*	The TLS certificate of the server
`exportname`	`string`	*RO/runtime*	The exportname to request over NBD. This holds details including an authentication token, so it must be protected appropriately. Clients should regard the exportname as an opaque string or token.
`port`	`int`	*RO/runtime*	The TCP port
`subject`	`string`	*RO/runtime*	For convenience, this redundant field holds a DNS (hostname) subject of the certificate. This can be a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			wildcard, but only for a certificate that has a wildcard subject and no concrete hostname subjects.

RPCs associated with class: vdinbdserverinfo

Class vdinbdserverinfo has no RPCs associated with it.

Class: VGPU

A virtual GPU vGPU

Fields for class: VGPU

FIELD	TYPE	QUALIFIER	DESCRIPTION
`compatibility_metadata`	`(string -> string) map`	*RO/runtime*	VGPU metadata to determine whether a VGPU can migrate between two PGPUs
`currently_attached`	`bool`	*RO/runtime*	Reflects whether the virtual device is currently connected to a physical device
`device`	`string`	*RO/runtime*	Order in which the devices are plugged into the VM
`extra_args`	`string`	*RW*	Extra arguments for vGPU and passed to demu
`GPU_group`	`GPU_group ref`	*RO/runtime*	GPU group used by the vGPU
`other_config`	`(string -> string) map`	*RW*	Additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`PCI`	`PCI ref`	*RO/runtime*	Device passed trough to VM, either as full device or SR-IOV virtual function
`resident_on`	`PGPU ref`	*RO/runtime*	The PGPU on which this VGPU is running
`scheduled_to_be_resident_on`	`PGPU ref`	*RO/runtime*	The PGPU on which this VGPU is scheduled to run
`type`	`VGPU_type ref`	*RO/runtime*	Preset type for this VGPU
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VM`	`VM ref`	*RO/runtime*	VM that owns the vGPU

RPCs associated with class: VGPU

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VGPU.

Signature:

```
void add_to_other_config (session ref session_ref, VGPU ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Signature:

```
VGPU ref create (session ref session_ref, VM ref VM, GPU_group ref
GPU_group, string device, (string -> string) map other_config, VGPU_type
ref type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`VM`	
`GPU_group ref`	`GPU_group`	
`string`	`device`	
`(string -> string) map`	`other_config`	
`VGPU_type ref`	`type`	

Minimum Role: pool-operator

Return Type: `VGPU ref`

The reference of the created VGPU object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	The vGPU to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VGPUs known to the system.

Signature:

```
VGPU ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VGPU ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of VGPU references to VGPU records for all VGPUs known to the system.

Signature:

```
(VGPU ref -> VGPU record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VGPU ref -> VGPU record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VGPU instance with the specified UUID.

Signature:

```
VGPU ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VGPU ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the object

RPC name: getcompatibilitymetadata

Overview:

Get the compatibilitymetadata field of the given VGPU.

Signature:

```
(string -> string) map get_compatibility_metadata (session ref
session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VGPU.

Signature:

```
bool get_currently_attached (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdevice

Overview:

Get the device field of the given VGPU.

Signature:

```
string get_device (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getextraargs

Overview:

Get the extraargs field of the given VGPU.

Signature:

```
string get_extra_args (session ref session_ref, VGPU ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getGPUgroup

Overview:

Get the GPUgroup field of the given VGPU.

Signature:

```
GPU_group ref get_GPU_group (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VGPU.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_other_config (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPCI

Overview:

Get the PCI field of the given VGPU.

Signature:

```
PCI ref get_PCI (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

Overview:

Get a record containing the current state of the given VGPU.

Signature:

```
VGPU record get_record (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU record`

all fields from the object

RPC name: getresidenton

Overview:

Get the residenton field of the given VGPU.

Signature:

```
PGPU ref get_resident_on (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `PGPU ref`

value of the field

RPC name: getscheduledtoberesidenton

Overview:

Get the scheduledtoberesidenton field of the given VGPU.

Signature:

```
PGPU ref get_scheduled_to_be_resident_on (session ref session_ref, VGPU
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VGPU.

Signature:

```
VGPU_type ref get_type (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VGPU.

Signature:

```
string get_uuid (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VGPU.

Signature:

```
VM ref get_VM (session ref session_ref, VGPU ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VGPU. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VGPU ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextraargs

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the extraargs field of the given VGPU.

Signature:

```
void set_extra_args (session ref session_ref, VGPU ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VGPU.

Signature:

```
void set_other_config (session ref session_ref, VGPU ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Class: VGPUtype

A type of virtual GPU

Fields for class: VGPUtype

FIELD	TYPE	QUALIFIER	DESCRIPTION
`compatible_types_in_vm`	`VGPU_type ref set`	*RO/runtime*	List of VGPU types which are compatible in one VM
`enabled_on_GPU_groups`	`GPU_group ref set`	*RO/runtime*	List of GPU groups in which at least one have this VGPU type enabled
`enabled_on_PGPUs`	`PGPU ref set`	*RO/runtime*	List of PGPUs that have this VGPU type enabled
`experimental`	`bool`	*RO/constructor*	Indicates whether VGPUs of this type should be considered experimental
`framebuffer_size`	`int`	*RO/constructor*	Framebuffer size of the VGPU type, in bytes
`identifier`	`string`	*RO/constructor*	Key used to identify VGPU types and avoid creating duplicates - this field is used internally and not intended for interpretation by API clients

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`implementation`	`vgpu_type_implementation`	*RO/constructor*	The internal implementation of this VGPU type
`max_heads`	`int`	*RO/constructor*	Maximum number of displays supported by the VGPU type
`max_resolution_x`	`int`	*RO/constructor*	Maximum resolution (width) supported by the VGPU type
`max_resolution_y`	`int`	*RO/constructor*	Maximum resolution (height) supported by the VGPU type
`model_name`	`string`	*RO/constructor*	Model name associated with the VGPU type
`supported_on_GPU_groups`	`GPU_group ref set`	*RO/runtime*	List of GPU groups in which at least one PGPU supports this VGPU type
`supported_on_PGPUs`	`PGPU ref set`	*RO/runtime*	List of PGPUs that support this VGPU type
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_name`	`string`	*RO/constructor*	Name of VGPU vendor
`VGPUs`	`VGPU ref set`	*RO/runtime*	List of VGPUs of this type

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VGPUtype

RPC name: getall

Overview:

Return a list of all the VGPUtypes known to the system.

Signature:

```
VGPU_type ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VGPU_type ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VGPUtype references to VGPUtype records for all VGPUtypes known to the system.

Signature:

```
(VGPU_type ref -> VGPU_type record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VGPU_type ref -> VGPU_type record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VGPUtype instance with the specified UUID.

Signature:

```
VGPU_type ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VGPU_type ref`

reference to the object

RPC name: getcompatibletypesinvm

Overview:

Get the compatibletypesinvm field of the given VGPUtype.

Signature:

```
VGPU_type ref set get_compatible_types_in_vm (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getenabledonGPUgroups

Overview:

Get the enabledonGPUgroups field of the given VGPUtype.

Signature:

```
GPU_group ref set get_enabled_on_GPU_groups (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref set`

value of the field

RPC name: getenabledonPGPUs

Overview:

Get the enabledonPGPUs field of the given VGPUtype.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PGPU ref set get_enabled_on_PGPUs (session ref session_ref, VGPU_type ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getexperimental

Overview:

Get the experimental field of the given VGPUtype.

Signature:

```
bool get_experimental (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getframebuffersize

Overview:

Get the framebuffersize field of the given VGPUtype.

Signature:

```
int get_framebuffer_size (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getidentifier

Overview:

Get the identifier field of the given VGPUtype.

Signature:

```
string get_identifier (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getimplementation

Overview:

Get the implementation field of the given VGPUtype.

Signature:

```
vgpu_type_implementation get_implementation (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vgpu_type_implementation`

value of the field

RPC name: getmaxheads

Overview:

Get the maxheads field of the given VGPUtype.

Signature:

```
int get_max_heads (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmaxresolutionx

Overview:

Get the maxresolutionx field of the given VGPUtype.

Signature:

```
int get_max_resolution_x (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmaxresolutiony

Overview:

Get the maxresolutiony field of the given VGPUtype.

Signature:

```
int get_max_resolution_y (session ref session_ref, VGPU_type ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmodelname

Overview:

Get the modelname field of the given VGPUtype.

Signature:

```
string get_model_name (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VGPUtype.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VGPU_type record get_record (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type record`

all fields from the object

RPC name: getsupportedonGPUgroups

Overview:

Get the supportedonGPUgroups field of the given VGPUtype.

Signature:

```
GPU_group ref set get_supported_on_GPU_groups (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsupportedonPGPUs

Overview:

Get the supportedonPGPUs field of the given VGPUtype.

Signature:

```
PGPU ref set get_supported_on_PGPUs (session ref session_ref, VGPU_type
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VGPUtype.

Signature:

```
string get_uuid (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorname

Overview:

Get the vendorname field of the given VGPUtype.

Signature:

```
string get_vendor_name (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVGPUs

Overview:

Get the VGPUs field of the given VGPUtype.

Signature:

```
VGPU ref set get_VGPUs (session ref session_ref, VGPU_type ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

Class: VIF

A virtual network interface

Fields for class: VIF

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vif_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`current_operations`	`(string -> vif_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`currently_attached`	`bool`	*RO/constructor*	is the device currently attached (erased on reboot)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`device`	`string`	*RO/constructor*	order in which VIF backends are created by xapi. Guaranteed to be an unsigned decimal integer.
`ipv4_addresses`	`string set`	*RO/runtime*	IPv4 addresses in CIDR format
`ipv4_allowed`	`string set`	*RO/constructor*	A list of IPv4 addresses which can be used to filter traffic passing through this VIF
`ipv4_configuration_mode`	`vif_ipv4_configuration_mode`	*RO/runtime*	Determines whether IPv4 addresses are configured on the VIF
`ipv4_gateway`	`string`	*RO/runtime*	IPv4 gateway (the empty string means that no gateway is set)
`ipv6_addresses`	`string set`	*RO/runtime*	IPv6 addresses in CIDR format
`ipv6_allowed`	`string set`	*RO/constructor*	A list of IPv6 addresses which can be used to filter traffic passing through this VIF
`ipv6_configuration_mode`	`vif_ipv6_configuration_mode`	*RO/runtime*	Determines whether IPv6 addresses are configured on the VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`ipv6_gateway`	`string`	*RO/runtime*	IPv6 gateway (the empty string means that no gateway is set)
`locking_mode`	`vif_locking_mode`	*RO/constructor*	current locking mode of the VIF
`MAC`	`string`	*RO/constructor*	ethernet MAC address of virtual interface, as exposed to guest
`MAC_autogenerated`	`bool`	*RO/runtime*	true if the MAC was autogenerated; false indicates it was set manually
`metrics`	`VIF_metrics ref`	*RO/runtime*	**Removed.** metrics associated with this VIF
`MTU`	`int`	*RO/constructor*	MTU in octets
`network`	`network ref`	*RO/constructor*	virtual network to which this vif is connected
`other_config`	`(string -> string) map`	*RW*	additional configuration
`qos_algorithm_params`	`(string -> string) map`	*RW*	parameters for chosen QoS algorithm
`qos_algorithm_type`	`string`	*RW*	QoS algorithm to use
`qos_supported_algorithms`	`string set`	*RO/runtime*	supported QoS algorithms for this VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`runtime_properties`	`(string -> string) map`	*RO/runtime*	Device runtime properties
`status_code`	`int`	*RO/runtime*	error/success code associated with last attach-operation (erased on reboot)
`status_detail`	`string`	*RO/runtime*	error/success information associated with last attach-operation status (erased on reboot)
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VM`	`VM ref`	*RO/constructor*	virtual machine to which this vif is connected

RPCs associated with class: VIF

RPC name: addipv4allowed

Overview:

Associates an IPv4 address with this VIF

Signature:

```
void add_ipv4_allowed (session ref session_ref, VIF ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	The VIF which the IP address will be associated with
`string`	`value`	The IP address which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: addipv6allowed

Overview:

Associates an IPv6 address with this VIF

Signature:

```
void add_ipv6_allowed (session ref session_ref, VIF ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF which the IP address will be associated with
`string`	`value`	The IP address which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VIF.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void add_to_other_config (session ref session_ref, VIF ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoqosalgorithmparams

Overview:

Add the given keyvalue pair to the qos/algorithmparams field of the given VIF.

Signature:

```
void add_to_qos_algorithm_params (session ref session_ref, VIF ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: configureipv4

Overview:

Configure IPv4 settings for this virtual interface

Signature:

```
void configure_ipv4 (session ref session_ref, VIF ref self,
vif_ipv4_configuration_mode mode, string address, string gateway)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to configure
`vif_ipv4_configuration_mode`	`mode`	Whether to use DHCP, static or no IPv4 assignment
`string`	`address`	The IPv4 address in <addr>/<prefix length> format (for static mode only)
`string`	`gateway`	The IPv4 gateway (for static mode only; leave empty to not set a gateway)

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: configureipv6

Overview:

Configure IPv6 settings for this virtual interface

Signature:

```
void configure_ipv6 (session ref session_ref, VIF ref self,
vif_ipv6_configuration_mode mode, string address, string gateway)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to configure
`vif_ipv6_configuration_mode`	`mode`	Whether to use autoconf, static or no IPv6 assignment
`string`	`address`	The IPv6 address in <addr>/<prefix length> format (for static mode only)
`string`	`gateway`	The IPv6 gateway (for static mode only; leave empty to not set a gateway)

Minimum Role: vm-operator

Return Type: `void`

RPC name: create

Overview:

Create a new VIF instance, and return its handle.

Signature:

```
VIF ref create (session ref session_ref, VIF record args)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VIF ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VIF instance.

Signature:

```
void destroy (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VIFs known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VIF ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VIF ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VIF references to VIF records for all VIFs known to the system.

Signature:

```
(VIF ref -> VIF record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VIF ref -> VIF record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
vif_operations set get_allowed_operations (session ref session_ref, VIF
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_operations set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VIF instance with the specified UUID.

Signature:

```
VIF ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VIF ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VIF.

Signature:

```
(string -> vif_operations) map get_current_operations (session ref
session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vif_operations) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VIF.

Signature:

```
bool get_currently_attached (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdevice

Overview:

Get the device field of the given VIF.

Signature:

```
string get_device (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getipv4addresses

Overview:

Get the ipv4addresses field of the given VIF.

Signature:

```
string set get_ipv4_addresses (session ref session_ref, VIF ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv4allowed

Overview:

Get the ipv4allowed field of the given VIF.

Signature:

```
string set get_ipv4_allowed (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv4configurationmode

Overview:

Get the ipv4configurationmode field of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
vif_ipv4_configuration_mode get_ipv4_configuration_mode (session ref
session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_ipv4_configuration_mode`

value of the field

RPC name: getipv4gateway

Overview:

Get the ipv4gateway field of the given VIF.

Signature:

```
string get_ipv4_gateway (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getipv6addresses

Overview:

Get the ipv6addresses field of the given VIF.

Signature:

```
string set get_ipv6_addresses (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv6allowed

Overview:

Get the ipv6allowed field of the given VIF.

Signature:

```
string set get_ipv6_allowed (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string set`

value of the field

RPC name: getipv6configurationmode

Overview:

Get the ipv6configurationmode field of the given VIF.

Signature:

```
vif_ipv6_configuration_mode get_ipv6_configuration_mode (session ref
session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_ipv6_configuration_mode`

value of the field

RPC name: getipv6gateway

Overview:

Get the ipv6gateway field of the given VIF.

Signature:

```
string get_ipv6_gateway (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getlockingmode

Overview:

Get the lockingmode field of the given VIF.

Signature:

```
vif_locking_mode get_locking_mode (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_locking_mode`

value of the field

RPC name: getMAC

Overview:

Get the MAC field of the given VIF.

Signature:

```
string get_MAC (session ref session_ref, VIF ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getMACautogenerated

Overview:

Get the MACautogenerated field of the given VIF.

Signature:

```
bool get_MAC_autogenerated (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetrics

This message is removed.

Overview:

Get the metrics field of the given VIF.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VIF_metrics ref get_metrics (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF_metrics ref`

value of the field

RPC name: getMTU

Overview:

Get the MTU field of the given VIF.

Signature:

```
int get_MTU (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnetwork

Overview:

Get the network field of the given VIF.

Signature:

```
network ref get_network (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VIF.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmparams

Overview:

Get the qos/algorithmparams field of the given VIF.

Signature:

```
(string -> string) map get_qos_algorithm_params (session ref session_ref,
VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmtype

Overview:

Get the qos/algorithmtype field of the given VIF.

Signature:

```
string get_qos_algorithm_type (session ref session_ref, VIF ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getqossupportedalgorithms

Overview:

Get the qos/supportedalgorithms field of the given VIF.

Signature:

```
string set get_qos_supported_algorithms (session ref session_ref, VIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VIF record get_record (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF record`

all fields from the object

RPC name: getruntimeproperties

Overview:

Get the runtimeproperties field of the given VIF.

Signature:

```
(string -> string) map get_runtime_properties (session ref session_ref,
VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getstatuscode

Overview:

Get the statuscode field of the given VIF.

Signature:

```
int get_status_code (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getstatusdetail

Overview:

Get the statusdetail field of the given VIF.

Signature:

```
string get_status_detail (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VIF.

Signature:

```
string get_uuid (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VIF.

Signature:

```
VM ref get_VM (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: move

Overview:

Move the specified VIF to the specified network, even while the VM is running

Signature:

```
void move (session ref session_ref, VIF ref self, network ref network)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to move
`network ref`	`network`	The network to move it to

Minimum Role: vm-admin

Return Type: `void`

RPC name: plug

Overview:

Hotplug the specified VIF, dynamically attaching it to the running VM

Signature:

```
void plug (session ref session_ref, VIF ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to hotplug

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VIF. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VIF ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromqosalgorithmparams

Overview:

Remove the given key and its corresponding value from the qos/algorithmparams field of the given VIF. If the key is not in that Map, then do nothing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void remove_from_qos_algorithm_params (session ref session_ref, VIF ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removeipv4allowed

Overview:

Removes an IPv4 address from this VIF

Signature:

```
void remove_ipv4_allowed (session ref session_ref, VIF ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF from which the IP address will be removed
`string`	`value`	The IP address which will be removed from the VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: removeipv6allowed

Overview:

Removes an IPv6 address from this VIF

Signature:

```
void remove_ipv6_allowed (session ref session_ref, VIF ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF from which the IP address will be removed
`string`	`value`	The IP address which will be removed from the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: setipv4allowed

Overview:

Set the IPv4 addresses to which traffic on this VIF can be restricted

Signature:

```
void set_ipv4_allowed (session ref session_ref, VIF ref self, string set
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF which the IP addresses will be associated with
`string set`	`value`	The IP addresses which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: setipv6allowed

Overview:

Set the IPv6 addresses to which traffic on this VIF can be restricted

Signature:

```
void set_ipv6_allowed (session ref session_ref, VIF ref self, string set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF which the IP addresses will be associated with
`string set`	`value`	The IP addresses which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setlockingmode

Overview:

Set the locking mode for this VIF

Signature:

```
void set_locking_mode (session ref session_ref, VIF ref self,
vif_locking_mode value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF whose locking mode will be set
`vif_locking_mode`	`value`	The new locking mode for the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VIF.

Signature:

```
void set_other_config (session ref session_ref, VIF ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmparams

Overview:

Set the qos/algorithmparams field of the given VIF.

Signature:

```
void set_qos_algorithm_params (session ref session_ref, VIF ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmtype

Overview:

Set the qos/algorithmtype field of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_qos_algorithm_type (session ref session_ref, VIF ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: unplug

Overview:

Hotunplug the specified VIF, dynamically unattaching it from the running VM

Signature:

```
void unplug (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to hot-unplug

Minimum Role: vm-admin

Return Type: `void`

RPC name: unplugforce

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Forcibly unplug the specified VIF

Signature:

```
void unplug_force (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to forcibly unplug

Minimum Role: vm-admin

Return Type: `void`

Class: VIFmetrics

This class is removed.

The metrics associated with a virtual network device

Fields for class: VIFmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`io_read_kbs`	`float`	*RO/runtime*	**Removed.** Read bandwidth (KiB/s)
`io_write_kbs`	`float`	*RO/runtime*	**Removed.** Write bandwidth (KiB/s)
`last_updated`	`datetime`	*RO/runtime*	**Removed.** Time at which this information was last updated
`other_config`	`(string -> string) map`	*RW*	**Removed.** additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	**Removed.** Unique identifier/object reference

RPCs associated with class: VIFmetrics

RPC name: addtootherconfig

This message is removed.

Overview:

Add the given keyvalue pair to the otherconfig field of the given VIFmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VIF_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

This message is removed.

Overview:

Return a list of all the VIFmetrics instances known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VIF_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VIF_metrics ref set`

references to all objects

RPC name: getallrecords

This message is removed.

Overview:

Return a map of VIFmetrics references to VIFmetrics records for all VIFmetrics instances known to the system.

Signature:

```
(VIF_metrics ref -> VIF_metrics record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VIF_metrics ref -> VIF_metrics record) map`

records of all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbyuuid

This message is removed.

Overview:

Get a reference to the VIFmetrics instance with the specified UUID.

Signature:

```
VIF_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VIF_metrics ref`

reference to the object

RPC name: getioreadkbs

This message is removed.

Overview:

Get the io/readkbs field of the given VIFmetrics.

Signature:

```
float get_io_read_kbs (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getiowritekbs

This message is removed.

Overview:

Get the io/writekbs field of the given VIFmetrics.

Signature:

```
float get_io_write_kbs (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getlastupdated

This message is removed.

Overview:

Get the lastupdated field of the given VIFmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
datetime get_last_updated (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getotherconfig

This message is removed.

Overview:

Get the otherconfig field of the given VIFmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

This message is removed.

Overview:

Get a record containing the current state of the given VIFmetrics.

Signature:

```
VIF_metrics record get_record (session ref session_ref, VIF_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF_metrics record`

all fields from the object

RPC name: getuuid

This message is removed.

Overview:

Get the uuid field of the given VIFmetrics.

Signature:

```
string get_uuid (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is removed.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VIFmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VIF_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the otherconfig field of the given VIFmetrics.

Signature:

```
void set_other_config (session ref session_ref, VIF_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VLAN

A VLAN mux/demux

Fields for class: VLAN

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	additional configuration
`tag`	`int`	*RO/constructor*	VLAN tag in use
`tagged_PIF`	`PIF ref`	*RO/constructor*	interface on which traffic is tagged
`untagged_PIF`	`PIF ref`	*RO/runtime*	interface on which traffic is untagged
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VLAN

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VLAN.

Signature:

```
void add_to_other_config (session ref session_ref, VLAN ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a VLAN mux/demuxer

Signature:

```
VLAN ref create (session ref session_ref, PIF ref tagged_PIF, int tag,
network ref network)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`tagged_PIF`	PIF which receives the tagged traffic
`int`	`tag`	VLAN tag to use
`network ref`	`network`	Network to receive the untagged traffic

Minimum Role: pool-operator

Return Type: `VLAN ref`

The reference of the created VLAN object

RPC name: destroy

Overview:

Destroy a VLAN mux/demuxer

Signature:

```
void destroy (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	VLAN mux/demuxer to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return a list of all the VLANs known to the system.

Signature:

```
VLAN ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VLAN ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VLAN references to VLAN records for all VLANs known to the system.

Signature:

```
(VLAN ref -> VLAN record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VLAN ref -> VLAN record) map`

records of all objects

RPC name: getbyuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a reference to the VLAN instance with the specified UUID.

Signature:

```
VLAN ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VLAN ref`

reference to the object

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VLAN.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VLAN ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VLAN.

Signature:

```
VLAN record get_record (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VLAN record`

all fields from the object

RPC name: gettag

Overview:

Get the tag field of the given VLAN.

Signature:

```
int get_tag (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gettaggedPIF

Overview:

Get the taggedPIF field of the given VLAN.

Signature:

```
PIF ref get_tagged_PIF (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getuntaggedPIF

Overview:

Get the untaggedPIF field of the given VLAN.

Signature:

```
PIF ref get_untagged_PIF (session ref session_ref, VLAN ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VLAN.

Signature:

```
string get_uuid (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VLAN. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_other_config (session ref session_ref, VLAN ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VLAN.

Signature:

```
void set_other_config (session ref session_ref, VLAN ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: VM

A virtual machine or 'guest'.

Fields for class: VM

FIELD	TYPE	QUALIFIER	DESCRIPTION
`actions_after_crash`	`on_crash_behaviour`	*RO/constructor*	action to take if the guest crashes
`actions_after_reboot`	`on_normal_exit`	*RW*	action to take after the guest has rebooted itself
`actions_after_shutdown`	`on_normal_exit`	*RW*	action to take after the guest has shutdown itself
`actions_after_softreboot`	`on_softreboot_behavior`	*RW*	action to take after soft reboot
`affinity`	`host ref`	*RW*	A host which the VM has some affinity for (or NULL). This is used as a hint to the start call when it decides where to run the VM. Resource constraints may cause the VM to be started elsewhere.
`allowed_operations`	`vm_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`appliance`	`VM_appliance ref`	*RO/constructor*	the appliance to which this VM belongs
`attached_PCIs`	`PCI ref set`	*RO/runtime*	Currently passed-through PCI devices
`bios_strings`	`(string -> string) map`	*RO/runtime*	BIOS strings
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this VM
`blocked_operations`	`(vm_operations -> string) map`	*RO/constructor*	List of operations which have been explicitly blocked and an error code
`children`	`VM ref set`	*RO/runtime*	List pointing to all the children of this VM
`consoles`	`console ref set`	*RO/runtime*	virtual console devices
`crash_dumps`	`crashdump ref set`	*RO/runtime*	crash dumps associated with this VM
`current_operations`	`(string -> vm_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`domain_type`	`domain_type`	*RO/constructor*	The type of domain that will be created

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			when the VM is started
`domarch`	`string`	*RO/runtime*	Domain architecture (if available, null string otherwise)
`domid`	`int`	*RO/runtime*	domain ID (if available, -1 otherwise)
`generation_id`	`string`	*RO/constructor*	Generation ID of the VM
`groups`	`VM_group ref set`	*RO/runtime*	VM groups associated with the VM
`guest_metrics`	`VM_guest_metrics ref`	*RO/runtime*	metrics associated with the running guest
`ha_always_run`	`bool`	*RO/constructor*	**Deprecated.** if true then the system will attempt to keep the VM running as much as possible.
`ha_restart_priority`	`string`	*RO/constructor*	has possible values: "best-effort" meaning "try to restart this VM if possible but don't consider the Pool to be overcommitted if this is not possible"; "restart" meaning "this VM should be restarted"; "" meaning "do not try to restart this VM"

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`hardware_platform_version`	`int`	*RW*	The host virtual hardware platform version the VM can run on
`has_vendor_device`	`bool`	*RO/constructor*	When an HVM guest starts, this controls the presence of the emulated C000 PCI device which triggers Windows Update to fetch or update PV drivers.
`HVM_boot_params`	`(string -> string) map`	*RW*	HVM boot params
`HVM_boot_policy`	`string`	*RO/constructor*	**Deprecated.** HVM boot policy
`HVM_shadow_multiplier`	`float`	*RO/constructor*	multiplier applied to the amount of shadow that will be made available to the guest
`is_a_snapshot`	`bool`	*RO/runtime*	true if this is a snapshot. Snapshotted VMs can never be started, they are used only for cloning other VMs
`is_a_template`	`bool`	*RW*	true if this is a template. Template VMs can never be started, they are used only for cloning other VMs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`is_control_domain`	`bool`	*RO/runtime*	true if this is a control domain (domain 0 or a driver domain)
`is_default_template`	`bool`	*RO/runtime*	true if this is a default template. Default template VMs can never be started or migrated, they are used only for cloning other VMs
`is_snapshot_from_vmpp`	`bool`	*RO/constructor*	**Removed.** true if this snapshot was created by the protection policy
`is_vmss_snapshot`	`bool`	*RO/constructor*	true if this snapshot was created by the snapshot schedule
`last_boot_CPU_flags`	`(string -> string) map`	*RO/constructor*	describes the CPU flags on which the VM was last booted
`last_booted_record`	`string`	*RO/constructor*	marshalled value containing VM record at time of last boot
`memory_dynamic_max`	`int`	*RO/constructor*	Dynamic maximum (bytes)
`memory_dynamic_min`	`int`	*RO/constructor*	Dynamic minimum (bytes)
`memory_overhead`	`int`	*RO/runtime*	Virtualization memory overhead (bytes).
`memory_static_max`	`int`	*RO/constructor*	Statically-set (i.e. absolute) maximum

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			(bytes). The value of this field at VM start time acts as a hard limit of the amount of memory a guest can use. New values only take effect on reboot.
`memory_static_min`	`int`	*RO/constructor*	Statically-set (i.e. absolute) mininum (bytes). The value of this field indicates the least amount of memory this VM can boot with without crashing.
`memory_target`	`int`	*RO/constructor*	**Deprecated.** Dynamically-set memory target (bytes). The value of this field indicates the current target for memory available to this VM.
`metrics`	`VM_metrics ref`	*RO/runtime*	metrics associated with this VM
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`NVRAM`	`(string -> string) map`	*RO/constructor*	initial value for guest NVRAM (containing UEFI variables, etc). Cannot be changed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			while the VM is running
`order`	`int`	*RO/constructor*	The point in the startup or shutdown sequence at which this VM will be started
`other_config`	`(string -> string) map`	*RO/constructor*	additional configuration
`parent`	`VM ref`	*RO/runtime*	Ref pointing to the parent of this VM
`PCI_bus`	`string`	*RW*	**Deprecated.** PCI bus path for pass-through devices
`pending_guidances`	`update_guidances set`	*RO/runtime*	The set of pending mandatory guidances after applying updates, which must be applied, otherwise there may be, for example, VM failures
`pending_guidances_full`	`update_guidances set`	*RO/runtime*	The set of pending full guidances after applying updates, which a user should follow to make some updates, for example, specific hardware drivers or CPU features, fully effective, but the 'average user' doesn't need to

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`pending_guidances_recommended`	`update_guidances set`	*RO/runtime*	The set of pending recommended guidances after applying updates, which most users should follow to make the updates effective, but if not followed, will not cause a failure
`platform`	`(string -> string) map`	*RO/constructor*	platform-specific configuration
`power_state`	`vm_power_state`	*RO/constructor*	Current power state of the machine
`protection_policy`	`VMPP ref`	*RO/constructor*	**Deprecated.** Ref pointing to a protection policy for this VM
`PV_args`	`string`	*RW*	kernel command-line arguments
`PV_bootloader`	`string`	*RW*	name of or path to bootloader
`PV_bootloader_args`	`string`	*RW*	miscellaneous arguments for the bootloader
`PV_kernel`	`string`	*RW*	path to the kernel
`PV_legacy_args`	`string`	*RW*	to make Zurich guests boot
`PV_ramdisk`	`string`	*RW*	path to the initrd
`recommendations`	`string`	*RW*	An XML specification of recommended values and ranges

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			for properties of this VM
`reference_label`	`string`	*RO/constructor*	Textual reference to the template used to create a VM. This can be used by clients in need of an immutable reference to the template since the latter's uuid and name_label may change, for example, after a package installation or upgrade.
`requires_reboot`	`bool`	*RO/runtime*	Indicates whether a VM requires a reboot in order to update its configuration, for example, its memory allocation.
`resident_on`	`host ref`	*RO/runtime*	the host the VM is currently resident on
`scheduled_to_be_resident_on`	`host ref`	*RO/runtime*	the host on which the VM is due to be started/resumed/migrated. This acts as a memory reservation indicator
`secureboot_certificates_state`	`vm_secureboot_certificates_state`	*RO/runtime*	The state of the Secure Boot certificates, showing whether an update is available, already scheduled, or not needed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`shutdown_delay`	`int`	*RO/constructor*	The delay to wait before proceeding to the next order in the shutdown sequence (seconds)
`snapshot_info`	`(string -> string) map`	*RO/runtime*	Human-readable information concerning this snapshot
`snapshot_metadata`	`string`	*RO/runtime*	Encoded information about the VM's metadata this is a snapshot of
`snapshot_of`	`VM ref`	*RO/runtime*	Ref pointing to the VM this snapshot is of.
`snapshot_schedule`	`VMSS ref`	*RO/constructor*	Ref pointing to a snapshot schedule for this VM
`snapshot_time`	`datetime`	*RO/runtime*	Date/time when this snapshot was created.
`snapshots`	`VM ref set`	*RO/runtime*	List pointing to all the VM snapshots.
`start_delay`	`int`	*RO/constructor*	The delay to wait before proceeding to the next order in the startup sequence (seconds)
`suspend_SR`	`SR ref`	*RW*	The SR on which a suspend image is stored

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`suspend_VDI`	`VDI ref`	*RO/constructor*	The VDI that a suspend image is stored on. (Only has meaning if VM is currently suspended)
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`transportable_snapshot_id`	`string`	*RO/runtime*	Transportable ID of the snapshot VM
`user_version`	`int`	*RW*	Creators of VMs and templates may store version information here.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VBDs`	`VBD ref set`	*RO/runtime*	virtual block devices
`VCPUs_at_startup`	`int`	*RO/constructor*	Boot number of VCPUs
`VCPUs_max`	`int`	*RO/constructor*	Max number of VCPUs
`VCPUs_params`	`(string -> string) map`	*RW*	configuration parameters for the selected VCPU policy
`version`	`int`	*RO/constructor*	The number of times this VM has been recovered
`VGPUs`	`VGPU ref set`	*RO/runtime*	Virtual GPUs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VIFs`	`VIF ref set`	*RO/runtime*	virtual network interfaces
`VTPMs`	`VTPM ref set`	*RO/runtime*	virtual TPMs
`VUSBs`	`VUSB ref set`	*RO/runtime*	virtual usb devices
`xenstore_data`	`(string -> string) map`	*RW*	data to be inserted into the xenstore tree (/local/domain/<domid>/vm-data) after the VM is created.

RPCs associated with class: VM

RPC name: addtags

Overview:

Add the given value to the tags field of the given VM. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: addtoblockedoperations

Overview:

Update list of operations which have been explicitly blocked and an error code

Signature:

```
void add_to_blocked_operations (session ref session_ref, VM ref self,
vm_operations key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`vm_operations`	`key`	Blocked operation
`string`	`value`	Error code

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoHVMbootparams

Overview:

Add the given keyvalue pair to the HVM/bootparams field of the given VM.

Signature:

```
void add_to_HVM_boot_params (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoNVRAM

Overview:

Signature:

```
void add_to_NVRAM (session ref session_ref, VM ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`key`	The key
`string`	`value`	The value

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_other_config (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoplatform

Overview:

Add the given keyvalue pair to the platform field of the given VM.

Signature:

```
void add_to_platform (session ref session_ref, VM ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoVCPUsparams

Overview:

Add the given keyvalue pair to the VCPUs/params field of the given VM.

Signature:

```
void add_to_VCPUs_params (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoVCPUsparamslive

Overview:

Add the given keyvalue pair to VM.VCPUsparams, and apply that value on the running VM

Signature:

```
void add_to_VCPUs_params_live (session ref session_ref, VM ref self,
string key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`key`	The key
`string`	`value`	The value

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoxenstoredata

Overview:

Add the given keyvalue pair to the xenstoredata field of the given VM.

Signature:

```
void add_to_xenstore_data (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: assertagile

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Returns an error if the VM is not considered agile, for example, because it is tied to a resource local to a host

Signature:

```
void assert_agile (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: read-only

Return Type: `void`

RPC name: assertcanberecovered

Overview:

Assert whether all SRs required to recover this VM are available.

Signature:

```
void assert_can_be_recovered (session ref session_ref, VM ref self,
session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to recover
`session ref`	`session_to`	The session to which the VM is to be recovered.

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `VM_IS_PART_OF_AN_APPLIANCE`, `VM_REQUIRES_SR`

RPC name: assertcanboothere

Overview:

Returns an error if the VM could not boot on this host for some reason

Signature:

```
void assert_can_boot_here (session ref session_ref, VM ref self, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`host ref`	`host`	The host

Minimum Role: read-only

Return Type: `void`

Possible Error Codes: `HOST_NOT_ENOUGH_FREE_MEMORY`, `HOST_NOT_ENOUGH_PCPUS`, `NETWORK_SRIOV_INSUFFICIENT_CAPACITY`, `HOST_NOT_LIVE`, `HOST_DISABLED`, `HOST_CANNOT_ATTACH_NETWORK`, `VM_HVM_REQUIRED`, `VM_REQUIRES_GPU`, `VM_REQUIRES_IOMMU`, `VM_REQUIRES_NETWORK`, `VM_REQUIRES_SR`, `VM_REQUIRES_VGPU`, `VM_HOST_INCOMPATIBLE_VERSION`, `VM_HOST_INCOMPATIBLE_VIRTUAL_HARDWARE_PLATFORM_VERSION`, `INVALID_VALUE`, `MEMORY_CONSTRAINT_VIOLATION`, `OPERATION_NOT_ALLOWED`, `VALUE_NOT_SUPPORTED`, `VM_INCOMPATIBLE_WITH_THIS_HOST`

RPC name: assertcanmigrate

Overview:

Assert whether a VM can be migrated to the specified destination.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void assert_can_migrate (session ref session_ref, VM ref vm, (string ->
string) map dest, bool live, (VDI ref -> SR ref) map vdi_map, (VIF ref ->
network ref) map vif_map, (string -> string) map options, (VGPU ref ->
GPU_group ref) map vgpu_map)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`(string -> string) map`	`dest`	The result of a Host.migrate_receive call.
`bool`	`live`	Live migration
`(VDI ref -> SR ref) map`	`vdi_map`	Map of source VDI to destination SR
`(VIF ref -> network ref) map`	`vif_map`	Map of source VIF to destination network
`(string -> string) map`	`options`	Other parameters
`(VGPU ref -> GPU_group ref) map`	`vgpu_map`	Map of source vGPU to destination GPU group

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `LICENCE_RESTRICTION`

RPC name: assertoperationvalid

Overview:

Check to see whether this operation is acceptable in the current state of the system, raising an error if the operation is invalid for some reason

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void assert_operation_valid (session ref session_ref, VM ref self,
vm_operations op)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`vm_operations`	`op`	proposed operation

Minimum Role: read-only

Return Type: `void`

RPC name: callhostplugin

Overview:

Call an API plugin on the host where this vm resides

Signature:

```
string call_host_plugin (session ref session_ref, VM ref vm, string
plugin, string fn, (string -> string) map args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The vm
`string`	`plugin`	The name of the plugin
`string`	`fn`	The name of the function within the plugin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`args`	Arguments for the function

Minimum Role: vm-operator

Return Type: `string`

Result from the plugin

RPC name: callplugin

Overview:

Call an API plugin on this vm

Signature:

```
string call_plugin (session ref session_ref, VM ref vm, string plugin,
string fn, (string -> string) map args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The vm
`string`	`plugin`	The name of the plugin
`string`	`fn`	The name of the function within the plugin
`(string -> string) map`	`args`	Arguments for the function

Minimum Role: vm-operator

Return Type: `string`

Result from the plugin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: checkpoint

Overview:

Checkpoints the specified VM, making a new VM. Checkpoint automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write and saves the memory image as well.

Signature:

```
VM ref checkpoint (session ref session_ref, VM ref vm, string new_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be checkpointed
`string`	`new_name`	The name of the checkpointed VM

Minimum Role: vm-power-admin

Return Type: `VM ref`

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `VM_CHECKPOINT_SUSPEND_FAILED`, `VM_CHECKPOINT_RESUME_FAILED`

RPC name: cleanreboot

Overview:

Attempt to cleanly shutdown the specified VM Note: this may not be supported, for example, if a guest agent is not installed. This can only be called when the specified VM is in the Running state.

Signature:

```
void clean_reboot (session ref session_ref, VM ref vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to shutdown

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: cleanshutdown

Overview:

Attempt to cleanly shutdown the specified VM. Note: this may not be supported, for example, if a guest agent is not installed. This can only be called when the specified VM is in the Running state.

Signature:

```
void clean_shutdown (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to shutdown

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: clone

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Clones the specified VM, making a new VM. Clone automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write. This function can only be called when the VM is in the Halted State.

Signature:

```
VM ref clone (session ref session_ref, VM ref vm, string new_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be cloned
`string`	`new_name`	The name of the cloned VM

Minimum Role: vm-admin

Return Type: `VM ref`

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: computememoryoverhead

Overview:

Computes the virtualization memory overhead of a VM.

Signature:

```
int compute_memory_overhead (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`vm`	The VM for which to compute the memory overhead

Minimum Role: read-only

Return Type: `int`

the virtualization memory overhead of the VM.

RPC name: copy

Overview:

Copied the specified VM, making a new VM. Unlike clone, copy does not exploits the capabilities of the underlying storage repository in which the VM's disk images are stored. Instead, copy guarantees that the disk images of the newly created VM will be 'full disks' i.e. not part of a CoW chain. This function can only be called when the VM is in the Halted State.

Signature:

```
VM ref copy (session ref session_ref, VM ref vm, string new_name, SR ref
sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be copied
`string`	`new_name`	The name of the copied VM
`SR ref`	`sr`	An SR to copy all the VM's disks into (if an invalid reference then it uses the existing SRs)

Minimum Role: vm-admin

Return Type: `VM ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: copybiosstrings

Overview:

Copy the BIOS strings from the given host to this VM

Signature:

```
void copy_bios_strings (session ref session_ref, VM ref vm, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to modify
`host ref`	`host`	The host to copy the BIOS strings from

Minimum Role: vm-admin

Return Type: `void`

RPC name: create

Overview:

NOT RECOMMENDED VM.clone or VM.copy or VM.import is a better choice in almost all situations. The standard way to obtain a new VM is to call VM.clone on a template VM, then call VM.provision on the new clone. Caution: if VM.create is used and then the new VM is attached to a virtual disc that has an operating system already installed, then there is no guarantee that the operating system will boot and run. Any software that calls VM.create on a future version of this API may fail or give unexpected results. For example this could happen if an additional parameter were added to VM.create. VM.create is intended only for use in the automatic creation of the system VM templates. It creates a new VM instance, and returns its handle.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM ref create (session ref session_ref, VM record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VM ref`

reference to the newly created object

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this VM

Signature:

```
blob ref create_new_blob (session ref session_ref, VM ref vm, string name,
string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`string`	`name`	The name associated with the blob

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: vm-power-admin

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: destroy

Overview:

Destroy the specified VM. The VM is completely removed from the system. This function can only be called when the VM is in the Halted State.

Signature:

```
void destroy (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: forgetdatasourcearchives

Overview:

Forget the recorded statistics related to the specified data source

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void forget_data_source_archives (session ref session_ref, VM ref self,
string data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`data_source`	The data source whose archives are to be forgotten

Minimum Role: vm-admin

Return Type: `void`

RPC name: getactionsaftercrash

Overview:

Get the actions/aftercrash field of the given VM.

Signature:

```
on_crash_behaviour get_actions_after_crash (session ref session_ref, VM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_crash_behaviour`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getactionsafterreboot

Overview:

Get the actions/afterreboot field of the given VM.

Signature:

```
on_normal_exit get_actions_after_reboot (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_normal_exit`

value of the field

RPC name: getactionsaftershutdown

Overview:

Get the actions/aftershutdown field of the given VM.

Signature:

```
on_normal_exit get_actions_after_shutdown (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_normal_exit`

value of the field

RPC name: getactionsaftersoftreboot

Overview:

Get the actions/aftersoftreboot field of the given VM.

Signature:

```
on_softreboot_behavior get_actions_after_softreboot (session ref
session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_softreboot_behavior`

value of the field

RPC name: getaffinity

Overview:

Get the affinity field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref get_affinity (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getall

Overview:

Return a list of all the VMs known to the system.

Signature:

```
VM ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VM references to VM records for all VMs known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(VM ref -> VM record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM ref -> VM record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VM.

Signature:

```
vm_operations set get_allowed_operations (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_operations set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallowedVBDdevices

Overview:

Returns a list of the allowed values that a VBD device field can take

Signature:

```
string set get_allowed_VBD_devices (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to query

Minimum Role: read-only

Return Type: `string set`

The allowed values

RPC name: getallowedVIFdevices

Overview:

Returns a list of the allowed values that a VIF device field can take

Signature:

```
string set get_allowed_VIF_devices (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to query

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string set`

The allowed values

RPC name: getappliance

Overview:

Get the appliance field of the given VM.

Signature:

```
VM_appliance ref get_appliance (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_appliance ref`

value of the field

RPC name: getattachedPCIs

Overview:

Get the attachedPCIs field of the given VM.

Signature:

```
PCI ref set get_attached_PCIs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref set`

value of the field

RPC name: getbiosstrings

Overview:

Get the biosstrings field of the given VM.

Signature:

```
(string -> string) map get_bios_strings (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> blob ref) map get_blobs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getblockedoperations

Overview:

Get the blockedoperations field of the given VM.

Signature:

```
(vm_operations -> string) map get_blocked_operations (session ref
session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(vm_operations -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbootrecord

This message is deprecated.

Overview:

Returns a record describing the VM's dynamic state, initialised when the VM boots and updated to reflect runtime configuration changes, for example CPU hotplug

Signature:

```
VM record get_boot_record (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM whose boot-time state to return

Minimum Role: read-only

Return Type: `VM record`

A record describing the VM

RPC name: getbynamelabel

Overview:

Get all the VM instances with the given label.

Signature:

```
VM ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VM ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VM instance with the specified UUID.

Signature:

```
VM ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM ref`

reference to the object

RPC name: getchildren

Overview:

Get the children field of the given VM.

Signature:

```
VM ref set get_children (session ref session_ref, VM ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: getconsoles

Overview:

Get the consoles field of the given VM.

Signature:

```
console ref set get_consoles (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `console ref set`

value of the field

RPC name: getcooperative

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return true if the VM is currently 'cooperative' i.e. is expected to reach a balloon target and actually has done

Signature:

```
bool get_cooperative (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: read-only

Return Type: `bool`

true if the VM is currently 'cooperative'; false otherwise

RPC name: getcrashdumps

Overview:

Get the crashdumps field of the given VM.

Signature:

```
crashdump ref set get_crash_dumps (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `crashdump ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VM.

Signature:

```
(string -> vm_operations) map get_current_operations (session ref
session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vm_operations) map`

value of the field

RPC name: getdatasources

Overview:

Signature:

```
data_source record set get_data_sources (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to interrogate

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `data_source record set`

A set of data sources

RPC name: getdomaintype

Overview:

Get the domaintype field of the given VM.

Signature:

```
domain_type get_domain_type (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `domain_type`

value of the field

RPC name: getdomarch

Overview:

Get the domarch field of the given VM.

Signature:

```
string get_domarch (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdomid

Overview:

Get the domid field of the given VM.

Signature:

```
int get_domid (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getgenerationid

Overview:

Get the generationid field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_generation_id (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getgroups

Overview:

Get the groups field of the given VM.

Signature:

```
VM_group ref set get_groups (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_group ref set`

value of the field

RPC name: getguestmetrics

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the guestmetrics field of the given VM.

Signature:

```
VM_guest_metrics ref get_guest_metrics (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_guest_metrics ref`

value of the field

RPC name: gethaalwaysrun

This message is deprecated.

Overview:

Get the haalwaysrun field of the given VM.

Signature:

```
bool get_ha_always_run (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getharestartpriority

Overview:

Get the harestartpriority field of the given VM.

Signature:

```
string get_ha_restart_priority (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethardwareplatformversion

Overview:

Get the hardwareplatformversion field of the given VM.

Signature:

```
int get_hardware_platform_version (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gethasvendordevice

Overview:

Get the hasvendordevice field of the given VM.

Signature:

```
bool get_has_vendor_device (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getHVMbootparams

Overview:

Get the HVM/bootparams field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_HVM_boot_params (session ref session_ref, VM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getHVMbootpolicy

This message is deprecated.

Overview:

Get the HVM/bootpolicy field of the given VM.

Signature:

```
string get_HVM_boot_policy (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getHVMshadowmultiplier

Overview:

Get the HVM/shadowmultiplier field of the given VM.

Signature:

```
float get_HVM_shadow_multiplier (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getisasnapshot

Overview:

Get the isasnapshot field of the given VM.

Signature:

```
bool get_is_a_snapshot (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getisatemplate

Overview:

Get the isatemplate field of the given VM.

Signature:

```
bool get_is_a_template (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getiscontroldomain

Overview:

Get the iscontroldomain field of the given VM.

Signature:

```
bool get_is_control_domain (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisdefaulttemplate

Overview:

Get the isdefaulttemplate field of the given VM.

Signature:

```
bool get_is_default_template (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getissnapshotfromvmpp

This message is removed.

Overview:

Get the issnapshotfromvmpp field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_is_snapshot_from_vmpp (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisvmsssnapshot

Overview:

Get the isvmsssnapshot field of the given VM.

Signature:

```
bool get_is_vmss_snapshot (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlastbootCPUflags

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the lastbootCPUflags field of the given VM.

Signature:

```
(string -> string) map get_last_boot_CPU_flags (session ref session_ref,
VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlastbootedrecord

Overview:

Get the lastbootedrecord field of the given VM.

Signature:

```
string get_last_booted_record (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getmemorydynamicmax

Overview:

Get the memory/dynamicmax field of the given VM.

Signature:

```
int get_memory_dynamic_max (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorydynamicmin

Overview:

Get the memory/dynamicmin field of the given VM.

Signature:

```
int get_memory_dynamic_min (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemoryoverhead

Overview:

Get the memory/overhead field of the given VM.

Signature:

```
int get_memory_overhead (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorystaticmax

Overview:

Get the memory/staticmax field of the given VM.

Signature:

```
int get_memory_static_max (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorystaticmin

Overview:

Get the memory/staticmin field of the given VM.

Signature:

```
int get_memory_static_min (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorytarget

This message is deprecated.

Overview:

Get the memory/target field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int get_memory_target (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmetrics

Overview:

Get the metrics field of the given VM.

Signature:

```
VM_metrics ref get_metrics (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_metrics ref`

value of the field

RPC name: getnamedescription

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the name/description field of the given VM.

Signature:

```
string get_name_description (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VM.

Signature:

```
string get_name_label (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getNVRAM

Overview:

Get the NVRAM field of the given VM.

Signature:

```
(string -> string) map get_NVRAM (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getorder

Overview:

Get the order field of the given VM.

Signature:

```
int get_order (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VM.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getparent

Overview:

Get the parent field of the given VM.

Signature:

```
VM ref get_parent (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: getPCIbus

This message is deprecated.

Overview:

Get the PCIbus field of the given VM.

Signature:

```
string get_PCI_bus (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpendingguidances

Overview:

Get the pendingguidances field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
update_guidances set get_pending_guidances (session ref session_ref, VM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getpendingguidancesfull

Overview:

Get the pendingguidancesfull field of the given VM.

Signature:

```
update_guidances set get_pending_guidances_full (session ref session_ref,
VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpendingguidancesrecommended

Overview:

Get the pendingguidancesrecommended field of the given VM.

Signature:

```
update_guidances set get_pending_guidances_recommended (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getplatform

Overview:

Get the platform field of the given VM.

Signature:

```
(string -> string) map get_platform (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpossiblehosts

Overview:

Return the list of hosts on which this VM may run.

Signature:

```
host ref set get_possible_hosts (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM

Minimum Role: read-only

Return Type: `host ref set`

The possible hosts

RPC name: getpowerstate

Overview:

Get the powerstate field of the given VM.

Signature:

```
vm_power_state get_power_state (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_power_state`

value of the field

RPC name: getprotectionpolicy

This message is deprecated.

Overview:

Get the protectionpolicy field of the given VM.

Signature:

```
VMPP ref get_protection_policy (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VMPP ref`

value of the field

RPC name: getPVargs

Overview:

Get the PV/args field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_PV_args (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVbootloader

Overview:

Get the PV/bootloader field of the given VM.

Signature:

```
string get_PV_bootloader (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVbootloaderargs

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the PV/bootloaderargs field of the given VM.

Signature:

```
string get_PV_bootloader_args (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVkernel

Overview:

Get the PV/kernel field of the given VM.

Signature:

```
string get_PV_kernel (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getPVlegacyargs

Overview:

Get the PV/legacyargs field of the given VM.

Signature:

```
string get_PV_legacy_args (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVramdisk

Overview:

Get the PV/ramdisk field of the given VM.

Signature:

```
string get_PV_ramdisk (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getrecommendations

Overview:

Get the recommendations field of the given VM.

Signature:

```
string get_recommendations (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VM.

Signature:

```
VM record get_record (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM record`

all fields from the object

RPC name: getreferencelabel

Overview:

Get the referencelabel field of the given VM.

Signature:

```
string get_reference_label (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrequiresreboot

Overview:

Get the requiresreboot field of the given VM.

Signature:

```
bool get_requires_reboot (session ref session_ref, VM ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getresidenton

Overview:

Get the residenton field of the given VM.

Signature:

```
host ref get_resident_on (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getscheduledtoberesidenton

Overview:

Get the scheduledtoberesidenton field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref get_scheduled_to_be_resident_on (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getsecurebootcertificatesstate

Overview:

Get the securebootcertificatesstate field of the given VM.

Signature:

```
vm_secureboot_certificates_state get_secureboot_certificates_state
(session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_secureboot_certificates_state`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsecurebootreadiness

Overview:

Return the secureboot readiness of the VM

Signature:

```
vm_secureboot_readiness get_secureboot_readiness (session ref session_ref,
VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: pool-admin

Return Type: `vm_secureboot_readiness`

The secureboot readiness of the VM

RPC name: getshutdowndelay

Overview:

Get the shutdowndelay field of the given VM.

Signature:

```
int get_shutdown_delay (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getsnapshotinfo

Overview:

Get the snapshotinfo field of the given VM.

Signature:

```
(string -> string) map get_snapshot_info (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsnapshotmetadata

Overview:

Get the snapshotmetadata field of the given VM.

Signature:

```
string get_snapshot_metadata (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsnapshotof

Overview:

Get the snapshotof field of the given VM.

Signature:

```
VM ref get_snapshot_of (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: getsnapshotschedule

Overview:

Get the snapshotschedule field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VMSS ref get_snapshot_schedule (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VMSS ref`

value of the field

RPC name: getsnapshottime

Overview:

Get the snapshottime field of the given VM.

Signature:

```
datetime get_snapshot_time (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getsnapshots

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the snapshots field of the given VM.

Signature:

```
VM ref set get_snapshots (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: getSRsrequiredforrecovery

Overview:

List all the SR's that are required for the VM to be recovered

Signature:

```
SR ref set get_SRs_required_for_recovery (session ref session_ref, VM ref
self, session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM for which the SRs have to be recovered
`session ref`	`session_to`	The session to which the SRs of the VM have to be recovered.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `SR ref set`

refs for SRs required to recover the VM

RPC name: getstartdelay

Overview:

Get the startdelay field of the given VM.

Signature:

```
int get_start_delay (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getsuspendSR

Overview:

Get the suspendSR field of the given VM.

Signature:

```
SR ref get_suspend_SR (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getsuspendVDI

Overview:

Get the suspendVDI field of the given VM.

Signature:

```
VDI ref get_suspend_VDI (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: gettags

Overview:

Get the tags field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set get_tags (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettransportablesnapshotid

Overview:

Get the transportablesnapshotid field of the given VM.

Signature:

```
string get_transportable_snapshot_id (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuserversion

Overview:

Get the userversion field of the given VM.

Signature:

```
int get_user_version (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VM.

Signature:

```
string get_uuid (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getVBDs

Overview:

Get the VBDs field of the given VM.

Signature:

```
VBD ref set get_VBDs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VBD ref set`

value of the field

RPC name: getVCPUsatstartup

Overview:

Get the VCPUs/atstartup field of the given VM.

Signature:

```
int get_VCPUs_at_startup (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVCPUsmax

Overview:

Get the VCPUs/max field of the given VM.

Signature:

```
int get_VCPUs_max (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVCPUsparams

Overview:

Get the VCPUs/params field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_VCPUs_params (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getversion

Overview:

Get the version field of the given VM.

Signature:

```
int get_version (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getVGPUs

Overview:

Get the VGPUs field of the given VM.

Signature:

```
VGPU ref set get_VGPUs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

RPC name: getVIFs

Overview:

Get the VIFs field of the given VM.

Signature:

```
VIF ref set get_VIFs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VIF ref set`

value of the field

RPC name: getVTPMs

Overview:

Get the VTPMs field of the given VM.

Signature:

```
VTPM ref set get_VTPMs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VTPM ref set`

value of the field

RPC name: getVUSBs

Overview:

Get the VUSBs field of the given VM.

Signature:

```
VUSB ref set get_VUSBs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VUSB ref set`

value of the field

RPC name: getxenstoredata

Overview:

Get the xenstoredata field of the given VM.

Signature:

```
(string -> string) map get_xenstore_data (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: hardreboot

Overview:

Stop executing the specified VM without attempting a clean shutdown and immediately restart the VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void hard_reboot (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to reboot

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: hardshutdown

Overview:

Stop executing the specified VM without attempting a clean shutdown.

Signature:

```
void hard_shutdown (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to destroy

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: import

Overview:

Import an XVA from a URI

Signature:

```
VM ref set import (session ref session_ref, string url, SR ref sr, bool full_restore, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`url`	The URL of the XVA file
`SR ref`	`sr`	The destination SR for the disks
`bool`	`full_restore`	Perform a full restore
`bool`	`force`	Force the import

Minimum Role: pool-operator

Return Type: `VM ref set`

Imported VM reference

RPC name: importconvert

Overview:

Import using a conversion service.

Signature:

```
void import_convert (session ref session_ref, string type, string username, string password, SR ref sr, (string -> string) map remote_config)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`type`	Type of the conversion
`string`	`username`	Admin username on the host
`string`	`password`	Password on the host
`SR ref`	`sr`	The destination SR
`(string -> string) map`	`remote_config`	Remote configuration options

Minimum Role: vm-admin

Return Type: `void`

RPC name: maximisememory

Overview:

Returns the maximum amount of guest memory which will fit, together with overheads, in the supplied amount of physical memory. If 'exact' is true then an exact calculation is performed using the VM's current settings. If 'exact' is false then a more conservative approximation is used

Signature:

```
int maximise_memory (session ref session_ref, VM ref self, int total, bool approximate)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`total`	Total amount of physical RAM to fit within

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`approximate`	If false the limit is calculated with the guest's current exact configuration. Otherwise a more approximate calculation is performed

Minimum Role: read-only

Return Type: `int`

The maximum possible staticmax

RPC name: migratesend

Overview:

Migrate the VM to another host. This can only be called when the specified VM is in the Running state.

Signature:

```
VM ref migrate_send (session ref session_ref, VM ref vm, (string ->
string) map dest, bool live, (VDI ref -> SR ref) map vdi_map, (VIF ref ->
network ref) map vif_map, (string -> string) map options, (VGPU ref ->
GPU_group ref) map vgpu_map, (VDI ref -> string) map vdi_format_map)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`(string -> string) map`	`dest`	The result of a Host.migrate_receive call.
`bool`	`live`	Live migration
`(VDI ref -> SR ref) map`	`vdi_map`	Map of source VDI to destination SR

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(VIF ref -> network ref) map`	`vif_map`	Map of source VIF to destination network
`(string -> string) map`	`options`	Other parameters
`(VGPU ref -> GPU_group ref) map`	`vgpu_map`	Map of source vGPU to destination GPU group
`(VDI ref -> string) map`	`vdi_format_map`	Map of source VDI to it's expected type on destination

Minimum Role: vm-power-admin

Return Type: `VM ref`

The reference of the newly created VM in the destination pool

Possible Error Codes: `VM_BAD_POWER_STATE`, `LICENCE_RESTRICTION`

RPC name: pause

Overview:

Pause the specified VM. This can only be called when the specified VM is in the Running state.

Signature:

```
void pause (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to pause

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: poolmigrate

Overview:

Migrate a VM to another Host.

Signature:

```
void pool_migrate (session ref session_ref, VM ref vm, host ref host,
(string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to migrate
`host ref`	`host`	The target host
`(string -> string) map`	`options`	Extra configuration operations: force, live, copy, compress. Each is a boolean option, taking 'true' or 'false' as a value. Option 'compress' controls the use of stream compression during migration.

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `VM_IS_TEMPLATE`, `OPERATION_NOT_ALLOWED`, `VM_BAD_POWER_STATE`

RPC name: powerstatereset

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Reset the powerstate of the VM to halted in the database only. Used to recover from slave failures in pooling scenarios by resetting the powerstates of VMs running on dead slaves to halted. This is a potentially dangerous operation; use with care.

Signature:

```
void power_state_reset (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to reset

Minimum Role: pool-operator

Return Type: `void`

RPC name: provision

Overview:

Inspects the disk configuration contained within the VM's otherconfig, creates VDIs and VBDs and then executes any applicable postinstall script.

Signature:

```
void provision (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be provisioned

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: querydatasource

Overview:

Query the latest value of the specified data source

Signature:

```
float query_data_source (session ref session_ref, VM ref self, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`data_source`	The data source to query

Minimum Role: read-only

Return Type: `float`

The latest value, averaged over the last 5 seconds

RPC name: queryservices

Overview:

Query the system services advertised by this VM and register them. This can only be applied to a system domain.

Signature:

```
(string -> string) map query_services (session ref session_ref, VM ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: pool-admin

Return Type: `(string -> string) map`

map of service type to name

RPC name: recorddatasource

Overview:

Start recording the specified data source

Signature:

```
void record_data_source (session ref session_ref, VM ref self, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`data_source`	The data source to record

Minimum Role: vm-admin

Return Type: `void`

RPC name: recover

Overview:

Recover the VM

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void recover (session ref session_ref, VM ref self, session ref
session_to, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to recover
`session ref`	`session_to`	The session to which the VM is to be recovered.
`bool`	`force`	Whether the VM should replace newer versions of itself.

Minimum Role: read-only

Return Type: `void`

RPC name: removefromblockedoperations

Overview:

Update list of operations which have been explicitly blocked and an error code

Signature:

```
void remove_from_blocked_operations (session ref session_ref, VM ref self,
vm_operations key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`vm_operations`	`key`	Blocked operation

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromHVMbootparams

Overview:

Remove the given key and its corresponding value from the HVM/bootparams field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_HVM_boot_params (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromNVRAM

Overview:

Signature:

```
void remove_from_NVRAM (session ref session_ref, VM ref self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM
`string`	`key`	The key

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to object
`string`	`key`	Key of entry to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromplatform

Overview:

Remove the given key and its corresponding value from the platform field of the given VM. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_platform (session ref session_ref, VM ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to object
`string`	`key`	Key of entry to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromVCPUsparams

Overview:

Remove the given key and its corresponding value from the VCPUs/params field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_VCPUs_params (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removefromxenstoredata

Overview:

Remove the given key and its corresponding value from the xenstoredata field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_xenstore_data (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given VM. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: restartdevicemodels

Overview:

Restart device models of the VM

Signature:

```
void restart_device_models (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `VM_IS_TEMPLATE`, `OPERATION_NOT_ALLOWED`, `VM_BAD_POWER_STATE`

RPC name: resume

Overview:

Awaken the specified VM and resume it. This can only be called when the specified VM is in the Suspended state.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void resume (session ref session_ref, VM ref vm, bool start_paused, bool
force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to resume
`bool`	`start_paused`	Resume VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to resume. If this flag is false then the VM may fail pre-resume safety checks (for example, if the CPU the VM was running on looks substantially different to the current one)

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: resumeon

Overview:

Awaken the specified VM and resume it on a particular Host. This can only be called when the specified VM is in the Suspended state.

Signature:

```
void resume_on (session ref session_ref, VM ref vm, host ref host, bool
start_paused, bool force)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to resume
`host ref`	`host`	The Host on which to resume the VM
`bool`	`start_paused`	Resume VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to resume. If this flag is false then the VM may fail pre-resume safety checks (for example, if the CPU the VM was running on looks substantially different to the current one)

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: retrievewlbrecommendations

Overview:

Returns mapping of hosts to ratings, indicating the suitability of starting the VM at that location according to wlb. Rating is replaced with an error if the VM cannot boot there.

Signature:

```
(host ref -> string set) map retrieve_wlb_recommendations (session ref
session_ref, VM ref vm)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM

Minimum Role: read-only

Return Type: `(host ref -> string set) map`

The potential hosts and their corresponding recommendations or errors

RPC name: revert

Overview:

Reverts the specified VM to a previous state.

Signature:

```
void revert (session ref session_ref, VM ref snapshot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`snapshot`	The snapshotted state that we revert to

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `SR_FULL`, `VM_REVERT_FAILED`

RPC name: sendsysrq

Overview:

Send the given key as a sysrq to this VM. The key is specified as a single character a String of length 1. This can only be called when the specified VM is in the Running state.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void send_sysrq (session ref session_ref, VM ref vm, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`string`	`key`	The key to send

Minimum Role: pool-admin

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`

RPC name: sendtrigger

Overview:

Send the named trigger to this VM. This can only be called when the specified VM is in the Running state.

Signature:

```
void send_trigger (session ref session_ref, VM ref vm, string trigger)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`string`	`trigger`	The trigger to send

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`

RPC name: setactionsaftercrash

Overview:

Sets the actionsaftercrash parameter

Signature:

```
void set_actions_after_crash (session ref session_ref, VM ref self,
on_crash_behaviour value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to set
`on_crash_behaviour`	`value`	The new value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setactionsafterreboot

Overview:

Set the actions/afterreboot field of the given VM.

Signature:

```
void set_actions_after_reboot (session ref session_ref, VM ref self,
on_normal_exit value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`on_normal_exit`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setactionsaftershutdown

Overview:

Set the actions/aftershutdown field of the given VM.

Signature:

```
void set_actions_after_shutdown (session ref session_ref, VM ref self,
on_normal_exit value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`on_normal_exit`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setactionsaftersoftreboot

Overview:

Set the actions/aftersoftreboot field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_actions_after_softreboot (session ref session_ref, VM ref self,
on_softreboot_behavior value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`on_softreboot_behavior`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setaffinity

Overview:

Set the affinity field of the given VM.

Signature:

```
void set_affinity (session ref session_ref, VM ref self, host ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`host ref`	`value`	New value to set

Minimum Role: vm-power-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setappliance

Overview:

Assign this VM to an appliance.

Signature:

```
void set_appliance (session ref session_ref, VM ref self, VM_appliance ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to assign to an appliance.
`VM_appliance ref`	`value`	The appliance to which this VM should be assigned.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbiosstrings

Overview:

Set custom BIOS strings to this VM. VM will be given a default set of BIOS strings, only some of which can be overridden by the supplied values. Allowed keys are: 'biosvendor', 'biosversion', 'systemmanufacturer', 'systemproductname', 'systemversion', 'systemserialnumber', 'enclosureassettag', 'baseboardmanufacturer', 'baseboardproductname', 'baseboardversion', 'baseboardserialnumber', 'baseboardassettag', 'baseboardlocationinchassis', 'enclosureassettag'

Signature:

```
void set_bios_strings (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`(string -> string) map`	`value`	The custom BIOS strings as a list of key-value pairs

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `VM_BIOS_STRINGS_ALREADY_SET`, `INVALID_VALUE`

RPC name: setblockedoperations

Overview:

Update list of operations which have been explicitly blocked and an error code

Signature:

```
void set_blocked_operations (session ref session_ref, VM ref self,
(vm_operations -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`(vm_operations -> string) map`	`value`	Blocked operations

Minimum Role: vm-admin

Return Type: `void`

RPC name: setdomaintype

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the VM.domaintype field of the given VM, which will take effect when it is next started

Signature:

```
void set_domain_type (session ref session_ref, VM ref self, domain_type
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`domain_type`	`value`	The new domain type

Minimum Role: vm-admin

Return Type: `void`

RPC name: setgroups

Overview:

Associate this VM with VM groups.

Signature:

```
void set_groups (session ref session_ref, VM ref self, VM_group ref set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`VM_group ref set`	`value`	The VM groups to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `OPERATION_NOT_ALLOWED`

RPC name: sethaalwaysrun

This message is deprecated.

Overview:

Set the value of the haalwaysrun

Signature:

```
void set_ha_always_run (session ref session_ref, VM ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`bool`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: setharestartpriority

Overview:

Set the value of the harestartpriority field

Signature:

```
void set_ha_restart_priority (session ref session_ref, VM ref self, string
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethardwareplatformversion

Overview:

Set the hardwareplatformversion field of the given VM.

Signature:

```
void set_hardware_platform_version (session ref session_ref, VM ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: sethasvendordevice

Overview:

Controls whether, when the VM starts in HVM mode, its virtual hardware will include the emulated PCI device for which drivers may be available through Windows Update. Usually this should never

© 1997–2026 Citrix Systems, Inc. All rights reserved.

be changed on a VM on which Windows has been installed: changing it on such a VM is likely to lead to a crash on next start.

Signature:

```
void set_has_vendor_device (session ref session_ref, VM ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM on which to set this flag
`bool`	`value`	True to provide the vendor PCI device.

Minimum Role: vm-admin

Return Type: `void`

RPC name: setHVMbootparams

Overview:

Set the HVM/bootparams field of the given VM.

Signature:

```
void set_HVM_boot_params (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setHVMbootpolicy

This message is deprecated.

Overview:

Set the VM.HVMbootpolicy field of the given VM, which will take effect when it is next started

Signature:

```
void set_HVM_boot_policy (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`value`	The new HVM boot policy

Minimum Role: vm-admin

Return Type: `void`

RPC name: setHVMshadowmultiplier

Overview:

Set the shadow memory multiplier on a halted VM

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_HVM_shadow_multiplier (session ref session_ref, VM ref self,
float value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`float`	`value`	The new shadow memory multiplier to set

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setisatemplate

Overview:

Set the isatemplate field of the given VM.

Signature:

```
void set_is_a_template (session ref session_ref, VM ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setmemory

Overview:

Set the memory allocation of this VM. Sets all of memorystaticmax, memorydynamicmin, and memorydynamicmax to the given value, and leaves memorystaticmin untouched.

Signature:

```
void set_memory (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	The new memory allocation (bytes).

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorydynamicmax

Overview:

Set the value of the memorydynamicmax field

Signature:

```
void set_memory_dynamic_max (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_dynamic_max

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorydynamicmin

Overview:

Set the value of the memorydynamicmin field

Signature:

```
void set_memory_dynamic_min (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_dynamic_min

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorydynamicrange

Overview:

Set the minimum and maximum amounts of physical memory the VM is allowed to use.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_memory_dynamic_range (session ref session_ref, VM ref self, int
min, int max)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`min`	The new minimum value
`int`	`max`	The new maximum value

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorylimits

Overview:

Set the memory limits of this VM.

Signature:

```
void set_memory_limits (session ref session_ref, VM ref self, int
static_min, int static_max, int dynamic_min, int dynamic_max)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`static_min`	The new value of memory_static_min.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`int`	`static_max`	The new value of memory_static_max.
`int`	`dynamic_min`	The new value of memory_dynamic_min.
`int`	`dynamic_max`	The new value of memory_dynamic_max.

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorystaticmax

Overview:

Set the value of the memorystaticmax field

Signature:

```
void set_memory_static_max (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_static_max

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setmemorystaticmin

Overview:

Set the value of the memorystaticmin field

Signature:

```
void set_memory_static_min (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_static_min

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorystaticrange

Overview:

Set the static ie boottime range of virtual memory that the VM is allowed to use.

Signature:

```
void set_memory_static_range (session ref session_ref, VM ref self, int
min, int max)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM
`int`	`min`	The new minimum value
`int`	`max`	The new maximum value

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorytargetlive

This message is deprecated.

Overview:

Set the memory target for a running VM

Signature:

```
void set_memory_target_live (session ref session_ref, VM ref self, int
target)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`target`	The target in bytes

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_name_description (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VM.

Signature:

```
void set_name_label (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setNVRAM

Overview:

Signature:

```
void set_NVRAM (session ref session_ref, VM ref self, (string -> string)
map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`(string -> string) map`	`value`	The value

Minimum Role: vm-admin

Return Type: `void`

RPC name: setorder

Overview:

Set this VM's boot order

Signature:

```
void set_order (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	This VM's boot order

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VM.

Signature:

```
void set_other_config (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPCIbus

This message is deprecated.

Overview:

Set the PCIbus field of the given VM.

Signature:

```
void set_PCI_bus (session ref session_ref, VM ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setplatform

Overview:

Set the platform field of the given VM.

Signature:

```
void set_platform (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setprotectionpolicy

This message is removed.

Overview:

Set the value of the protectionpolicy field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_protection_policy (session ref session_ref, VM ref self, VMPP ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`VMPP ref`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: setPVargs

Overview:

Set the PV/args field of the given VM.

Signature:

```
void set_PV_args (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setPVbootloader

Overview:

Set the PV/bootloader field of the given VM.

Signature:

```
void set_PV_bootloader (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVbootloaderargs

Overview:

Set the PV/bootloaderargs field of the given VM.

Signature:

```
void set_PV_bootloader_args (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVkernel

Overview:

Set the PV/kernel field of the given VM.

Signature:

```
void set_PV_kernel (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVlegacyargs

Overview:

Set the PV/legacyargs field of the given VM.

Signature:

```
void set_PV_legacy_args (session ref session_ref, VM ref self, string
value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVramdisk

Overview:

Set the PV/ramdisk field of the given VM.

Signature:

```
void set_PV_ramdisk (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setrecommendations

Overview:

Set the recommendations field of the given VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_recommendations (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setshadowmultiplierlive

Overview:

Set the shadow memory multiplier on a running VM

Signature:

```
void set_shadow_multiplier_live (session ref session_ref, VM ref self,
float multiplier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`float`	`multiplier`	The new shadow memory multiplier to set

Minimum Role: vm-power-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setshutdowndelay

Overview:

Set this VM's shutdown delay in seconds

Signature:

```
void set_shutdown_delay (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	This VM's shutdown delay in seconds

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsnapshotschedule

Overview:

Set the value of the snapshot schedule field

Signature:

```
void set_snapshot_schedule (session ref session_ref, VM ref self, VMSS ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM
`VMSS ref`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: setstartdelay

Overview:

Set this VM's start delay in seconds

Signature:

```
void set_start_delay (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	This VM's start delay in seconds

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsuspendSR

Overview:

Set the suspendSR field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_suspend_SR (session ref session_ref, VM ref self, SR ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setsuspendVDI

Overview:

Set this VM's suspend VDI, which must be indentical to its current one

Signature:

```
void set_suspend_VDI (session ref session_ref, VM ref self, VDI ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`VDI ref`	`value`	The suspend VDI uuid

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the tags field of the given VM.

Signature:

```
void set_tags (session ref session_ref, VM ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: setuefimode

Overview:

Set the UEFI mode of a VM

Signature:

```
string set_uefi_mode (session ref session_ref, VM ref self, vm_uefi_mode
mode)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`vm_uefi_mode`	`mode`	The UEFI mode to set

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

Result from the varstoresbstate call

RPC name: setuserversion

Overview:

Set the userversion field of the given VM.

Signature:

```
void set_user_version (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setVCPUsatstartup

Overview:

Set the number of startup VCPUs for a halted VM

Signature:

```
void set_VCPUs_at_startup (session ref session_ref, VM ref self, int
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	The new maximum number of VCPUs

Minimum Role: vm-admin

Return Type: `void`

RPC name: setVCPUsmax

Overview:

Set the maximum number of VCPUs for a halted VM

Signature:

```
void set_VCPUs_max (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	The new maximum number of VCPUs

Minimum Role: vm-admin

Return Type: `void`

RPC name: setVCPUsnumberlive

Overview:

Set the number of VCPUs for a running VM

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_VCPUs_number_live (session ref session_ref, VM ref self, int
nvcpu)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`nvcpu`	The number of VCPUs

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: setVCPUsparams

Overview:

Set the VCPUs/params field of the given VM.

Signature:

```
void set_VCPUs_params (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: setxenstoredata

Overview:

Set the xenstoredata field of the given VM.

Signature:

```
void set_xenstore_data (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: shutdown

Overview:

Attempts to first clean shutdown a VM and if it should fail then perform a hard shutdown on it.

Signature:

```
void shutdown (session ref session_ref, VM ref vm)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to shutdown

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: snapshot

Overview:

Snapshots the specified VM, making a new VM. Snapshot automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write.

Signature:

```
VM ref snapshot (session ref session_ref, VM ref vm, string new_name, VDI ref set ignore_vdis)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be snapshotted
`string`	`new_name`	The name of the snapshotted VM
`VDI ref set`	`ignore_vdis`	A list of VDIs to ignore for the snapshot

Minimum Role: vm-power-admin

Return Type: `VM ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`

RPC name: snapshotwithquiesce

This message is removed.

Overview:

Snapshots the specified VM with quiesce, making a new VM. Snapshot automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write.

Signature:

```
VM ref snapshot_with_quiesce (session ref session_ref, VM ref vm, string new_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be snapshotted
`string`	`new_name`	The name of the snapshotted VM

Minimum Role: vm-power-admin

Return Type: `VM ref`

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `VM_SNAPSHOT_WITH_QUIESCE_FAILED`, `VM_SNAPSHOT_WITH_QUIESCE_TIMEOUT`, `VM_SNAPSHOT_WITH_QUIESCE_PLUGIN_DEOS_NOT_RESPOND`, `VM_SNAPSHOT_WITH_QUIESCE_NOT_SUPPORTED`

RPC name: start

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Start the specified VM. This function can only be called with the VM is in the Halted State.

Signature:

```
void start (session ref session_ref, VM ref vm, bool start_paused, bool
force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to start
`bool`	`start_paused`	Instantiate VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to start. If this flag is false then the VM may fail pre-boot safety checks (for example, if the CPU the VM last booted on looks substantially different to the current one)

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `VM_HVM_REQUIRED`, `VM_IS_TEMPLATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `BOOTLOADER_FAILED`, `UNKNOWN_BOOTLOADER`, `NO_HOSTS_AVAILABLE`, `LICENCE_RESTRICTION`

RPC name: starton

Overview:

Start the specified VM on a particular host. This function can only be called with the VM is in the Halted State.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void start_on (session ref session_ref, VM ref vm, host ref host, bool
start_paused, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to start
`host ref`	`host`	The Host on which to start the VM
`bool`	`start_paused`	Instantiate VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to start. If this flag is false then the VM may fail pre-boot safety checks (for example, if the CPU the VM last booted on looks substantially different to the current one)

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `VM_IS_TEMPLATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `BOOTLOADER_FAILED`, `UNKNOWN_BOOTLOADER`

RPC name: suspend

Overview:

Suspend the specified VM to disk. This can only be called when the specified VM is in the Running state.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void suspend (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to suspend

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: sysprep

Overview:

Pass unattend.xml to Windows sysprep and wait for the VM to shut down as part of a reboot.

Signature:

```
void sysprep (session ref session_ref, VM ref self, string unattend, float
timeout)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`unattend`	XML content passed to sysprep
`float`	`timeout`	timeout in seconds for expected reboot

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: unpause

Overview:

Resume the specified VM. This can only be called when the specified VM is in the Paused state.

Signature:

```
void unpause (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to unpause

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: updateallowedoperations

Overview:

Recomputes the list of acceptable operations

Signature:

```
void update_allowed_operations (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: updatesecurebootcertificatesonboot

Overview:

Mark or unmark secure boot certificate update on VM boot

Signature:

```
void update_secureboot_certificates_on_boot (session ref session_ref, VM ref self, bool mark)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`bool`	`mark`	If true: mark certificates for update on next boot. If false: remove the mark

Minimum Role: vm-admin

Return Type: `void`

RPC name: waitmemorytargetlive

This message is deprecated.

Overview:

Wait for a running VM to reach its current memory target

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void wait_memory_target_live (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: read-only

Return Type: `void`

Class: VMappliance

VM appliance

Fields for class: VMappliance

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vm_appliance_operation set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`current_operations`	`(string -> vm_appliance_operation) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VMs`	`VM ref set`	*RO/runtime*	all VMs in this appliance

RPCs associated with class: VMappliance

RPC name: assertcanberecovered

Overview:

Assert whether all SRs required to recover this VM appliance are available.

Signature:

```
void assert_can_be_recovered (session ref session_ref, VM_appliance ref self, session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance to recover
`session ref`	`session_to`	The session to which the VM appliance is to be recovered.

Minimum Role: read-only

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `VM_REQUIRES_SR`

RPC name: cleanshutdown

Overview:

Perform a clean shutdown of all the VMs in the appliance

Signature:

```
void clean_shutdown (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

RPC name: create

Overview:

Create a new VMappliance instance, and return its handle.

Signature:

```
VM_appliance ref create (session ref session_ref, VM_appliance record
args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_appliance record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `VM_appliance ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VMappliance instance.

Signature:

```
void destroy (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMappliances known to the system.

Signature:

```
VM_appliance ref set get_all (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_appliance ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMappliance references to VMappliance records for all VMappliances known to the system.

Signature:

```
(VM_appliance ref -> VM_appliance record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM_appliance ref -> VM_appliance record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VMappliance.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
vm_appliance_operation set get_allowed_operations (session ref
session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_appliance_operation set`

value of the field

RPC name: getbynamelabel

Overview:

Get all the VMappliance instances with the given label.

Signature:

```
VM_appliance ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VM_appliance ref set`

references to objects with matching names

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbyuuid

Overview:

Get a reference to the VMappliance instance with the specified UUID.

Signature:

```
VM_appliance ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_appliance ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VMappliance.

Signature:

```
(string -> vm_appliance_operation) map get_current_operations (session ref
session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> vm_appliance_operation) map`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given VMappliance.

Signature:

```
string get_name_description (session ref session_ref, VM_appliance ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VMappliance.

Signature:

```
string get_name_label (session ref session_ref, VM_appliance ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMappliance.

Signature:

```
VM_appliance record get_record (session ref session_ref, VM_appliance ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_appliance record`

all fields from the object

RPC name: getSRsrequiredforrecovery

Overview:

Get the list of SRs required by the VM appliance to recover.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR ref set get_SRs_required_for_recovery (session ref session_ref,
VM_appliance ref self, session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance for which the required list of SRs has to be recovered.
`session ref`	`session_to`	The session to which the list of SRs have to be recovered .

Minimum Role: read-only

Return Type: `SR ref set`

refs for SRs required to recover the VM

RPC name: getuuid

Overview:

Get the uuid field of the given VMappliance.

Signature:

```
string get_uuid (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getVMs

Overview:

Get the VMs field of the given VMappliance.

Signature:

```
VM ref set get_VMs (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: hardshutdown

Overview:

Perform a hard shutdown of all the VMs in the appliance

Signature:

```
void hard_shutdown (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_appliance ref`	`self`	The VM appliance

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

RPC name: recover

Overview:

Recover the VM appliance

Signature:

```
void recover (session ref session_ref, VM_appliance ref self, session ref session_to, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance to recover
`session ref`	`session_to`	The session to which the VM appliance is to be recovered.
`bool`	`force`	Whether the VMs should replace newer versions of themselves.

Minimum Role: read-only

Return Type: `void`

Possible Error Codes: `VM_REQUIRES_SR`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setnamedescription

Overview:

Set the name/description field of the given VMappliance.

Signature:

```
void set_name_description (session ref session_ref, VM_appliance ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VMappliance.

Signature:

```
void set_name_label (session ref session_ref, VM_appliance ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: shutdown

Overview:

For each VM in the appliance, try to shut it down cleanly. If this fails, perform a hard shutdown of the VM.

Signature:

```
void shutdown (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

RPC name: start

Overview:

Start all VMs in the appliance

Signature:

```
void start (session ref session_ref, VM_appliance ref self, bool paused)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance
`bool`	`paused`	Instantiate all VMs belonging to this appliance in paused state if set to true.

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

Class: VMgroup

A VM group

Fields for class: VMgroup

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`placement`	`placement_policy`	*RO/constructor*	The placement policy of the VM group
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VMs`	`VM ref set`	*RO/runtime*	The list of VMs associated with the group

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VMgroup

RPC name: create

Overview:

Create a new VMgroup instance, and return its handle.

Signature:

```
VM_group ref create (session ref session_ref, VM_group record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VM_group ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VMgroup instance.

Signature:

```
void destroy (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMgroups known to the system.

Signature:

```
VM_group ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_group ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMgroup references to VMgroup records for all VMgroups known to the system.

Signature:

```
(VM_group ref -> VM_group record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VM_group ref -> VM_group record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the VMgroup instances with the given label.

Signature:

```
VM_group ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VM_group ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VMgroup instance with the specified UUID.

Signature:

```
VM_group ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_group ref`

reference to the object

RPC name: getnamedescription

Overview:

Get the name/description field of the given VMgroup.

Signature:

```
string get_name_description (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VMgroup.

Signature:

```
string get_name_label (session ref session_ref, VM_group ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getplacement

Overview:

Get the placement field of the given VMgroup.

Signature:

```
placement_policy get_placement (session ref session_ref, VM_group ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `placement_policy`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMgroup.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM_group record get_record (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_group record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given VMgroup.

Signature:

```
string get_uuid (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getVMs

Overview:

Get the VMs field of the given VMgroup.

Signature:

```
VM ref set get_VMs (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: setnamedescription

Overview:

Set the name/description field of the given VMgroup.

Signature:

```
void set_name_description (session ref session_ref, VM_group ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VMgroup.

Signature:

```
void set_name_label (session ref session_ref, VM_group ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VMguestmetrics

The metrics reported by the guest as opposed to inferred from outside

Fields for class: VMguestmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`can_use_hotplug_vbd`	`tristate_type`	*RO/runtime*	The guest's statement of whether

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			it supports VBD hotplug, i.e. whether it is capable of responding immediately to instantiation of a new VBD by bringing online a new PV block device. If the guest states that it is not capable, then the VBD plug and unplug operations will not be allowed while the guest is running.
`can_use_hotplug_vif`	`tristate_type`	*RO/runtime*	The guest's statement of whether it supports VIF hotplug, i.e. whether it is capable of responding immediately to instantiation of a new VIF by bringing online a new PV network device. If the guest states that it is not capable, then the VIF plug and unplug operations will not be allowed while the guest is running.
`disks`	`(string -> string) map`	*RO/runtime*	**Removed.** This field exists but has no data.
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`live`	`bool`	*RO/runtime*	True if the guest is sending heartbeat messages via the guest agent
`memory`	`(string -> string) map`	*RO/runtime*	**Removed.** This field exists but has no data. Use the memory and memory_internal_free RRD data-sources instead.
`netbios_name`	`(string -> string) map`	*RO/runtime*	The NETBIOS name of the machine
`networks`	`(string -> string) map`	*RO/runtime*	network configuration
`os_version`	`(string -> string) map`	*RO/runtime*	version of the OS
`other`	`(string -> string) map`	*RO/runtime*	anything else
`other_config`	`(string -> string) map`	*RW*	additional configuration
`PV_drivers_detected`	`bool`	*RO/runtime*	At least one of the guest's devices has successfully connected to the backend.
`PV_drivers_up_to_date`	`bool`	*RO/runtime*	**Deprecated.** Logically equivalent to PV_drivers_detected
`PV_drivers_version`	`(string -> string) map`	*RO/runtime*	version of the PV drivers

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`services`	`(string -> string) map`	*RO/runtime*	The guest's services data.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: VMguestmetrics

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VMguestmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VM_guest_metrics ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMguestmetrics instances known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_guest_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_guest_metrics ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMguestmetrics references to VMguestmetrics records for all VMguestmetrics instances known to the system.

Signature:

```
(VM_guest_metrics ref -> VM_guest_metrics record) map get_all_records
(session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM_guest_metrics ref -> VM_guest_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VMguestmetrics instance with the specified UUID.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM_guest_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_guest_metrics ref`

reference to the object

RPC name: getcanusehotplugvbd

Overview:

Get the canusehotplugvbd field of the given VMguestmetrics.

Signature:

```
tristate_type get_can_use_hotplug_vbd (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tristate_type`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getcanusehotplugvif

Overview:

Get the canusehotplugvif field of the given VMguestmetrics.

Signature:

```
tristate_type get_can_use_hotplug_vif (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tristate_type`

value of the field

RPC name: getdisks

This message is removed.

Overview:

Get the disks field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_disks (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given VMguestmetrics.

Signature:

```
datetime get_last_updated (session ref session_ref, VM_guest_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlive

Overview:

Get the live field of the given VMguestmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_live (session ref session_ref, VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmemory

This message is removed.

Overview:

Get the memory field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_memory (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnetbiosname

Overview:

Get the netbiosname field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_netbios_name (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getnetworks

Overview:

Get the networks field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_networks (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getosversion

Overview:

Get the osversion field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_os_version (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getother

Overview:

Get the other field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_other (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPVdriversdetected

Overview:

Get the PVdriversdetected field of the given VMguestmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_PV_drivers_detected (session ref session_ref, VM_guest_metrics
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getPVdriversuptodate

This message is deprecated.

Overview:

Get the PVdriversuptodate field of the given VMguestmetrics.

Signature:

```
bool get_PV_drivers_up_to_date (session ref session_ref, VM_guest_metrics
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getPVdriversversion

Overview:

Get the PVdriversversion field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_PV_drivers_version (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMguestmetrics.

Signature:

```
VM_guest_metrics record get_record (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VM_guest_metrics record`

all fields from the object

RPC name: getservices

Overview:

Get the services field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_services (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VMguestmetrics.

Signature:

```
string get_uuid (session ref session_ref, VM_guest_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VMguestmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VM_guest_metrics ref self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VMguestmetrics.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_other_config (session ref session_ref, VM_guest_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VMmetrics

The metrics associated with a VM

Fields for class: VMmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`current_domain_type`	`domain_type`	*RO/runtime*	The current domain type of the VM (for running,suspended, or paused VMs). The last-known domain type for halted VMs.
`hvm`	`bool`	*RO/runtime*	hardware virtual machine
`install_time`	`datetime`	*RO/runtime*	Time at which the VM was installed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated
`memory_actual`	`int`	*RO/runtime*	Guest's actual memory (bytes)
`nested_virt`	`bool`	*RO/runtime*	VM supports nested virtualisation
`nomigrate`	`bool`	*RO/runtime*	VM is immobile and can't migrate between hosts
`numa_node_memory`	`(int -> int) map`	*RO/runtime*	mapping a NUMA node (int) to an amount of memory (bytes) in that node.
`numa_nodes`	`int`	*RO/constructor*	number of NUMA nodes of the host the VM is using
`numa_optimised`	`bool`	*RO/constructor*	If the VM is optimised for NUMA
`other_config`	`(string -> string) map`	*RW*	additional configuration
`start_time`	`datetime`	*RO/runtime*	Time at which this VM was last booted
`state`	`string set`	*RO/runtime*	The state of the guest, eg blocked, dying etc
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VCPUs_CPU`	`(int -> int) map`	*RO/runtime*	VCPU to PCPU map

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VCPUs_flags`	`(int -> string set) map`	*RO/runtime*	CPU flags (blocked,online,running)
`VCPUs_number`	`int`	*RO/runtime*	Current number of VCPUs
`VCPUs_params`	`(string -> string) map`	*RO/runtime*	The live equivalent to VM.VCPUs_params
`VCPUs_utilisation`	`(int -> float) map`	*RO/runtime*	**Removed.** Utilisation for all of guest's current VCPUs

RPCs associated with class: VMmetrics

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VMmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VM_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMmetrics instances known to the system.

Signature:

```
VM_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_metrics ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMmetrics references to VMmetrics records for all VMmetrics instances known to the system.

Signature:

```
(VM_metrics ref -> VM_metrics record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VM_metrics ref -> VM_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VMmetrics instance with the specified UUID.

Signature:

```
VM_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_metrics ref`

reference to the object

RPC name: getcurrentdomaintype

Overview:

Get the currentdomaintype field of the given VMmetrics.

Signature:

```
domain_type get_current_domain_type (session ref session_ref, VM_metrics
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `domain_type`

value of the field

RPC name: gethvm

Overview:

Get the hvm field of the given VMmetrics.

Signature:

```
bool get_hvm (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getinstalltime

Overview:

Get the installtime field of the given VMmetrics.

Signature:

```
datetime get_install_time (session ref session_ref, VM_metrics ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given VMmetrics.

Signature:

```
datetime get_last_updated (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getmemoryactual

Overview:

Get the memory/actual field of the given VMmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int get_memory_actual (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnestedvirt

Overview:

Get the nestedvirt field of the given VMmetrics.

Signature:

```
bool get_nested_virt (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnomigrate

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the nomigrate field of the given VMmetrics.

Signature:

```
bool get_nomigrate (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnumanodememory

Overview:

Get the numanodememory field of the given VMmetrics.

Signature:

```
(int -> int) map get_numa_node_memory (session ref session_ref, VM_metrics
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(int -> int) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getnumanodes

Overview:

Get the numanodes field of the given VMmetrics.

Signature:

```
int get_numa_nodes (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnumaoptimised

Overview:

Get the numaoptimised field of the given VMmetrics.

Signature:

```
bool get_numa_optimised (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VMmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMmetrics.

Signature:

```
VM_metrics record get_record (session ref session_ref, VM_metrics ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_metrics record`

all fields from the object

RPC name: getstarttime

Overview:

Get the starttime field of the given VMmetrics.

Signature:

```
datetime get_start_time (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getstate

Overview:

Get the state field of the given VMmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set get_state (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VMmetrics.

Signature:

```
string get_uuid (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVCPUsCPU

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the VCPUs/CPU field of the given VMmetrics.

Signature:

```
(int -> int) map get_VCPUs_CPU (session ref session_ref, VM_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(int -> int) map`

value of the field

RPC name: getVCPUsflags

Overview:

Get the VCPUs/flags field of the given VMmetrics.

Signature:

```
(int -> string set) map get_VCPUs_flags (session ref session_ref,
VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(int -> string set) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getVCPUsnumber

Overview:

Get the VCPUs/number field of the given VMmetrics.

Signature:

```
int get_VCPUs_number (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVCPUsparams

Overview:

Get the VCPUs/params field of the given VMmetrics.

Signature:

```
(string -> string) map get_VCPUs_params (session ref session_ref,
VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getVCPUsutilisation

This message is removed.

Overview:

Get the VCPUs/utilisation field of the given VMmetrics.

Signature:

```
(int -> float) map get_VCPUs_utilisation (session ref session_ref,
VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(int -> float) map`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VMmetrics. If the key is not in that Map, then do nothing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void remove_from_other_config (session ref session_ref, VM_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VMmetrics.

Signature:

```
void set_other_config (session ref session_ref, VM_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Class: VMPP

This class is removed.

VM Protection Policy

Fields for class: VMPP

FIELD	TYPE	QUALIFIER	DESCRIPTION
`alarm_config`	`(string -> string) map`	*RO/constructor*	**Removed.** configuration for the alarm
`archive_frequency`	`vmpp_archive_frequency`	*RO/constructor*	**Removed.** frequency of the archive schedule
`archive_last_run_time`	`datetime`	*RO/runtime*	**Removed.** time of the last archive
`archive_schedule`	`(string -> string) map`	*RO/constructor*	**Removed.** schedule of the archive containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone
`archive_target_config`	`(string -> string) map`	*RO/constructor*	**Removed.** configuration for the archive, including its 'location', 'username', 'password'
`archive_target_type`	`vmpp_archive_target_type`	*RO/constructor*	**Removed.** type of the archive target config

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`backup_frequency`	`vmpp_backup_frequency`	*RO/constructor*	**Removed.** frequency of the backup schedule
`backup_last_run_time`	`datetime`	*RO/runtime*	**Removed.** time of the last backup
`backup_retention_value`	`int`	*RO/constructor*	**Removed.** maximum number of backups that should be stored at any time
`backup_schedule`	`(string -> string) map`	*RO/constructor*	**Removed.** schedule of the backup containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone
`backup_type`	`vmpp_backup_type`	*RW*	**Removed.** type of the backup sub-policy
`is_alarm_enabled`	`bool`	*RO/constructor*	**Removed.** true if alarm is enabled for this policy
`is_archive_running`	`bool`	*RO/runtime*	**Removed.** true if this protection policy's archive is running
`is_backup_running`	`bool`	*RO/runtime*	**Removed.** true if this protection policy's backup is running
`is_policy_enabled`	`bool`	*RW*	**Removed.** enable or disable this policy
`name_description`	`string`	*RW*	**Removed.** a notes field containing

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			human-readable description
`name_label`	`string`	*RW*	**Removed.** a human-readable name
`recent_alerts`	`string set`	*RO/runtime*	**Removed.** recent alerts
`uuid`	`string`	*RO/runtime*	**Removed.** Unique identifier/object reference
`VMs`	`VM ref set`	*RO/runtime*	**Removed.** all VMs attached to this protection policy

RPCs associated with class: VMPP

RPC name: addtoalarmconfig

This message is removed.

Overview:

Signature:

```
void add_to_alarm_config (session ref session_ref, VMPP ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add
`string`	`value`	the value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtoarchiveschedule

This message is removed.

Overview:

Signature:

```
void add_to_archive_schedule (session ref session_ref, VMPP ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtoarchivetargetconfig

This message is removed.

Overview:

Signature:

```
void add_to_archive_target_config (session ref session_ref, VMPP ref self,
string key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtobackupschedule

This message is removed.

Overview:

Signature:

```
void add_to_backup_schedule (session ref session_ref, VMPP ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: archivenow

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

This call archives the snapshot provided as a parameter

Signature:

```
string archive_now (session ref session_ref, VM ref snapshot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`snapshot`	The snapshot to archive

Minimum Role: vm-power-admin

Return Type: `string`

An XMLRPC result

RPC name: create

This message is removed.

Overview:

Create a new VMPP instance, and return its handle.

Signature:

```
VMPP ref create (session ref session_ref, VMPP record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP record`	`args`	All constructor arguments

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VMPP ref`

reference to the newly created object

RPC name: destroy

This message is removed.

Overview:

Destroy the specified VMPP instance.

Signature:

```
void destroy (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getalarmconfig

This message is removed.

Overview:

Get the alarmconfig field of the given VMPP.

Signature:

```
(string -> string) map get_alarm_config (session ref session_ref, VMPP ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getalerts

This message is removed.

Overview:

This call fetches a history of alerts for a given protection policy

Signature:

```
string set get_alerts (session ref session_ref, VMPP ref vmpp, int
hours_from_now)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`vmpp`	The protection policy
`int`	`hours_from_now`	how many hours in the past the oldest record to fetch is

Minimum Role: pool-operator

Return Type: `string set`

A list of alerts encoded in xml

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getall

This message is removed.

Overview:

Return a list of all the VMPPs known to the system.

Signature:

```
VMPP ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VMPP ref set`

references to all objects

RPC name: getallrecords

This message is removed.

Overview:

Return a map of VMPP references to VMPP records for all VMPPs known to the system.

Signature:

```
(VMPP ref -> VMPP record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VMPP ref -> VMPP record) map`

records of all objects

RPC name: getarchivefrequency

This message is removed.

Overview:

Get the archivefrequency field of the given VMPP.

Signature:

```
vmpp_archive_frequency get_archive_frequency (session ref session_ref,
VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmpp_archive_frequency`

value of the field

RPC name: getarchivelastruntime

This message is removed.

Overview:

Get the archivelastruntime field of the given VMPP.

Signature:

```
datetime get_archive_last_run_time (session ref session_ref, VMPP ref
self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getarchiveschedule

This message is removed.

Overview:

Get the archiveschedule field of the given VMPP.

Signature:

```
(string -> string) map get_archive_schedule (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getarchivetargetconfig

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Get the archivetargetconfig field of the given VMPP.

Signature:

```
(string -> string) map get_archive_target_config (session ref session_ref,
VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getarchivetargettype

This message is removed.

Overview:

Get the archivetargettype field of the given VMPP.

Signature:

```
vmpp_archive_target_type get_archive_target_type (session ref session_ref,
VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `vmpp_archive_target_type`

value of the field

RPC name: getbackupfrequency

This message is removed.

Overview:

Get the backupfrequency field of the given VMPP.

Signature:

```
vmpp_backup_frequency get_backup_frequency (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmpp_backup_frequency`

value of the field

RPC name: getbackuplastruntime

This message is removed.

Overview:

Get the backuplastruntime field of the given VMPP.

Signature:

```
datetime get_backup_last_run_time (session ref session_ref, VMPP ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getbackupretentionvalue

This message is removed.

Overview:

Get the backupretentionvalue field of the given VMPP.

Signature:

```
int get_backup_retention_value (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getbackupschedule

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the backupschedule field of the given VMPP.

Signature:

```
(string -> string) map get_backup_schedule (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getbackuptype

This message is removed.

Overview:

Get the backuptype field of the given VMPP.

Signature:

```
vmpp_backup_type get_backup_type (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `vmpp_backup_type`

value of the field

RPC name: getbynamelabel

This message is removed.

Overview:

Get all the VMPP instances with the given label.

Signature:

```
VMPP ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VMPP ref set`

references to objects with matching names

RPC name: getbyuuid

This message is removed.

Overview:

Get a reference to the VMPP instance with the specified UUID.

Signature:

```
VMPP ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VMPP ref`

reference to the object

RPC name: getisalarmenabled

This message is removed.

Overview:

Get the isalarmenabled field of the given VMPP.

Signature:

```
bool get_is_alarm_enabled (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisarchiverunning

This message is removed.

Overview:

Get the isarchiverunning field of the given VMPP.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
bool get_is_archive_running (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisbackuprunning

This message is removed.

Overview:

Get the isbackuprunning field of the given VMPP.

Signature:

```
bool get_is_backup_running (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getispolicyenabled

This message is removed.

Overview:

Get the ispolicyenabled field of the given VMPP.

Signature:

```
bool get_is_policy_enabled (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

This message is removed.

Overview:

Get the name/description field of the given VMPP.

Signature:

```
string get_name_description (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

This message is removed.

Overview:

Get the name/label field of the given VMPP.

Signature:

```
string get_name_label (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecentalerts

This message is removed.

Overview:

Get the recentalerts field of the given VMPP.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set get_recent_alerts (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

This message is removed.

Overview:

Get a record containing the current state of the given VMPP.

Signature:

```
VMPP record get_record (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VMPP record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

This message is removed.

Overview:

Get the uuid field of the given VMPP.

Signature:

```
string get_uuid (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVMs

This message is removed.

Overview:

Get the VMs field of the given VMPP.

Signature:

```
VM ref set get_VMs (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: protectnow

This message is removed.

Overview:

This call executes the protection policy immediately

Signature:

```
string protect_now (session ref session_ref, VMPP ref vmpp)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`vmpp`	The protection policy to execute

Minimum Role: pool-operator

Return Type: `string`

An XMLRPC result

RPC name: removefromalarmconfig

This message is removed.

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_alarm_config (session ref session_ref, VMPP ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromarchiveschedule

This message is removed.

Overview:

Signature:

```
void remove_from_archive_schedule (session ref session_ref, VMPP ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removefromarchivetargetconfig

This message is removed.

Overview:

Signature:

```
void remove_from_archive_target_config (session ref session_ref, VMPP ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefrombackupschedule

This message is removed.

Overview:

Signature:

```
void remove_from_backup_schedule (session ref session_ref, VMPP ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setalarmconfig

This message is removed.

Overview:

Signature:

```
void set_alarm_config (session ref session_ref, VMPP ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setarchivefrequency

This message is removed.

Overview:

Set the value of the archivefrequency field

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_archive_frequency (session ref session_ref, VMPP ref self,
vmpp_archive_frequency value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`vmpp_archive_frequency`	`value`	the archive frequency

Minimum Role: pool-operator

Return Type: `void`

RPC name: setarchivelastruntime

This message is removed.

Overview:

Signature:

```
void set_archive_last_run_time (session ref session_ref, VMPP ref self,
datetime value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`datetime`	`value`	The time at which the last archive was created. When the timezone is missing, UTC is assumed

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setarchiveschedule

This message is removed.

Overview:

Signature:

```
void set_archive_schedule (session ref session_ref, VMPP ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setarchivetargetconfig

This message is removed.

Overview:

Signature:

```
void set_archive_target_config (session ref session_ref, VMPP ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setarchivetargettype

This message is removed.

Overview:

Set the value of the archivetargetconfigtype field

Signature:

```
void set_archive_target_type (session ref session_ref, VMPP ref self,
vmpp_archive_target_type value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`vmpp_archive_target_type`	`value`	the archive target config type

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbackupfrequency

This message is removed.

Overview:

Set the value of the backupfrequency field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_backup_frequency (session ref session_ref, VMPP ref self,
vmpp_backup_frequency value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`vmpp_backup_frequency`	`value`	the backup frequency

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbackuplastruntime

This message is removed.

Overview:

Signature:

```
void set_backup_last_run_time (session ref session_ref, VMPP ref self,
datetime value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`datetime`	`value`	The time at which the last backup was done. When the timezone is missing, UTC is assumed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setbackupretentionvalue

This message is removed.

Overview:

Signature:

```
void set_backup_retention_value (session ref session_ref, VMPP ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`int`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbackupschedule

This message is removed.

Overview:

Signature:

```
void set_backup_schedule (session ref session_ref, VMPP ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMPP ref`	`self`	The protection policy
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbackuptype

This message is removed.

Overview:

Set the backuptype field of the given VMPP.

Signature:

```
void set_backup_type (session ref session_ref, VMPP ref self,
vmpp_backup_type value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object
`vmpp_backup_type`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setisalarmenabled

This message is removed.

Overview:

Set the value of the isalarmenabled field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_is_alarm_enabled (session ref session_ref, VMPP ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`bool`	`value`	true if alarm is enabled for this policy

Minimum Role: pool-operator

Return Type: `void`

RPC name: setispolicyenabled

This message is removed.

Overview:

Set the ispolicyenabled field of the given VMPP.

Signature:

```
void set_is_policy_enabled (session ref session_ref, VMPP ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object
`bool`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

This message is removed.

Overview:

Set the name/description field of the given VMPP.

Signature:

```
void set_name_description (session ref session_ref, VMPP ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

This message is removed.

Overview:

Set the name/label field of the given VMPP.

Signature:

```
void set_name_label (session ref session_ref, VMPP ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: VMSS

VM Snapshot Schedule

Fields for class: VMSS

FIELD	TYPE	QUALIFIER	DESCRIPTION
`enabled`	`bool`	*RW*	enable or disable this snapshot schedule
`frequency`	`vmss_frequency`	*RO/constructor*	frequency of taking snapshot from snapshot schedule
`last_run_time`	`datetime`	*RO/runtime*	time of the last snapshot
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`retained_snapshots`	`int`	*RO/constructor*	maximum number of snapshots that should be stored at any time

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`schedule`	`(string -> string) map`	*RO/constructor*	schedule of the snapshot containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone
`type`	`vmss_type`	*RO/constructor*	type of the snapshot schedule
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VMs`	`VM ref set`	*RO/runtime*	all VMs attached to this snapshot schedule

RPCs associated with class: VMSS

RPC name: addtoschedule

Overview:

Signature:

```
void add_to_schedule (session ref session_ref, VMSS ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: create

Overview:

Create a new VMSS instance, and return its handle.

Signature:

```
VMSS ref create (session ref session_ref, VMSS record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `VMSS ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VMSS instance.

Signature:

```
void destroy (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMSSs known to the system.

Signature:

```
VMSS ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VMSS ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMSS references to VMSS records for all VMSSs known to the system.

Signature:

```
(VMSS ref -> VMSS record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VMSS ref -> VMSS record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the VMSS instances with the given label.

Signature:

```
VMSS ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VMSS ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VMSS instance with the specified UUID.

Signature:

```
VMSS ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VMSS ref`

reference to the object

RPC name: getenabled

Overview:

Get the enabled field of the given VMSS.

Signature:

```
bool get_enabled (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getfrequency

Overview:

Get the frequency field of the given VMSS.

Signature:

```
vmss_frequency get_frequency (session ref session_ref, VMSS ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmss_frequency`

value of the field

RPC name: getlastruntime

Overview:

Get the lastruntime field of the given VMSS.

Signature:

```
datetime get_last_run_time (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given VMSS.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_name_description (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VMSS.

Signature:

```
string get_name_label (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a record containing the current state of the given VMSS.

Signature:

```
VMSS record get_record (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VMSS record`

all fields from the object

RPC name: getretainedsnapshots

Overview:

Get the retainedsnapshots field of the given VMSS.

Signature:

```
int get_retained_snapshots (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getschedule

Overview:

Get the schedule field of the given VMSS.

Signature:

```
(string -> string) map get_schedule (session ref session_ref, VMSS ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VMSS.

Signature:

```
vmss_type get_type (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `vmss_type`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VMSS.

Signature:

```
string get_uuid (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVMs

Overview:

Get the VMs field of the given VMSS.

Signature:

```
VM ref set get_VMs (session ref session_ref, VMSS ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: removefromschedule

Overview:

Signature:

```
void remove_from_schedule (session ref session_ref, VMSS ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setenabled

Overview:

Set the enabled field of the given VMSS.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_enabled (session ref session_ref, VMSS ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setfrequency

Overview:

Set the value of the frequency field

Signature:

```
void set_frequency (session ref session_ref, VMSS ref self, vmss_frequency
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`vmss_frequency`	`value`	the snapshot schedule frequency

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setlastruntime

Overview:

Signature:

```
void set_last_run_time (session ref session_ref, VMSS ref self, datetime
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`datetime`	`value`	The time at which the schedule was last run. When the timezone is missing, UTC is assumed

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given VMSS.

Signature:

```
void set_name_description (session ref session_ref, VMSS ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VMSS.

Signature:

```
void set_name_label (session ref session_ref, VMSS ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setretainedsnapshots

Overview:

Signature:

```
void set_retained_snapshots (session ref session_ref, VMSS ref self, int
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The schedule snapshot
`int`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setschedule

Overview:

Signature:

```
void set_schedule (session ref session_ref, VMSS ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settype

Overview:

Signature:

```
void set_type (session ref session_ref, VMSS ref self, vmss_type value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`vmss_type`	`value`	the snapshot schedule type

Minimum Role: pool-operator

Return Type: `void`

RPC name: snapshotnow

Overview:

This call executes the snapshot schedule immediately

Signature:

```
string snapshot_now (session ref session_ref, VMSS ref vmss)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`vmss`	Snapshot Schedule to execute

Minimum Role: pool-operator

Return Type: `string`

An XMLRPC result

Class: VTPM

A virtual TPM device

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: VTPM

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vtpm_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`backend`	`VM ref`	*RO/runtime*	The domain where the backend is located (unused)
`current_operations`	`(string -> vtpm_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`is_protected`	`bool`	*RO/runtime*	Whether the contents of the VTPM are secured according to the TPM spec
`is_unique`	`bool`	*RO/constructor*	Whether the contents are never copied, satisfying the TPM spec
`persistence_backend`	`persistence_backend`	*RO/runtime*	The backend where the vTPM is persisted
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VM`	`VM ref`	*RO/constructor*	The virtual machine the TPM is attached to

RPCs associated with class: VTPM

RPC name: create

Overview:

Create a new VTPM instance, and return its handle.

Signature:

```
VTPM ref create (session ref session_ref, VM ref vM, bool is_unique)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vM`	The VM reference the VTPM will be attached to
`bool`	`is_unique`	Whether the VTPM must be unique

Minimum Role: vm-admin

Return Type: `VTPM ref`

The reference of the newly created VTPM

RPC name: destroy

Overview:

Destroy the specified VTPM instance, along with its state.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void destroy (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	The reference to the VTPM object

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VTPMs known to the system.

Signature:

```
VTPM ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VTPM ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VTPM references to VTPM records for all VTPMs known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(VTPM ref -> VTPM record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VTPM ref -> VTPM record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VTPM.

Signature:

```
vtpm_operations set get_allowed_operations (session ref session_ref, VTPM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vtpm_operations set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbackend

Overview:

Get the backend field of the given VTPM.

Signature:

```
VM ref get_backend (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VTPM instance with the specified UUID.

Signature:

```
VTPM ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VTPM ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VTPM.

Signature:

```
(string -> vtpm_operations) map get_current_operations (session ref
session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vtpm_operations) map`

value of the field

RPC name: getisprotected

Overview:

Get the isprotected field of the given VTPM.

Signature:

```
bool get_is_protected (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisunique

Overview:

Get the isunique field of the given VTPM.

Signature:

```
bool get_is_unique (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getpersistencebackend

Overview:

Get the persistencebackend field of the given VTPM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
persistence_backend get_persistence_backend (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `persistence_backend`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VTPM.

Signature:

```
VTPM record get_record (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VTPM record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given VTPM.

Signature:

```
string get_uuid (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VTPM.

Signature:

```
VM ref get_VM (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VM ref`

value of the field

Class: VUSB

Describes the vusb device

Fields for class: VUSB

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vusb_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`current_operations`	`(string -> vusb_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`currently_attached`	`bool`	*RO/runtime*	is the device currently attached
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`USB_group`	`USB_group ref`	*RO/runtime*	USB group used by the VUSB
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VM`	`VM ref`	*RO/runtime*	VM that owns the VUSB

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VUSB

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VUSB.

Signature:

```
void add_to_other_config (session ref session_ref, VUSB ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

Overview:

Create a new VUSB record in the database only

Signature:

```
VUSB ref create (session ref session_ref, VM ref VM, USB_group ref
USB_group, (string -> string) map other_config)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`VM`	The VM
`USB_group ref`	`USB_group`	
`(string -> string) map`	`other_config`	

Minimum Role: pool-admin

Return Type: `VUSB ref`

The ref of the newly created VUSB record.

RPC name: destroy

Overview:

Removes a VUSB record from the database

Signature:

```
void destroy (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	The VUSB to destroy about

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VUSBs known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VUSB ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VUSB ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VUSB references to VUSB records for all VUSBs known to the system.

Signature:

```
(VUSB ref -> VUSB record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VUSB ref -> VUSB record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VUSB.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
vusb_operations set get_allowed_operations (session ref session_ref, VUSB
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vusb_operations set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VUSB instance with the specified UUID.

Signature:

```
VUSB ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VUSB ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VUSB.

Signature:

```
(string -> vusb_operations) map get_current_operations (session ref
session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vusb_operations) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VUSB.

Signature:

```
bool get_currently_attached (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VUSB.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VUSB.

Signature:

```
VUSB record get_record (session ref session_ref, VUSB ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VUSB record`

all fields from the object

RPC name: getUSBgroup

Overview:

Get the USBgroup field of the given VUSB.

Signature:

```
USB_group ref get_USB_group (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `USB_group ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VUSB.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VUSB.

Signature:

```
VM ref get_VM (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given VUSB. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VUSB ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VUSB.

Signature:

```
void set_other_config (session ref session_ref, VUSB ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: unplug

Overview:

Unplug the vusb device from the vm.

Signature:

```
void unplug (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	vusb deivce

Minimum Role: pool-admin

Return Type: `void`

Error Codes

The following is a list of all possible errors that can be issued by API calls.

ACTIVATIONWHILENOTFREE

An activation key can only be applied when the edition is set to 'free'.

Signature:

```
ACTIVATION_WHILE_NOT_FREE()
```

ADDRESSVIOLATESLOCKINGCONSTRAINT

The specified IP address violates the VIF locking configuration.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
ADDRESS_VIOLATES_LOCKING_CONSTRAINT(address)
```

APPLYGUIDANCEFAILED

Failed to apply guidance on a host after updating.

Signature:

```
APPLY_GUIDANCE_FAILED(ref)
```

APPLYLIVEPATCHFAILED

Failed to apply a livepatch.

Signature:

```
APPLY_LIVEPATCH_FAILED(livepatch)
```

APPLYUPDATESFAILED

Failed to apply updates on a host.

Signature:

```
APPLY_UPDATES_FAILED(ref)
```

APPLYUPDATESINPROGRESS

The operation could not be performed because applying updates is in progress.

Signature:

```
APPLY_UPDATES_IN_PROGRESS()
```

AUTHALREADYENABLED

External authentication for this server is already enabled.

Signature:

```
AUTH_ALREADY_ENABLED(current auth_type, current service_name)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHDISABLEFAILED

The host failed to disable external authentication.

Signature:

```
AUTH_DISABLE_FAILED(message)
```

AUTHDISABLEFAILEDPERMISSIONDENIED

The host failed to disable external authentication, permission denied.

Signature:

```
AUTH_DISABLE_FAILED_PERMISSION_DENIED(message)
```

AUTHDISABLEFAILEDWRONGCREDENTIALS

The host failed to disable external authentication, wrong credentials.

Signature:

```
AUTH_DISABLE_FAILED_WRONG_CREDENTIALS(message)
```

AUTHENABLEFAILED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED(message)
```

AUTHENABLEFAILEDDOMAINLOOKUPFAILED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED(message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHENABLEFAILEDINVALIDACCOUNT

The host failed to enable external authentication, invalid account.

Signature:

```
AUTH_ENABLE_FAILED_INVALID_ACCOUNT(message)
```

AUTHENABLEFAILEDINVALIDOU

The host failed to enable external authentication, invalid OU.

Signature:

```
AUTH_ENABLE_FAILED_INVALID_OU(message)
```

AUTHENABLEFAILEDINVALIDTRUSTEDCERTS

The host failed to enable external authentication, invalid trusted certificates.

Signature:

```
AUTH_ENABLE_FAILED_INVALID_TRUSTED_CERTS(message)
```

AUTHENABLEFAILEDNOTRUSTEDCERTS

The host failed to enable external authentication, no trusted certificates

Signature:

```
AUTH_ENABLE_FAILED_NO_TRUSTED_CERTS(message)
```

AUTHENABLEFAILEDPERMISSIONDENIED

The host failed to enable external authentication, permission denied.

Signature:

```
AUTH_ENABLE_FAILED_PERMISSION_DENIED(message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHENABLEFAILEDUNAVAILABLE

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_UNAVAILABLE(message)
```

AUTHENABLEFAILEDWRONGCREDENTIALS

The host failed to enable external authentication, wrong credentials.

Signature:

```
AUTH_ENABLE_FAILED_WRONG_CREDENTIALS(message)
```

AUTHINVALIDTRUSTEDCERTS

The trusted certificates are invalid to setup a TLS connection to Active Directory.

Signature:

```
AUTH_INVALID_TRUSTED_CERTS(message)
```

AUTHISDISABLED

External authentication is disabled

Signature:

```
AUTH_IS_DISABLED()
```

AUTHNOTRUSTEDCERTS

No trusted CA certificates found to setup a TLS connection to Active Directory

Signature:

```
AUTH_NO_TRUSTED_CERTS(message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHSERVICEERROR

Error querying the external directory service.

Signature:

```
AUTH_SERVICE_ERROR(message)
```

AUTHSETUPTLSCONNECTION

Failed to set up TLS connection for external authentication.

Signature:

```
AUTH_SETUP_TLS_CONNECTION(message)
```

AUTHSETLDAPSFAILED

The pool failed to set LDAPS configuration.

Signature:

```
AUTH_SET_LDAPS_FAILED(host, message)
```

AUTHUNKNOWNTYPE

Unknown type of external authentication.

Signature:

```
AUTH_UNKNOWN_TYPE(type)
```

BACKUPSCRIPTFAILED

The backup could not be performed because the backup script failed.

Signature:

```
BACKUP_SCRIPT_FAILED(log)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

BALLOONINGTIMEOUTBEFOREMIGRATION

Timeout trying to balloon down memory before VM migration. If the error occurs repeatedly, consider increasing the memorydynamicmin value.

Signature:

```
BALLOONING_TIMEOUT_BEFORE_MIGRATION(vm)
```

BLOCKEDREPOURL

Cannot create the repository as the url is blocked, please check your settings.

Signature:

```
BLOCKED_REPO_URL(url)
```

BOOTLOADERFAILED

The bootloader returned an error

Signature:

```
BOOTLOADER_FAILED(vm, msg)
```

BOOTDEVICEPASSTHROUGHDISALLOWED

Passing through a PCI device backing a boot disk is disallowed

Signature:

```
BOOT_DEVICE_PASSTHROUGH_DISALLOWED(device)
```

BRIDGENAMEEXISTS

The specified bridge already exists.

Signature:

```
BRIDGE_NAME_EXISTS(bridge)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

BRIDGENOTAVAILABLE

Could not find bridge required by VM.

Signature:

```
BRIDGE_NOT_AVAILABLE(bridge)
```

BUNDLEREPOSITORYALREADYEXISTS

The bundle repository already exists.

Signature:

```
BUNDLE_REPOSITORY_ALREADY_EXISTS(ref)
```

BUNDLEREPONOTENABLED

Cannot sync bundle as the bundle repository is not enabled.

Signature:

```
BUNDLE_REPO_NOT_ENABLED()
```

BUNDLESYNCFAILED

Syncing with bundle repository failed.

Signature:

```
BUNDLE_SYNC_FAILED()
```

BUNDLEUNPACKFAILED

Failed to unpack bundle file.

Signature:

```
BUNDLE_UNPACK_FAILED(error)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTADDTUNNELTOBONDSLAVE

This PIF is a bond member and cannot have a tunnel on it.

Signature:

```
CANNOT_ADD_TUNNEL_TO_BOND_SLAVE(PIF)
```

CANNOTADDTUNNELTOSRIOVLOGICAL

This is a network SRIOV logical PIF and cannot have a tunnel on it.

Signature:

```
CANNOT_ADD_TUNNEL_TO_SRIOV_LOGICAL(PIF)
```

CANNOTADDTUNNELTOVLANONSRIOVLOGICAL

This is a vlan PIF on network SRIOV and cannot have a tunnel on it.

Signature:

```
CANNOT_ADD_TUNNEL_TO_VLAN_ON_SRIOV_LOGICAL(PIF)
```

CANNOTADDVLANTOBONDSLAVE

This PIF is a bond member and cannot have a VLAN on it.

Signature:

```
CANNOT_ADD_VLAN_TO_BOND_SLAVE(PIF)
```

CANNOTCHANGEPIFPROPERTIES

The properties of this PIF cannot be changed. Only the properties of nonbonded physical PIFs, or bond interfaces can be changed.

Signature:

```
CANNOT_CHANGE_PIF_PROPERTIES(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTCONTACTHOST

Cannot forward messages because the server cannot be contacted. The server may be switched off or there may be network connectivity problems.

Signature:

```
CANNOT_CONTACT_HOST(host)
```

CANNOTCREATESTATEFILE

An HA statefile could not be created, perhaps because no SR with the appropriate capability was found.

Signature:

```
CANNOT_CREATE_STATE_FILE()
```

CANNOTDESTROYDISASTERRECOVERYTASK

The disaster recovery task could not be cleanly destroyed.

Signature:

```
CANNOT_DESTROY_DISASTER_RECOVERY_TASK(reason)
```

CANNOTDESTROYSYSTEMNETWORK

You tried to destroy a system network: these cannot be destroyed.

Signature:

```
CANNOT_DESTROY_SYSTEM_NETWORK(network)
```

CANNOTENABLEREDOLOG

Could not enable redo log.

Signature:

```
CANNOT_ENABLE_REDO_LOG(reason)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTEVACUATEHOST

This server cannot be evacuated.

Signature:

```
CANNOT_EVACUATE_HOST(errors)
```

CANNOTFETCHPATCH

The requested update could not be obtained from the coordinator.

Signature:

```
CANNOT_FETCH_PATCH(uuid)
```

CANNOTFINDOEMBACKUPPARTITION

The backup partition to stream the update to cannot be found.

Signature:

```
CANNOT_FIND_OEM_BACKUP_PARTITION()
```

CANNOTFINDPATCH

The requested update could not be found. This can occur when you designate a new coordinator or xe patchclean. Please upload the update again.

Signature:

```
CANNOT_FIND_PATCH()
```

CANNOTFINDSTATEPARTITION

This operation could not be performed because the state partition could not be found

Signature:

```
CANNOT_FIND_STATE_PARTITION()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTFINDUPDATE

The requested update could not be found. Please upload the update again. This can occur when you run xe updatepoolclean before xe updateapply.

Signature:

```
CANNOT_FIND_UPDATE()
```

CANNOTFORGETSRIOVLOGICAL

This is a network SRIOV logical PIF and cannot do forget on it

Signature:

```
CANNOT_FORGET_SRIOV_LOGICAL(PIF)
```

CANNOTPLUGBONDSLAVE

This PIF is a bond member and cannot be plugged.

Signature:

```
CANNOT_PLUG_BOND_SLAVE(PIF)
```

CANNOTPLUGVIF

Cannot plug VIF

Signature:

```
CANNOT_PLUG_VIF(VIF)
```

CANNOTRESETCONTROLDOMAIN

The powerstate of a control domain cannot be reset.

Signature:

```
CANNOT_RESET_CONTROL_DOMAIN(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTRESTARTDEVICEMODEL

Cannot restart device models of paused VMs residing on the host.

Signature:

```
CANNOT_RESTART_DEVICE_MODEL(ref)
```

CANNOTPERIODICSYNCUPDATES

The currently enabled repositories do not support periodic automatic updates.

Signature:

```
CAN_NOT_PERIODIC_SYNC_UPDATES()
```

CANNOTSYNCUPDATES

The currently enabled repositories do not support synchronization of updates.

Signature:

```
CAN_NOT_SYNC_UPDATES()
```

CERTIFICATEALREADYEXISTS

A certificate already exists with the specified name.

Signature:

```
CERTIFICATE_ALREADY_EXISTS(name)
```

CERTIFICATECORRUPT

The specified certificate is corrupt or unreadable.

Signature:

```
CERTIFICATE_CORRUPT(name)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CERTIFICATEDOESNOTEXIST

The specified certificate does not exist.

Signature:

```
CERTIFICATE_DOES_NOT_EXIST(name)
```

CERTIFICATELACKSPURPOSE

No purpose is specified for the provided certificate.

Signature:

```
CERTIFICATE_LACKS_PURPOSE()
```

CERTIFICATELIBRARYCORRUPT

The certificate library is corrupt or unreadable.

Signature:

```
CERTIFICATE_LIBRARY_CORRUPT()
```

CERTIFICATENAMEINVALID

The specified certificate name is invalid.

Signature:

```
CERTIFICATE_NAME_INVALID(name)
```

CHANGEPASSWORDREJECTED

The system rejected the password change request; perhaps the new password was too short?

Signature:

```
CHANGE_PASSWORD_REJECTED(msg)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CLUSTEREDSRDEGRADED

An SR is using clustered local storage. It is not safe to reboot a host at the moment.

Signature:

```
CLUSTERED_SR_DEGRADED(sr)
```

CLUSTERINGDISABLED

An operation was attempted while clustering was disabled on the clusterhost.

Signature:

```
CLUSTERING_DISABLED(cluster_host)
```

CLUSTERINGENABLED

An operation was attempted while clustering was enabled on the clusterhost.

Signature:

```
CLUSTERING_ENABLED(cluster_host)
```

CLUSTERALREADYEXISTS

A cluster already exists in the pool.

Signature:

```
CLUSTER_ALREADY_EXISTS()
```

CLUSTERCREATEINPROGRESS

The operation could not be performed because cluster creation is in progress.

Signature:

```
CLUSTER_CREATE_IN_PROGRESS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CLUSTERDOESNOTHAVEONENODE

An operation failed as it expected the cluster to have only one node but found multiple clusterhosts.

Signature:

```
CLUSTER_DOES_NOT_HAVE_ONE_NODE(number_of_nodes)
```

CLUSTERFORCEDESTROYFAILED

Force destroy failed on a Clusterhost while force destroying the cluster.

Signature:

```
CLUSTER_FORCE_DESTROY_FAILED(cluster)
```

CLUSTERHOSTISLAST

The last cluster host cannot be destroyed. Destroy the cluster instead

Signature:

```
CLUSTER_HOST_IS_LAST(cluster_host)
```

CLUSTERHOSTNOTJOINED

Clusterhost operation failed as the clusterhost has not joined the cluster.

Signature:

```
CLUSTER_HOST_NOT_JOINED(cluster_host)
```

CLUSTERSTACKINUSE

The cluster stack is still in use by at least one plugged PBD.

Signature:

```
CLUSTER_STACK_IN_USE(cluster_stack)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CONFIGUREREPOSITORIESINPROGRESS

The operation could not be performed because other repositoryies isare already being configured.

Signature:

```
CONFIGURE_REPOSITORIES_IN_PROGRESS()
```

COULDNOTFINDNETWORKINTERFACEWITHSPECIFIEDDEVICENAMEANDMACADDRESS

Could not find a network interface with the specified device name and MAC address.

Signature:

```
COULD_NOT_FIND_NETWORK_INTERFACE_WITH_SPECIFIED_DEVICE_NAME_AND_MAC_ADDRES
S(device, mac)
```

COULDNOTIMPORTDATABASE

An error occurred while attempting to import a database from a metadata VDI

Signature:

```
COULD_NOT_IMPORT_DATABASE(reason)
```

COULDNOTUPDATEIGMPSNOOPINGEVERYWHERE

The IGMP Snooping setting cannot be applied for some of the host, networks.

Signature:

```
COULD_NOT_UPDATE_IGMP_SNOOPING_EVERYWHERE()
```

CPUFEATUREMASKINGNOTSUPPORTED

The CPU does not support masking of features.

Signature:

```
CPU_FEATURE_MASKING_NOT_SUPPORTED(details)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CRLALREADYEXISTS

A CRL already exists with the specified name.

Signature:

```
CRL_ALREADY_EXISTS(name)
```

CRLCORRUPT

The specified CRL is corrupt or unreadable.

Signature:

```
CRL_CORRUPT(name)
```

CRLDOESNOTEXIST

The specified CRL does not exist.

Signature:

```
CRL_DOES_NOT_EXIST(name)
```

CRLNAMEINVALID

The specified CRL name is invalid.

Signature:

```
CRL_NAME_INVALID(name)
```

DBUNIQUENESSCONSTRAINTVIOLATION

You attempted an operation which would have resulted in duplicate keys in the database.

Signature:

```
DB_UNIQUENESS_CONSTRAINT_VIOLATION(table, field, value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DEFAULTSRNOTFOUND

The default SR reference does not point to a valid SR

Signature:

```
DEFAULT_SR_NOT_FOUND(sr)
```

DEVICEALREADYATTACHED

The device is already attached to a VM

Signature:

```
DEVICE_ALREADY_ATTACHED(device)
```

DEVICEALREADYDETACHED

The device is not currently attached

Signature:

```
DEVICE_ALREADY_DETACHED(device)
```

DEVICEALREADYEXISTS

A device with the name given already exists on the selected VM

Signature:

```
DEVICE_ALREADY_EXISTS(device)
```

DEVICEATTACHTIMEOUT

A timeout happened while attempting to attach a device to a VM.

Signature:

```
DEVICE_ATTACH_TIMEOUT(type, ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DEVICEDETACHREJECTED

The VM rejected the attempt to detach the device.

Signature:

```
DEVICE_DETACH_REJECTED(type, ref, msg)
```

DEVICEDETACHTIMEOUT

A timeout happened while attempting to detach a device from a VM.

Signature:

```
DEVICE_DETACH_TIMEOUT(type, ref)
```

DEVICENOTATTACHED

The operation could not be performed because the VBD was not connected to the VM.

Signature:

```
DEVICE_NOT_ATTACHED(VBD)
```

DISABLESSHFAILED

Failed to disable SSH access.

Signature:

```
DISABLE_SSH_FAILED(host)
```

DISABLESSHPARTIALLYFAILED

Some of hosts failed to disable SSH access.

Signature:

```
DISABLE_SSH_PARTIALLY_FAILED(hosts)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DISKVBDMUSTBEREADWRITEFORHVM

All VBDs of type 'disk' must be read/write for HVM guests

Signature:

```
DISK_VBD_MUST_BE_READWRITE_FOR_HVM(vbd)
```

DOMAINBUILDERERROR

An internal error generated by the domain builder.

Signature:

```
DOMAIN_BUILDER_ERROR(function, code, message)
```

DOMAINEXISTS

The operation could not be performed because a domain still exists for the specified VM.

Signature:

```
DOMAIN_EXISTS(vm, domid)
```

DUPLICATEMACSEED

This MAC seed is already in use by a VM in the pool

Signature:

```
DUPLICATE_MAC_SEED(seed)
```

DUPLICATEPIFDEVICENAME

A PIF with this specified device name already exists.

Signature:

```
DUPLICATE_PIF_DEVICE_NAME(device)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DUPLICATEVM

Cannot restore this VM because it would create a duplicate

Signature:

```
DUPLICATE_VM(vm)
```

ENABLESSHFAILED

Failed to enable SSH access.

Signature:

```
ENABLE_SSH_FAILED(host)
```

ENABLESSHPARTIALLYFAILED

Some of hosts failed to enable SSH access.

Signature:

```
ENABLE_SSH_PARTIALLY_FAILED(hosts)
```

EVENTSLOST

Some events have been lost from the queue and cannot be retrieved.

Signature:

```
EVENTS_LOST()
```

EVENTFROMTOKENPARSEFAILURE

The event.from token could not be parsed. Valid values include: '', and a value returned from a previous event.from call.

Signature:

```
EVENT_FROM_TOKEN_PARSE_FAILURE(token)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

EVENTSUBSCRIPTIONPARSEFAILURE

The server failed to parse your event subscription. Valid values include: , classname, classname/objectreference.

Signature:

```
EVENT_SUBSCRIPTION_PARSE_FAILURE(subscription)
```

FAILEDTOSTARTEMULATOR

An emulator required to run this VM failed to start

Signature:

```
FAILED_TO_START_EMULATOR(vm, name, msg)
```

FEATUREREQUIRESHVM

The VM is set up to use a feature that requires it to boot as HVM.

Signature:

```
FEATURE_REQUIRES_HVM(details)
```

FEATURERESTRICTED

The use of this feature is restricted.

Signature:

```
FEATURE_RESTRICTED()
```

FIELDTYPEERROR

The value specified is of the wrong type

Signature:

```
FIELD_TYPE_ERROR(field)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

GETHOSTUPDATESFAILED

Failed to get available updates from a host.

Signature:

```
GET_HOST_UPDATES_FAILED(ref)
```

GETUPDATESFAILED

Failed to get available updates from the pool.

Signature:

```
GET_UPDATES_FAILED()
```

GPUGROUPCONTAINSNOPGPUS

The GPU group does not contain any PGPUs.

Signature:

```
GPU_GROUP_CONTAINS_NO_PGPUS(gpu_group)
```

GPUGROUPCONTAINSPGPU

The GPU group contains active PGPUs and cannot be deleted.

Signature:

```
GPU_GROUP_CONTAINS_PGPU(pgpus)
```

GPUGROUPCONTAINSVGPU

The GPU group contains active VGPUs and cannot be deleted.

Signature:

```
GPU_GROUP_CONTAINS_VGPU(vgpus)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HANDLEINVALID

You gave an invalid object reference. The object may have recently been deleted. The class parameter gives the type of reference given, and the handle parameter echoes the bad value given.

Signature:

```
HANDLE_INVALID(class, handle)
```

HAABORTNEWMASTER

This server cannot accept the proposed new coordinator setting at this time.

Signature:

```
HA_ABORT_NEW_MASTER(reason)
```

HACANNOTCHANGEBONDSTATUSOFMGMTIFACE

This operation cannot be performed because creating or deleting a bond involving the management interface is not allowed while HA is on. In order to do that, disable HA, create or delete the bond then reenable HA.

Signature:

```
HA_CANNOT_CHANGE_BOND_STATUS_OF_MGMT_IFACE()
```

HACONSTRAINTVIOLATIONNETWORKNOTSHARED

This operation cannot be performed because the referenced network is not properly shared. The network must either be entirely virtual or must be physically present via a currentlyattached PIF on every host.

Signature:

```
HA_CONSTRAINT_VIOLATION_NETWORK_NOT_SHARED(network)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HACONSTRAINTVIOLATIONSRNOTSHARED

This operation cannot be performed because the referenced SR is not properly shared. The SR must both be marked as shared and a currentlyattached PBD must exist for each host.

Signature:

```
HA_CONSTRAINT_VIOLATION_SR_NOT_SHARED(SR)
```

HADISABLEINPROGRESS

The operation could not be performed because HA disable is in progress

Signature:

```
HA_DISABLE_IN_PROGRESS()
```

HAENABLEINPROGRESS

The operation could not be performed because HA enable is in progress

Signature:

```
HA_ENABLE_IN_PROGRESS()
```

HAFAILEDTOFORMLIVESET

HA could not be enabled on the Pool because a liveset could not be formed: check storage and network heartbeat paths.

Signature:

```
HA_FAILED_TO_FORM_LIVESET()
```

HAHEARTBEATDAEMONSTARTUPFAILED

The server could not join the liveset because the HA daemon failed to start.

Signature:

```
HA_HEARTBEAT_DAEMON_STARTUP_FAILED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HAHOSTCANNOTACCESSSTATEFILE

The server could not join the liveset because the HA daemon could not access the heartbeat disk.

Signature:

```
HA_HOST_CANNOT_ACCESS_STATEFILE()
```

HAHOSTCANNOTSEEPEERS

The operation failed because the HA software on the specified server could not see a subset of other servers. Check your network connectivity.

Signature:

```
HA_HOST_CANNOT_SEE_PEERS(host, all, subset)
```

HAHOSTISARMED

The operation could not be performed while the server is still armed; it must be disarmed first.

Signature:

```
HA_HOST_IS_ARMED(host)
```

HAISENABLED

The operation could not be performed because HA is enabled on the Pool

Signature:

```
HA_IS_ENABLED()
```

HALOSTSTATEFILE

This server lost access to the HA statefile.

Signature:

```
HA_LOST_STATEFILE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HANOTENABLED

The operation could not be performed because HA is not enabled on the Pool

Signature:

```
HA_NOT_ENABLED()
```

HANOTINSTALLED

The operation could not be performed because the HA software is not installed on this server.

Signature:

```
HA_NOT_INSTALLED(host)
```

HANOPLAN

Cannot find a plan for placement of VMs as there are no other servers available.

Signature:

```
HA_NO_PLAN()
```

HAOPERATIONWOULDBREAKFAILOVERPLAN

This operation cannot be performed because it would invalidate VM failover planning such that the system would be unable to guarantee to restart protected VMs after a Host failure.

Signature:

```
HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN()
```

HAPOOLISENABLEDBUTHOSTISDISABLED

This server cannot join the pool because the pool has HA enabled but this server has HA disabled.

Signature:

```
HA_POOL_IS_ENABLED_BUT_HOST_IS_DISABLED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HASHOULDBEFENCED

Server cannot rejoin pool because it should have fenced it is not in the coordinator's partition.

Signature:

```
HA_SHOULD_BE_FENCED(host)
```

HATOOFEWHOSTS

HA can only be enabled for 2 servers or more. Note that 2 servers requires a preconfigured quorum tiebreak script.

Signature:

```
HA_TOO_FEW_HOSTS()
```

HOSTSNOTCOMPATIBLE

The hosts in this pool are not compatible.

Signature:

```
HOSTS_NOT_COMPATIBLE()
```

HOSTSNOTHOMOGENEOUS

The hosts in this pool are not homogeneous.

Signature:

```
HOSTS_NOT_HOMOGENEOUS(reason)
```

HOSTBROKEN

This server failed in the middle of an automatic failover operation and needs to retry the failover action.

Signature:

```
HOST_BROKEN()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTCANNOTATTACHNETWORK

Server cannot attach network in the case of NIC bonding, this may be because attaching the network on this server would require other networks that are currently active to be taken down.

Signature:

```
HOST_CANNOT_ATTACH_NETWORK(host, network)
```

HOSTCANNOTDESTROYSELF

The pool coordinator host cannot be removed.

Signature:

```
HOST_CANNOT_DESTROY_SELF(host)
```

HOSTCANNOTREADMETRICS

The metrics of this server could not be read.

Signature:

```
HOST_CANNOT_READ_METRICS()
```

HOSTCDDRIVEEMPTY

The host CDROM drive does not contain a valid CD

Signature:

```
HOST_CD_DRIVE_EMPTY()
```

HOSTDISABLED

The specified server is disabled.

Signature:

```
HOST_DISABLED(host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTDISABLEDINDEFINITELY

The specified server is disabled and can only be reenabled manually with Host.enable.

Signature:

```
HOST_DISABLED_INDEFINITELY(host)
```

HOSTDISABLEDUNTILREBOOT

The specified server is disabled and cannot be reenabled until after it has rebooted.

Signature:

```
HOST_DISABLED_UNTIL_REBOOT(host)
```

HOSTDRIVERNOHARDWARE

No hardware present for this host driver variant

Signature:

```
HOST_DRIVER_NO_HARDWARE(driver variant)
```

HOSTEVACUATEINPROGRESS

This host is being evacuated.

Signature:

```
HOST_EVACUATE_IN_PROGRESS(host)
```

HOSTEVACUATIONISREQUIRED

Host evacuation is required before applying updates.

Signature:

```
HOST_EVACUATION_IS_REQUIRED(host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTHASNOMANAGEMENTIP

The server failed to acquire an IP address on its management interface and therefore cannot contact the coordinator.

Signature:

```
HOST_HAS_NO_MANAGEMENT_IP()
```

HOSTHASRESIDENTVMS

This server cannot be forgotten because there are user VMs still running.

Signature:

```
HOST_HAS_RESIDENT_VMS(host)
```

HOSTINEMERGENCYMODE

Cannot perform operation as the host is running in emergency mode.

Signature:

```
HOST_IN_EMERGENCY_MODE()
```

HOSTINUSE

This operation cannot be completed as the host is in use by at least the object of type and ref echoed below.

Signature:

```
HOST_IN_USE(host, type, ref)
```

HOSTISLIVE

This operation cannot be completed because the server is still live.

Signature:

```
HOST_IS_LIVE(host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTISSLAVE

You cannot make regular API calls directly on a supporter. Please pass API calls via the coordinator host.

Signature:

```
HOST_IS_SLAVE(Master IP address)
```

HOSTITSOWNSLAVE

The host is its own supporter. Please use poolemergencytransitiontomaster or poolemergencyresetmaster.

Signature:

```
HOST_ITS_OWN_SLAVE()
```

HOSTMASTERCANNOTTALKBACK

The coordinator reports that it cannot talk back to the supporter on the supplied management IP address.

Signature:

```
HOST_MASTER_CANNOT_TALK_BACK(ip)
```

HOSTNAMEINVALID

The server name is invalid.

Signature:

```
HOST_NAME_INVALID(reason)
```

HOSTNOTDISABLED

This operation cannot be performed because the host is not disabled. Please disable the host and then try again.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
HOST_NOT_DISABLED()
```

HOSTNOTENOUGHFREEMEMORY

Not enough server memory is available to perform this operation.

Signature:

```
HOST_NOT_ENOUGH_FREE_MEMORY(needed, available)
```

HOSTNOTENOUGHPCPUS

The host does not have enough pCPUs to run the VM. It needs at least as many as the VM has vCPUs.

Signature:

```
HOST_NOT_ENOUGH_PCPUS(vcpus, pcpus)
```

HOSTNOTLIVE

This operation cannot be completed as the server is not live.

Signature:

```
HOST_NOT_LIVE()
```

HOSTOFFLINE

You attempted an operation which involves a host which could not be contacted.

Signature:

```
HOST_OFFLINE(host)
```

HOSTPENDINGMANDATORYGUIDANCENOTEMPTY

Operation could not be performed on the host because there is pending mandatory update guidance on it.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
HOST_PENDING_MANDATORY_GUIDANCE_NOT_EMPTY(host)
```

HOSTPOWERONMODEDISABLED

This operation cannot be completed because the server power on mode is disabled.

Signature:

```
HOST_POWER_ON_MODE_DISABLED()
```

HOSTSTILLBOOTING

The host toolstack is still initialising. Please wait.

Signature:

```
HOST_STILL_BOOTING()
```

HOSTUNKNOWNTOMASTER

The coordinator says the server is not known to it. Is the server in the coordinator's database and pointing to the correct coordinator? Are all servers using the same pool secret?

Signature:

```
HOST_UNKNOWN_TO_MASTER(host)
```

HOSTXAPIVERSIONHIGHERTHANCOORDINATOR

The host xapi version is higher than the one in the coordinator

Signature:

```
HOST_XAPI_VERSION_HIGHER_THAN_COORDINATOR(host_xapi_version)
```

ILLEGALVBDDEVICE

The specified VBD device is not recognized: please use a nonnegative integer

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
ILLEGAL_VBD_DEVICE(vbd, device)
```

IMPORTERROR

The VM could not be imported.

Signature:

```
IMPORT_ERROR(msg)
```

IMPORTERRORATTACHEDDISKSNOTFOUND

The VM could not be imported because attached disks could not be found.

Signature:

```
IMPORT_ERROR_ATTACHED_DISKS_NOT_FOUND()
```

IMPORTERRORCANNOTHANDLECHUNKED

Cannot import VM using chunked encoding.

Signature:

```
IMPORT_ERROR_CANNOT_HANDLE_CHUNKED()
```

IMPORTERRORFAILEDTOFINDOBJECT

The VM could not be imported because a required object could not be found.

Signature:

```
IMPORT_ERROR_FAILED_TO_FIND_OBJECT(id)
```

IMPORTERRORPREMATUREEOF

The VM could not be imported; the end of the file was reached prematurely.

Signature:

```
IMPORT_ERROR_PREMATURE_EOF()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

IMPORTERRORSOMECHECKSUMSFAILED

Some data checksums were incorrect; the VM may be corrupt.

Signature:

```
IMPORT_ERROR_SOME_CHECKSUMS_FAILED()
```

IMPORTERRORUNEXPECTEDFILE

The VM could not be imported because the XVA file is invalid: an unexpected file was encountered.

Signature:

```
IMPORT_ERROR_UNEXPECTED_FILE(filename_expected, filename_found)
```

IMPORTINCOMPATIBLEVERSION

The import failed because this export has been created by a different incompatible product version

Signature:

```
IMPORT_INCOMPATIBLE_VERSION()
```

INCOMPATIBLECLUSTERSTACKACTIVE

This operation cannot be performed, because it is incompatible with the currently active HA cluster stack.

Signature:

```
INCOMPATIBLE_CLUSTER_STACK_ACTIVE(cluster_stack)
```

INCOMPATIBLEPIFPROPERTIES

These PIFs cannot be bonded, because their properties are different.

Signature:

```
INCOMPATIBLE_PIF_PROPERTIES()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INCOMPATIBLESTATEFILESR

The specified SR is incompatible with the selected HA cluster stack.

Signature:

```
INCOMPATIBLE_STATEFILE_SR(SR type)
```

INTERFACEHASNOIP

The specified interface cannot be used because it has no IP address

Signature:

```
INTERFACE_HAS_NO_IP(interface)
```

INTERNALERROR

The server failed to handle your request, due to an internal error. The given message may give details useful for debugging the problem.

Signature:

```
INTERNAL_ERROR(message)
```

INVALIDBASEURL

The base url in the repository is invalid.

Signature:

```
INVALID_BASE_URL(url)
```

INVALIDCIDRADDRESSSPECIFIED

A required parameter contained an invalid CIDR address <addr>/<prefix length>

Signature:

```
INVALID_CIDR_ADDRESS_SPECIFIED(parameter)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDCLUSTERSTACK

The cluster stack provided is not supported.

Signature:

```
INVALID_CLUSTER_STACK(cluster_stack)
```

INVALIDDEVICE

The device name is invalid

Signature:

```
INVALID_DEVICE(device)
```

INVALIDEDITION

The edition you supplied is invalid.

Signature:

```
INVALID_EDITION(edition)
```

INVALIDFEATURESTRING

The given feature string is not valid.

Signature:

```
INVALID_FEATURE_STRING(details)
```

INVALIDGPGKEYPATH

The GPG public key file name in the repository is invalid.

Signature:

```
INVALID_GPGKEY_PATH(gpgkey_path)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDIPADDRESSSSPECIFIED

A required parameter contained an invalid IP address

Signature:

```
INVALID_IP_ADDRESS_SPECIFIED(parameter)
```

INVALIDNTPCONFIG

The NTP configuration is invalid.

Signature:

```
INVALID_NTP_CONFIG(reason)
```

INVALIDPATCH

The uploaded patch file is invalid

Signature:

```
INVALID_PATCH()
```

INVALIDPATCHWITHLOG

The uploaded patch file is invalid. See attached log for more details.

Signature:

```
INVALID_PATCH_WITH_LOG(log)
```

INVALIDREPOMDXML

The repomd.xml is invalid.

Signature:

```
INVALID_REPOMD_XML()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDREPOSITORYDOMAINALLOWLIST

The repository domain allowlist has some invalid domains.

Signature:

```
INVALID_REPOSITORY_DOMAIN_ALLOWLIST(domains)
```

INVALIDREPOSITORYPROXYCREDENTIAL

The repository proxy username/password is invalid.

Signature:

```
INVALID_REPOSITORY_PROXY_CREDENTIAL()
```

INVALIDREPOSITORYPROXYURL

The repository proxy URL is invalid.

Signature:

```
INVALID_REPOSITORY_PROXY_URL(url)
```

INVALIDUPDATE

The uploaded update package is invalid.

Signature:

```
INVALID_UPDATE(info)
```

INVALIDUPDATEINFOXML

The updateinfo.xml is invalid.

Signature:

```
INVALID_UPDATEINFO_XML()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDUPDATESYNCDAY

Invalid day of the week chosen for weekly update sync.

Signature:

```
INVALID_UPDATE_SYNC_DAY(day)
```

INVALIDVALUE

The value given is invalid

Signature:

```
INVALID_VALUE(field, value)
```

ISTUNNELACCESSPIF

Cannot create a VLAN or tunnel on top of a tunnel access PIF use the underlying transport PIF instead.

Signature:

```
IS_TUNNEL_ACCESS_PIF(PIF)
```

JOININGHOSTCANNOTBEMASTEROFOTHERHOSTS

The server joining the pool cannot already be a coordinator of another pool.

Signature:

```
JOINING_HOST_CANNOT_BE_MASTER_OF_OTHER_HOSTS()
```

JOININGHOSTCANNOTCONTAINSHAREDSRS

The server joining the pool cannot contain any shared storage.

Signature:

```
JOINING_HOST_CANNOT_CONTAIN_SHARED_SRS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

JOININGHOSTCANNOTHAVERUNNINGORSUSPENDEDVMS

The server joining the pool cannot have any running or suspended VMs.

Signature:

```
JOINING_HOST_CANNOT_HAVE_RUNNING_OR_SUSPENDED_VMS()
```

JOININGHOSTCANNOTHAVERUNNINGVMS

The server joining the pool cannot have any running VMs.

Signature:

```
JOINING_HOST_CANNOT_HAVE_RUNNING_VMS()
```

JOININGHOSTCANNOTHAVEVMSWITHCURRENTOPERATIONS

The host joining the pool cannot have any VMs with active tasks.

Signature:

```
JOINING_HOST_CANNOT_HAVE_VMS_WITH_CURRENT_OPERATIONS()
```

JOININGHOSTCONNECTIONFAILED

There was an error connecting to the host while joining it in the pool.

Signature:

```
JOINING_HOST_CONNECTION_FAILED()
```

JOININGHOSTSERVICEFAILED

There was an error connecting to the server. The service contacted didn't reply properly.

Signature:

```
JOINING_HOST_SERVICE_FAILED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

LICENCERESTRICTION

This operation is not allowed because your license lacks a needed feature. Please contact your support representative.

Signature:

```
LICENCE_RESTRICTION(feature)
```

LICENSECANNOTDOWNGRADEWHILEINPOOL

Cannot downgrade license while in pool. Please disband the pool first, then downgrade licenses on hosts separately.

Signature:

```
LICENSE_CANNOT_DOWNGRADE_WHILE_IN_POOL()
```

LICENSECHECKOUTERROR

The license for the edition you requested is not available.

Signature:

```
LICENSE_CHECKOUT_ERROR(code, message)
```

LICENSEDOESNOTSUPPORTPOOLING

This server cannot join a pool because its license does not support pooling.

Signature:

```
LICENSE_DOES_NOT_SUPPORT_POOLING()
```

LICENSEDOESNOTSUPPORTXHA

HA cannot be enabled because this server's license does not allow it.

Signature:

```
LICENSE_DOES_NOT_SUPPORT_XHA()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

LICENSEEXPIRED

Your license has expired. Please contact your support representative.

Signature:

```
LICENSE_EXPIRED()
```

LICENSEFILEDEPRECATED

This type of license file is for previous versions of the server. Please upgrade to the new licensing system.

Signature:

```
LICENSE_FILE_DEPRECATED()
```

LICENSEHOSTPOOLMISMATCH

Host and pool have incompatible licenses editions.

Signature:

```
LICENSE_HOST_POOL_MISMATCH()
```

LICENSEPROCESSINGERROR

There was an error processing your license. Please contact your support representative.

Signature:

```
LICENSE_PROCESSING_ERROR()
```

LOCATIONNOTUNIQUE

A VDI with the specified location already exists within the SR

Signature:

```
LOCATION_NOT_UNIQUE(SR, location)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

MACDOESNOTEXIST

The MAC address specified does not exist on this server.

Signature:

```
MAC_DOES_NOT_EXIST(MAC)
```

MACINVALID

The MAC address specified is not valid.

Signature:

```
MAC_INVALID(MAC)
```

MACSTILLEXISTS

The MAC address specified still exists on this server.

Signature:

```
MAC_STILL_EXISTS(MAC)
```

MAPDUPLICATEKEY

You tried to add a keyvalue pair to a map, but that key is already there.

Signature:

```
MAP_DUPLICATE_KEY(type, param_name, uuid, key)
```

MEMORYCONSTRAINTVIOLATION

The dynamic memory range does not satisfy the following constraint.

Signature:

```
MEMORY_CONSTRAINT_VIOLATION(constraint)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

MEMORYCONSTRAINTVIOLATIONMAXPIN

The dynamic memory range violates constraint staticmin = dynamicmin = dynamicmax = staticmax.

Signature:

```
MEMORY_CONSTRAINT_VIOLATION_MAXPIN(reason)
```

MEMORYCONSTRAINTVIOLATIONORDER

The dynamic memory range violates constraint staticmin <= dynamicmin <= dynamicmax <= staticmax.

Signature:

```
MEMORY_CONSTRAINT_VIOLATION_ORDER()
```

MESSAGEDEPRECATED

This message has been deprecated.

Signature:

```
MESSAGE_DEPRECATED()
```

MESSAGEMETHODUNKNOWN

You tried to call a method that does not exist. The method name that you used is echoed.

Signature:

```
MESSAGE_METHOD_UNKNOWN(method)
```

MESSAGEPARAMETERCOUNTMISMATCH

You tried to call a method with the incorrect number of parameters. The fullyqualified method name that you used, and the number of received and expected parameters are returned.

Signature:

```
MESSAGE_PARAMETER_COUNT_MISMATCH(method, expected, received)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

MESSAGEREMOVED

This function is no longer available.

Signature:

```
MESSAGE_REMOVED()
```

MIRRORFAILED

The VDI mirroring cannot be performed

Signature:

```
MIRROR_FAILED(vdi)
```

MISSINGCONNECTIONDETAILS

The licenseserver connection details address or port were missing or incomplete.

Signature:

```
MISSING_CONNECTION_DETAILS()
```

MULTIPLEUPDATEREPOSITORIESENABLED

There is more than one update repository being enabled.

Signature:

```
MULTIPLE_UPDATE_REPOSITORIES_ENABLED()
```

NETWORKALREADYCONNECTED

You tried to create a PIF, but the network you tried to attach it to is already attached to some other PIF, and so the creation failed.

Signature:

```
NETWORK_ALREADY_CONNECTED(network, connected PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKCONTAINSPIF

The network contains active PIFs and cannot be deleted.

Signature:

```
NETWORK_CONTAINS_PIF(pifs)
```

NETWORKCONTAINSVIF

The network contains active VIFs and cannot be deleted.

Signature:

```
NETWORK_CONTAINS_VIF(vifs)
```

NETWORKHASINCOMPATIBLESRIOVPIFS

The PIF is not compatible with the selected SRIOV network

Signature:

```
NETWORK_HAS_INCOMPATIBLE_SRIOV_PIFS(PIF, network)
```

NETWORKHASINCOMPATIBLEVLANONSRIOVPIFS

VLAN on the PIF is not compatible with the selected SRIOV VLAN network

Signature:

```
NETWORK_HAS_INCOMPATIBLE_VLAN_ON_SRIOV_PIFS(PIF, network)
```

NETWORKINCOMPATIBLEPURPOSES

You tried to add a purpose to a network but the new purpose is not compatible with an existing purpose of the network or other networks.

Signature:

```
NETWORK_INCOMPATIBLE_PURPOSES(new_purpose, conflicting_purpose)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKINCOMPATIBLEWITHBOND

The network is incompatible with bond

Signature:

```
NETWORK_INCOMPATIBLE_WITH_BOND(network)
```

NETWORKINCOMPATIBLEWITHSRIOV

The network is incompatible with sriov

Signature:

```
NETWORK_INCOMPATIBLE_WITH_SRIOV(network)
```

NETWORKINCOMPATIBLEWITHTUNNEL

The network is incompatible with tunnel

Signature:

```
NETWORK_INCOMPATIBLE_WITH_TUNNEL(network)
```

NETWORKINCOMPATIBLEWITHVLANONBRIDGE

The network is incompatible with vlan on bridge

Signature:

```
NETWORK_INCOMPATIBLE_WITH_VLAN_ON_BRIDGE(network)
```

NETWORKINCOMPATIBLEWITHVLANONSRIOV

The network is incompatible with vlan on sriov

Signature:

```
NETWORK_INCOMPATIBLE_WITH_VLAN_ON_SRIOV(network)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKSRIOVALREADYENABLED

The PIF selected for the SRIOV network is already enabled

Signature:

```
NETWORK_SRIOV_ALREADY_ENABLED(PIF)
```

NETWORKSRIOVDISABLEFAILED

Failed to disable SRIOV on PIF

Signature:

```
NETWORK_SRIOV_DISABLE_FAILED(PIF, msg)
```

NETWORKSRIOVENABLEFAILED

Failed to enable SRIOV on PIF

Signature:

```
NETWORK_SRIOV_ENABLE_FAILED(PIF, msg)
```

NETWORKSRIOVINSUFFICIENTCAPACITY

There is insufficient capacity for VF reservation

Signature:

```
NETWORK_SRIOV_INSUFFICIENT_CAPACITY(network)
```

NETWORKUNMANAGED

The network is not managed by xapi.

Signature:

```
NETWORK_UNMANAGED(network)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTALLOWEDONOEMEDITION

This command is not allowed on the OEM edition.

Signature:

```
NOT_ALLOWED_ON_OEM_EDITION(command)
```

NOTALLOWEDWHENNTPISENABLED

The operation is not allowed on the host when the NTP is enabled.

Signature:

```
NOT_ALLOWED_WHEN_NTP_IS_ENABLED(host)
```

NOTIMPLEMENTED

The function is not implemented

Signature:

```
NOT_IMPLEMENTED(function)
```

NOTINEMERGENCYMODE

This pool is not in emergency mode.

Signature:

```
NOT_IN_EMERGENCY_MODE()
```

NOTSUPPORTEDDURINGUPGRADE

This operation is not supported during an upgrade.

Signature:

```
NOT_SUPPORTED_DURING_UPGRADE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTSYSTEMDOMAIN

The given VM is not registered as a system domain. This operation can only be performed on a registered system domain.

Signature:

```
NOT_SYSTEM_DOMAIN(vm)
```

NOTTRUSTEDCERTIFICATE

The provided certificate is not a trusted certificate.

Signature:

```
NOT_TRUSTED_CERTIFICATE(ref)
```

NOCLUSTERHOSTSREACHABLE

No other cluster host was reachable when joining

Signature:

```
NO_CLUSTER_HOSTS_REACHABLE(cluster)
```

NOCOMPATIBLECLUSTERHOST

Clustering is not enabled on this host or pool.

Signature:

```
NO_COMPATIBLE_CLUSTER_HOST(host)
```

NOHOSTSAVAILABLE

There were no servers available to complete the specified operation.

Signature:

```
NO_HOSTS_AVAILABLE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOMOREREDOLOGSALLOWED

The upper limit of active redo log instances was reached.

Signature:

```
NO_MORE_REDO_LOGS_ALLOWED()
```

NOREPOSITORIESCONFIGURED

No update repositories have been configured.

Signature:

```
NO_REPOSITORIES_CONFIGURED()
```

NOREPOSITORYENABLED

There is no repository being enabled.

Signature:

```
NO_REPOSITORY_ENABLED()
```

NVIDIASRIOVMISCONFIGURED

The NVidia GPU is not configured for SRIOV as expected

Signature:

```
NVIDIA_SRIOV_MISCONFIGURED(host, device_name)
```

NVIDIATOOLSERROR

Nvidia tools error. Please ensure that the latest Nvidia tools are installed

Signature:

```
NVIDIA_TOOLS_ERROR(host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

OBJECTNOLONGEREXISTS

The specified object no longer exists.

Signature:

```
OBJECT_NOLONGER_EXISTS()
```

ONLYALLOWEDONOEMEDITION

This command is only allowed on the OEM edition.

Signature:

```
ONLY_ALLOWED_ON_OEM_EDITION(command)
```

OPENVSWITCHNOTACTIVE

This operation needs the OpenVSwitch networking backend to be enabled on all hosts in the pool.

Signature:

```
OPENVSWITCH_NOT_ACTIVE()
```

OPERATIONBLOCKED

You attempted an operation that was explicitly blocked see the blockedoperations field of the given object.

Signature:

```
OPERATION_BLOCKED(ref, code)
```

OPERATIONNOTALLOWED

You attempted an operation that was not allowed.

Signature:

```
OPERATION_NOT_ALLOWED(reason)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

OPERATIONPARTIALLYFAILED

Some VMs belonging to the appliance threw an exception while carrying out the specified operation

Signature:

```
OPERATION_PARTIALLY_FAILED(operation)
```

OTHEROPERATIONINPROGRESS

Another operation involving the object is currently in progress

Signature:

```
OTHER_OPERATION_IN_PROGRESS(class, object, operation_type, operation_ref)
```

OUTOFSPACE

There is not enough space to upload the update

Signature:

```
OUT_OF_SPACE(location)
```

PATCHALREADYAPPLIED

This patch has already been applied

Signature:

```
PATCH_ALREADY_APPLIED(patch)
```

PATCHALREADYEXISTS

The uploaded patch file already exists

Signature:

```
PATCH_ALREADY_EXISTS(uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PATCHAPPLYFAILED

The patch apply failed. Please see attached output.

Signature:

```
PATCH_APPLY_FAILED(output)
```

PATCHAPPLYFAILEDBACKUPFILESEXIST

The patch apply failed: there are backup files created while applying patch. Please remove these backup files before applying patch again.

Signature:

```
PATCH_APPLY_FAILED_BACKUP_FILES_EXIST(output)
```

PATCHISAPPLIED

The specified patch is applied and cannot be destroyed.

Signature:

```
PATCH_IS_APPLIED()
```

PATCHPRECHECKFAILEDISOMOUNTED

Tools ISO must be ejected from all running VMs.

Signature:

```
PATCH_PRECHECK_FAILED_ISO_MOUNTED(patch)
```

PATCHPRECHECKFAILEDOUTOFSPACE

The patch precheck stage failed: the server does not have enough space.

Signature:

```
PATCH_PRECHECK_FAILED_OUT_OF_SPACE(patch, found_space, required_required)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PATCHPRECHECKFAILEDPREREQUISITEMISSING

The patch precheck stage failed: prerequisite patches are missing.

Signature:

```
PATCH_PRECHECK_FAILED_PREREQUISITE_MISSING(patch,
prerequisite_patch_uuid_list)
```

PATCHPRECHECKFAILEDUNKNOWNERROR

The patch precheck stage failed with an unknown error. See attached info for more details.

Signature:

```
PATCH_PRECHECK_FAILED_UNKNOWN_ERROR(patch, info)
```

PATCHPRECHECKFAILEDVMRUNNING

The patch precheck stage failed: there are one or more VMs still running on the server. All VMs must be suspended before the patch can be applied.

Signature:

```
PATCH_PRECHECK_FAILED_VM_RUNNING(patch)
```

PATCHPRECHECKFAILEDWRONGSERVERBUILD

The patch precheck stage failed: the server is of an incorrect build.

Signature:

```
PATCH_PRECHECK_FAILED_WRONG_SERVER_BUILD(patch, found_build,
required_build)
```

PATCHPRECHECKFAILEDWRONGSERVERVERSION

The patch precheck stage failed: the server is of an incorrect version.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PATCH_PRECHECK_FAILED_WRONG_SERVER_VERSION(patch, found_version,
required_version)
```

PBDEXISTS

A PBD already exists connecting the SR to the server.

Signature:

```
PBD_EXISTS(sr, host, pbd)
```

PERMISSIONDENIED

Caller not allowed to perform this operation.

Signature:

```
PERMISSION_DENIED(message)
```

PGPUINSUFFICIENTCAPACITYFORVGPU

There is insufficient capacity on this PGPU to run the VGPU.

Signature:

```
PGPU_INSUFFICIENT_CAPACITY_FOR_VGPU(pgpu, vgpu_type)
```

PGPUINUSEBYVM

This PGPU is currently in use by running VMs.

Signature:

```
PGPU_IN_USE_BY_VM(VMs)
```

PGPUNOTCOMPATIBLEWITHGPUGROUP

PGPU type not compatible with destination group.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PGPU_NOT_COMPATIBLE_WITH_GPU_GROUP(type, group_types)
```

PIFALLOWSUNPLUG

The operation you requested cannot be performed because the specified PIF allows unplug.

Signature:

```
PIF_ALLOWS_UNPLUG(PIF)
```

PIFALREADYBONDED

This operation cannot be performed because the pif is bonded.

Signature:

```
PIF_ALREADY_BONDED(PIF)
```

PIFBONDMORETHANONEIP

Only one PIF on a bond is allowed to have an IP configuration.

Signature:

```
PIF_BOND_MORE_THAN_ONE_IP()
```

PIFBONDNEEDSMOREMEMBERS

A bond must consist of at least two member interfaces

Signature:

```
PIF_BOND_NEEDS_MORE_MEMBERS()
```

PIFCANNOTBONDCROSSHOST

You cannot bond interfaces across different servers.

Signature:

```
PIF_CANNOT_BOND_CROSS_HOST()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFCONFIGURATIONERROR

An unknown error occurred while attempting to configure an interface.

Signature:

```
PIF_CONFIGURATION_ERROR(PIF, msg)
```

PIFDEVICENOTFOUND

The specified device was not found.

Signature:

```
PIF_DEVICE_NOT_FOUND()
```

PIFDOESNOTALLOWUNPLUG

The operation you requested cannot be performed because the specified PIF does not allow unplug.

Signature:

```
PIF_DOES_NOT_ALLOW_UNPLUG(PIF)
```

PIFHASFCOESRINUSE

The operation you requested cannot be performed because the specified PIF has FCoE SR in use.

Signature:

```
PIF_HAS_FCOE_SR_IN_USE(PIF, SR)
```

PIFHASNONETWORKCONFIGURATION

PIF has no IP configuration mode currently set to 'none'

Signature:

```
PIF_HAS_NO_NETWORK_CONFIGURATION(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFHASNOV6NETWORKCONFIGURATION

PIF has no IPv6 configuration mode currently set to 'none'

Signature:

```
PIF_HAS_NO_V6_NETWORK_CONFIGURATION(PIF)
```

PIFINCOMPATIBLEPRIMARYADDRESSTYPE

The primary address types are not compatible

Signature:

```
PIF_INCOMPATIBLE_PRIMARY_ADDRESS_TYPE(PIF)
```

PIFISMANAGEMENTINTERFACE

The operation you requested cannot be performed because the specified PIF is the management interface.

Signature:

```
PIF_IS_MANAGEMENT_INTERFACE(PIF)
```

PIFISNOTPHYSICAL

You tried to perform an operation which is only available on physical PIF

Signature:

```
PIF_IS_NOT_PHYSICAL(PIF)
```

PIFISNOTSRIOVCAPABLE

The selected PIF is not capable of network SRIOV

Signature:

```
PIF_IS_NOT_SRIOV_CAPABLE(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFISPHYSICAL

You tried to destroy a PIF, but it represents an aspect of the physical host configuration, and so cannot be destroyed. The parameter echoes the PIF handle you gave.

Signature:

```
PIF_IS_PHYSICAL(PIF)
```

PIFISSRIOVLOGICAL

You tried to create a bond on top of a network SRIOV logical PIF use the underlying physical PIF instead

Signature:

```
PIF_IS_SRIOV_LOGICAL(PIF)
```

PIFISVLAN

You tried to create a VLAN on top of another VLAN use the underlying physical PIF/bond instead

Signature:

```
PIF_IS_VLAN(PIF)
```

PIFNOTATTACHEDTOHOST

Clusterhost creation failed as the PIF provided is not attached to the host.

Signature:

```
PIF_NOT_ATTACHED_TO_HOST(pif, host)
```

PIFNOTPRESENT

This host has no PIF on the given network.

Signature:

```
PIF_NOT_PRESENT(host, network)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFSRIOVSTILLEXISTS

The PIF is still related with a network SRIOV

Signature:

```
PIF_SRIOV_STILL_EXISTS(PIF)
```

PIFTUNNELSTILLEXISTS

Operation cannot proceed while a tunnel exists on this interface.

Signature:

```
PIF_TUNNEL_STILL_EXISTS(PIF)
```

PIFUNMANAGED

The operation you requested cannot be performed because the specified PIF is not managed by xapi.

Signature:

```
PIF_UNMANAGED(PIF)
```

PIFVLANEXISTS

You tried to create a PIF, but it already exists.

Signature:

```
PIF_VLAN_EXISTS(PIF)
```

PIFVLANSTILLEXISTS

Operation cannot proceed while a VLAN exists on this interface.

Signature:

```
PIF_VLAN_STILL_EXISTS(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLAUTHALREADYENABLED

External authentication is already enabled for at least one server in this pool.

Signature:

```
POOL_AUTH_ALREADY_ENABLED(host)
```

POOLAUTHDISABLEFAILED

The pool failed to disable the external authentication of at least one host.

Signature:

```
POOL_AUTH_DISABLE_FAILED(host, message)
```

POOLAUTHDISABLEFAILEDINVALIDACCOUNT

External authentication has been disabled with errors: Some AD machine accounts were not disabled on the AD server due to invalid account.

Signature:

```
POOL_AUTH_DISABLE_FAILED_INVALID_ACCOUNT(host, message)
```

POOLAUTHDISABLEFAILEDPERMISSIONDENIED

External authentication has been disabled with errors: Your AD machine account was not disabled on the AD server as permission was denied.

Signature:

```
POOL_AUTH_DISABLE_FAILED_PERMISSION_DENIED(host, message)
```

POOLAUTHDISABLEFAILEDWRONGCREDENTIALS

External authentication has been disabled with errors: Some AD machine accounts were not disabled on the AD server due to invalid credentials.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
POOL_AUTH_DISABLE_FAILED_WRONG_CREDENTIALS(host, message)
```

POOLAUTHENABLEFAILED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED(host, message)
```

POOLAUTHENABLEFAILEDDOMAINLOOKUPFAILED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED(host, message)
```

POOLAUTHENABLEFAILEDDUPLICATEHOSTNAME

The pool failed to enable external authentication, duplicate hostname.

Signature:

```
POOL_AUTH_ENABLE_FAILED_DUPLICATE_HOSTNAME(host, message)
```

POOLAUTHENABLEFAILEDINVALIDACCOUNT

The pool failed to enable external authentication, invalid account.

Signature:

```
POOL_AUTH_ENABLE_FAILED_INVALID_ACCOUNT(host, message)
```

POOLAUTHENABLEFAILEDINVALIDOU

The pool failed to enable external authentication, invalid OU

Signature:

```
POOL_AUTH_ENABLE_FAILED_INVALID_OU(host, message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLAUTHENABLEFAILEDINVALIDTRUSTEDCERTS

The pool failed to enable external authentication, invalid trusted certificates.

Signature:

```
POOL_AUTH_ENABLE_FAILED_INVALID_TRUSTED_CERTS(host, message)
```

POOLAUTHENABLEFAILEDNOSUPPORTENCRYPTTYPE

The pool failed to enable external authentication: domain does not support encryption type, make sure AES based encryption type is enabled in kerberos authentication in AD and Administrator is not in use

Signature:

```
POOL_AUTH_ENABLE_FAILED_NO_SUPPORT_ENCRYPT_TYPE(host, message)
```

POOLAUTHENABLEFAILEDNOTRUSTEDCERTS

The pool failed to enable external authentication, no trusted certificates.

Signature:

```
POOL_AUTH_ENABLE_FAILED_NO_TRUSTED_CERTS(host, message)
```

POOLAUTHENABLEFAILEDPERMISSIONDENIED

The pool failed to enable external authentication, permission denied.

Signature:

```
POOL_AUTH_ENABLE_FAILED_PERMISSION_DENIED(host, message)
```

POOLAUTHENABLEFAILEDSETUPTLSCONNECTION

The pool failed to enable external authentication, failed to set up TLS connection.

Signature:

```
POOL_AUTH_ENABLE_FAILED_SETUP_TLS_CONNECTION(host, message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLAUTHENABLEFAILEDUNAVAILABLE

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_UNAVAILABLE(host, message)
```

POOLAUTHENABLEFAILEDWRONGCREDENTIALS

The pool failed to enable external authentication, wrong credentials.

Signature:

```
POOL_AUTH_ENABLE_FAILED_WRONG_CREDENTIALS(host, message)
```

POOLJOININGEXTERNALAUTHMISMATCH

Cannot join pool whose external authentication configuration is different.

Signature:

```
POOL_JOINING_EXTERNAL_AUTH_MISMATCH()
```

POOLJOININGHOSTCACERTIFICATESCONFLICT

The host joining the pool has different CA certificates from the pool coordinator while using the same name, uninstall them and try again.

Signature:

```
POOL_JOINING_HOST_CA_CERTIFICATES_CONFLICT()
```

POOLJOININGHOSTHASBONDS

The host joining the pool must not have any bonds.

Signature:

```
POOL_JOINING_HOST_HAS_BONDS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGHOSTHASNETWORKSRIOVS

The host joining the pool must not have any network SRIOVs.

Signature:

```
POOL_JOINING_HOST_HAS_NETWORK_SRIOVS()
```

POOLJOININGHOSTHASNONMANAGEMENTVLANS

The host joining the pool must not have any nonmanagement vlans.

Signature:

```
POOL_JOINING_HOST_HAS_NON_MANAGEMENT_VLANS()
```

POOLJOININGHOSTHASTUNNELS

The host joining the pool must not have any tunnels.

Signature:

```
POOL_JOINING_HOST_HAS_TUNNELS()
```

POOLJOININGHOSTMANAGEMENTVLANDOESNOTMATCH

The host joining the pool must have the same management vlan.

Signature:

```
POOL_JOINING_HOST_MANAGEMENT_VLAN_DOES_NOT_MATCH(local, remote)
```

POOLJOININGHOSTMUSTHAVEONLYONEIPONCLUSTERINGNETWORK

The host joining the pool must have one and only one IP on the clustering network

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_ONLY_ONE_IP_ON_CLUSTERING_NETWORK()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGHOSTMUSTHAVEPHYSICALMANAGEMENTNIC

The server joining the pool must have a physical management NIC i.e. the management NIC must not be on a VLAN or bonded PIF.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_PHYSICAL_MANAGEMENT_NIC()
```

POOLJOININGHOSTMUSTHAVESAMEAPIVERSION

The host joining the pool must have the same API version as the pool coordinator.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_SAME_API_VERSION(host_api_version,
master_api_version)
```

POOLJOININGHOSTMUSTHAVESAMEDBSCHEMA

The host joining the pool must have the same database schema as the pool coordinator.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_SAME_DB_SCHEMA(host_db_schema,
master_db_schema)
```

POOLJOININGHOSTMUSTHAVESAMEPRODUCTVERSION

The server joining the pool must have the same product version as the pool coordinator.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_SAME_PRODUCT_VERSION()
```

POOLJOININGHOSTMUSTONLYHAVEPHYSICALPIFS

The host joining the pool must not have any bonds, VLANs or tunnels.

Signature:

```
POOL_JOINING_HOST_MUST_ONLY_HAVE_PHYSICAL_PIFS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGHOSTTLSVERIFICATIONMISMATCH

The TLS verification check failed when the joining host attempted to open a verified connection to the pool coordinator using the imported pool certificate bundle.

Signature:

```
POOL_JOINING_HOST_TLS_VERIFICATION_MISMATCH()
```

POOLJOININGHOSTTRUSTEDCERTIFICATESCONFLICT

The joining host has a trusted certificate identical to one on the pool coordinator but with different purpose. Uninstall it then install it on the host again with the poolcompatible purpose, and try again.

Signature:

```
POOL_JOINING_HOST_TRUSTED_CERTIFICATES_CONFLICT(ref_in_pool, ref_on_host)
```

POOLJOININGMASTERCERTIFICATENOTINPOOLBUNDLE

The pool coordinator's own certificate is absent from the pool certificate bundle sent to the joining host. Run 'xe poolcertificatesync' on the coordinator and retry.

Signature:

```
POOL_JOINING_MASTER_CERTIFICATE_NOT_IN_POOL_BUNDLE(master_uuid)
```

POOLJOININGPOOLBUNDLEEMPTYAFTERIMPORT

The pool certificate bundle is empty or missing after import on the joining host. The bundle generation script updatecabundle.sh likely failed silently.

Signature:

```
POOL_JOINING_POOL_BUNDLE_EMPTY_AFTER_IMPORT(bundle_path)
```

POOLJOININGPOOLCANNOTENABLECLUSTERINGONVLANNETWORK

The host attempted to join a pool with clustering enabled on a nonmanagement VLAN network, which is not supported.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
POOL_JOINING_POOL_CANNOT_ENABLE_CLUSTERING_ON_VLAN_NETWORK(vlan)
```

POOLJOININGSMFEATURESINCOMPATIBLE

The host joining the pool has an incompatible set of sm features from the pool coordinator. Make sure the sm are of the same versions and try again.

Signature:

```
POOL_JOINING_SM_FEATURES_INCOMPATIBLE(pool_sm_ref, candidate_sm_ref)
```

PROVISIONFAILEDOUTOFSPACE

The provision call failed because it ran out of space.

Signature:

```
PROVISION_FAILED_OUT_OF_SPACE()
```

PROVISIONONLYALLOWEDONTEMPLATE

The provision call can only be invoked on templates, not regular VMs.

Signature:

```
PROVISION_ONLY_ALLOWED_ON_TEMPLATE()
```

PUSBVDICONFLICT

The VDI corresponding to this PUSB has existing VBDs.

Signature:

```
PUSB_VDI_CONFLICT(PUSB, VDI)
```

PVSCACHESTORAGEALREADYPRESENT

The PVS site already has cache storage configured for the host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_CACHE_STORAGE_ALREADY_PRESENT(site, host)
```

PVSCACHESTORAGEISINUSE

The PVS cache storage is in use by the site and cannot be removed.

Signature:

```
PVS_CACHE_STORAGE_IS_IN_USE(PVS_cache_storage)
```

PVSPROXYALREADYPRESENT

The VIF is already associated with a PVS proxy

Signature:

```
PVS_PROXY_ALREADY_PRESENT(proxies)
```

PVSPROXYPRESENTONHIGHERVIFDEVICE

The VM has a VIF, with a higher device number than the new VIF, that uses a PVS proxy. The VIF used by PVS proxy must be the one with the lowest device number.

Signature:

```
PVS_PROXY_PRESENT_ON_HIGHER_VIF_DEVICE(device)
```

PVSSERVERADDRESSINUSE

The address specified is already in use by an existing PVSserver object

Signature:

```
PVS_SERVER_ADDRESS_IN_USE(address)
```

PVSSITECONTAINSRUNNINGPROXIES

The PVS site contains running proxies.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_SITE_CONTAINS_RUNNING_PROXIES(proxies)
```

PVSSITECONTAINSSERVERS

The PVS site contains servers and cannot be forgotten.

Signature:

```
PVS_SITE_CONTAINS_SERVERS(servers)
```

PVSVIFMUSTBEFIRSTDEVICE

The VIF used by PVS proxy must be the one with the lowest device number

Signature:

```
PVS_VIF_MUST_BE_FIRST_DEVICE()
```

RBACPERMISSIONDENIED

RBAC permission denied.

Signature:

```
RBAC_PERMISSION_DENIED(permission, message)
```

REDOLOGISENABLED

The operation could not be performed because a redo log is enabled on the Pool.

Signature:

```
REDO_LOG_IS_ENABLED()
```

REPOSITORYALREADYEXISTS

The repository already exists.

Signature:

```
REPOSITORY_ALREADY_EXISTS(ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

REPOSITORYCLEANUPFAILED

Failed to clean up local repository on coordinator.

Signature:

```
REPOSITORY_CLEANUP_FAILED()
```

REPOSITORYISINUSE

The repository is in use.

Signature:

```
REPOSITORY_IS_IN_USE()
```

REPOSYNCFAILED

Syncing with remote YUM repository failed.

Signature:

```
REPOSYNC_FAILED()
```

REPOSHOULDBESINGLEONEENABLED

If the bundle repository or remotepool repository is enabled, it should be the only enabled repository of the pool.

Signature:

```
REPO_SHOULD_BE_SINGLE_ONE_ENABLED()
```

REQUIREDPIFISUNPLUGGED

The operation you requested cannot be performed because the specified PIF is currently unplugged.

Signature:

```
REQUIRED_PIF_IS_UNPLUGGED(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RESTOREINCOMPATIBLEVERSION

The restore could not be performed because this backup has been created by a different incompatible product version

Signature:

```
RESTORE_INCOMPATIBLE_VERSION()
```

RESTORESCRIPTFAILED

The restore could not be performed because the restore script failed. Is the file corrupt?

Signature:

```
RESTORE_SCRIPT_FAILED(log)
```

RESTORETARGETMGMTIFNOTINBACKUP

The restore could not be performed because the server's current management interface is not in the backup. The interfaces mentioned in the backup are:

Signature:

```
RESTORE_TARGET_MGMT_IF_NOT_IN_BACKUP()
```

RESTORETARGETMISSINGDEVICE

The restore could not be performed because a network interface is missing

Signature:

```
RESTORE_TARGET_MISSING_DEVICE(device)
```

ROLEALREADYEXISTS

Role already exists.

Signature:

```
ROLE_ALREADY_EXISTS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ROLENOTFOUND

Role cannot be found.

Signature:

```
ROLE_NOT_FOUND()
```

SERVERCERTIFICATECHAININVALID

The provided intermediate certificates are not in a PEMencoded X509 format.

Signature:

```
SERVER_CERTIFICATE_CHAIN_INVALID()
```

SERVERCERTIFICATEEXPIRED

The provided certificate has expired.

Signature:

```
SERVER_CERTIFICATE_EXPIRED(now, not_after)
```

SERVERCERTIFICATEINVALID

The provided certificate is not in a PEMencoded X509 format.

Signature:

```
SERVER_CERTIFICATE_INVALID()
```

SERVERCERTIFICATEKEYALGORITHMNOTSUPPORTED

The provided key uses an unsupported algorithm.

Signature:

```
SERVER_CERTIFICATE_KEY_ALGORITHM_NOT_SUPPORTED(algorithm_oid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SERVERCERTIFICATEKEYINVALID

The provided key is not in a PEMencoded PKCS8 format.

Signature:

```
SERVER_CERTIFICATE_KEY_INVALID()
```

SERVERCERTIFICATEKEYMISMATCH

The provided key does not match the provided certificate's public key.

Signature:

```
SERVER_CERTIFICATE_KEY_MISMATCH()
```

SERVERCERTIFICATEKEYRSALENGTHNOTSUPPORTED

The provided RSA key does not have a length between 2048 and 4096.

Signature:

```
SERVER_CERTIFICATE_KEY_RSA_LENGTH_NOT_SUPPORTED(length)
```

SERVERCERTIFICATEKEYRSAMULTINOTSUPPORTED

The provided RSA key is using more than 2 primes, expecting only 2.

Signature:

```
SERVER_CERTIFICATE_KEY_RSA_MULTI_NOT_SUPPORTED()
```

SERVERCERTIFICATENOTVALIDYET

The provided certificate is not valid yet.

Signature:

```
SERVER_CERTIFICATE_NOT_VALID_YET(now, not_before)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SERVERCERTIFICATESIGNATURENOTSUPPORTED

The provided certificate is not using one of the following SHA2 signature algorithms: SHA256, SHA512.

Signature:

```
SERVER_CERTIFICATE_SIGNATURE_NOT_SUPPORTED()
```

SESSIONAUTHENTICATIONFAILED

The credentials given by the user are incorrect, so access has been denied, and you have not been issued a session handle.

Signature:

```
SESSION_AUTHENTICATION_FAILED()
```

SESSIONAUTHORIZATIONFAILED

The credentials given by the user are correct, but the user could not be authorized, so access has been denied, and you have not been issued a session handle.

Signature:

```
SESSION_AUTHORIZATION_FAILED(username, msg)
```

SESSIONINVALID

You gave an invalid session reference. It may have been invalidated by a server restart, or timed out. You should get a new session handle, using one of the session.login calls. This error does not invalidate the current connection. The handle parameter echoes the bad value given.

Signature:

```
SESSION_INVALID(handle)
```

SESSIONNOTREGISTERED

This session is not registered to receive events. You must call event.register before event.next. The session handle you are using is echoed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
SESSION_NOT_REGISTERED(handle)
```

SETCONSOLETIMEOUTPARTIALLYFAILED

Some hosts failed to set console timeout.

Signature:

```
SET_CONSOLE_TIMEOUT_PARTIALLY_FAILED(hosts)
```

SETSSHAUTOMODEPARTIALLYFAILED

Some hosts failed to set SSH auto mode.

Signature:

```
SET_SSH_AUTO_MODE_PARTIALLY_FAILED(hosts)
```

SETSSHTIMEOUTPARTIALLYFAILED

Some hosts failed to set SSH timeout.

Signature:

```
SET_SSH_TIMEOUT_PARTIALLY_FAILED(hosts)
```

SLAVEREQUIRESMANAGEMENTINTERFACE

The management interface on a supporter cannot be disabled because the supporter would enter emergency mode.

Signature:

```
SLAVE_REQUIRES_MANAGEMENT_INTERFACE()
```

SMPLUGINCOMMUNICATIONFAILURE

The SM plugin did not respond to a query.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SM_PLUGIN_COMMUNICATION_FAILURE(sm)
```

SRATTACHFAILED

Attaching this SR failed.

Signature:

```
SR_ATTACH_FAILED(sr)
```

SRBACKENDFAILURE

There was an SR backend failure.

Signature:

```
SR_BACKEND_FAILURE(status, stdout, stderr)
```

SRDEVICEINUSE

The SR operation cannot be performed because a device underlying the SR is in use by the server.

Signature:

```
SR_DEVICE_IN_USE()
```

SRDOESNOTSUPPORTMIGRATION

Cannot migrate a VDI to or from an SR that doesn't support migration.

Signature:

```
SR_DOES_NOT_SUPPORT_MIGRATION(sr, capabilities)
```

SRFULL

The SR is full. Requested new size exceeds the maximum size

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR_FULL(requested, maximum)
```

SRHASMULTIPLEPBDS

The SR.shared flag cannot be set to false while the SR remains connected to multiple servers.

Signature:

```
SR_HAS_MULTIPLE_PBDS(PBD)
```

SRHASNOPBDS

The SR has no attached PBDs

Signature:

```
SR_HAS_NO_PBDS(sr)
```

SRHASPBD

The SR is still connected to a host via a PBD. It cannot be destroyed or forgotten.

Signature:

```
SR_HAS_PBD(sr)
```

SRINDESTRUCTIBLE

The SR could not be destroyed because the 'indestructible' flag was set on it.

Signature:

```
SR_INDESTRUCTIBLE(sr)
```

SRISCACHESR

The SR is currently being used as a local cache SR.

Signature:

```
SR_IS_CACHE_SR(host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SRNOTATTACHED

The SR is not attached.

Signature:

```
SR_NOT_ATTACHED(sr)
```

SRNOTEMPTY

The SR operation cannot be performed because the SR is not empty.

Signature:

```
SR_NOT_EMPTY()
```

SRNOTSHARABLE

The PBD could not be plugged because the SR is in use by another host and is not marked as sharable.

Signature:

```
SR_NOT_SHARABLE(sr, host)
```

SROPERATIONNOTSUPPORTED

The SR backend does not support the operation check the SR's allowed operations

Signature:

```
SR_OPERATION_NOT_SUPPORTED(sr)
```

SRREQUIRESUPGRADE

The operation cannot be performed until the SR has been upgraded

Signature:

```
SR_REQUIRES_UPGRADE(SR)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SRSOURCESPACEINSUFFICIENT

The source SR does not have sufficient temporary space available to proceed the operation.

Signature:

```
SR_SOURCE_SPACE_INSUFFICIENT(sr)
```

SRSUSPENDSPACEINSUFFICIENT

The suspend SR does not have sufficient free space to store the VM suspend image.

Signature:

```
SR_SUSPEND_SPACE_INSUFFICIENT(sr)
```

SRUNHEALTHY

The SR is currently unhealthy. See the suggestion on how to fix it.

Signature:

```
SR_UNHEALTHY(sr, health, fix)
```

SRUNKNOWNDRIVER

The SR could not be connected because the driver was not recognised.

Signature:

```
SR_UNKNOWN_DRIVER(driver)
```

SRUUIDEXISTS

An SR with that uuid already exists.

Signature:

```
SR_UUID_EXISTS(uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SRVDILOCKINGFAILED

The operation could not proceed because necessary VDIs were already locked at the storage level.

Signature:

```
SR_VDI_LOCKING_FAILED()
```

SSLVERIFYERROR

The remote system's SSL certificate failed to verify against our certificate library.

Signature:

```
SSL_VERIFY_ERROR(reason)
```

SUBJECTALREADYEXISTS

Subject already exists.

Signature:

```
SUBJECT_ALREADY_EXISTS()
```

SUBJECTCANNOTBERESOLVED

Subject cannot be resolved by the external directory service.

Signature:

```
SUBJECT_CANNOT_BE_RESOLVED()
```

SUSPENDIMAGENOTACCESSIBLE

The suspend image of a checkpoint is not accessible from the host on which the VM is running

Signature:

```
SUSPEND_IMAGE_NOT_ACCESSIBLE(vdi)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SYNCBUNDLEINPROGRESS

The operation could not be performed because syncing bundle is in progress.

Signature:

```
SYNC_BUNDLE_IN_PROGRESS()
```

SYNCUPDATESINPROGRESS

The operation could not be performed because syncing updates is in progress.

Signature:

```
SYNC_UPDATES_IN_PROGRESS()
```

SYSPREP

VM.sysprep error with details in the message

Signature:

```
SYSPREP(vm, message)
```

SYSTEMSTATUSMUSTUSETARONOEM

You must use tar output to retrieve system status from an OEM server.

Signature:

```
SYSTEM_STATUS_MUST_USE_TAR_ON_OEM()
```

SYSTEMSTATUSRETRIEVALFAILED

Retrieving system status from the host failed. A diagnostic reason suitable for support organisations is also returned.

Signature:

```
SYSTEM_STATUS_RETRIEVAL_FAILED(reason)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TASKCANCELLED

The request was asynchronously canceled.

Signature:

```
TASK_CANCELLED(task)
```

TELEMETRYNEXTCOLLECTIONTOOLATE

The next scheduled telemetry data collection is too far into the future. Pick a timestamp within two telemetry intervals starting from now.

Signature:

```
TELEMETRY_NEXT_COLLECTION_TOO_LATE(timestamp)
```

TLSCONNECTIONFAILED

Cannot contact the other host using TLS on the specified address and port

Signature:

```
TLS_CONNECTION_FAILED(address, port)
```

TLSVERIFICATIONNOTENABLEDINPOOL

TLS verification has not been enabled in the pool successfully, please enable it in XC or run xe poolenabletlsverification instead.

Signature:

```
TLS_VERIFICATION_NOT_ENABLED_IN_POOL()
```

TOOBUSY

The request was rejected because the server is too busy.

Signature:

```
TOO_BUSY()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TOOMANYGROUPS

VM can only belong to one group.

Signature:

```
TOO_MANY_GROUPS()
```

TOOMANYPENDINGTASKS

The request was rejected because there are too many pending tasks on the server.

Signature:

```
TOO_MANY_PENDING_TASKS()
```

TOOMANYSTORAGEMIGRATES

You reached the maximal number of concurrently migrating VMs.

Signature:

```
TOO_MANY_STORAGE_MIGRATES(number)
```

TOOMANYVUSBS

The VM has too many VUSBs.

Signature:

```
TOO_MANY_VUSBS(number)
```

TRANSPORTPIFNOTCONFIGURED

The tunnel transport PIF has no IP configuration set.

Signature:

```
TRANSPORT_PIF_NOT_CONFIGURED(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TRUSTEDCERTIFICATEALREADYEXISTS

A trusted certificate already exists with the same purpose.

Signature:

```
TRUSTED_CERTIFICATE_ALREADY_EXISTS(fingerprint)
```

TRUSTEDCERTIFICATEEXPIRED

The provided certificate has expired.

Signature:

```
TRUSTED_CERTIFICATE_EXPIRED(now, not_after)
```

TRUSTEDCERTIFICATEINVALID

The provided certificate is not in a PEMencoded X509 format.

Signature:

```
TRUSTED_CERTIFICATE_INVALID()
```

TRUSTEDCERTIFICATENOTVALIDYET

The provided certificate is not valid yet.

Signature:

```
TRUSTED_CERTIFICATE_NOT_VALID_YET(now, not_before)
```

UNIMPLEMENTEDINSMBACKEND

You have attempted a function which is not implemented

Signature:

```
UNIMPLEMENTED_IN_SM_BACKEND(message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UNKNOWNBOOTLOADER

The requested bootloader is unknown

Signature:

```
UNKNOWN_BOOTLOADER(vm, bootloader)
```

UPDATEINFOHASHMISMATCH

The hash of updateinfo doesn't match with current one. There may be newer available updates.

Signature:

```
UPDATEINFO_HASH_MISMATCH()
```

UPDATEALREADYAPPLIED

This update has already been applied.

Signature:

```
UPDATE_ALREADY_APPLIED(update)
```

UPDATEALREADYAPPLIEDINPOOL

This update has already been applied to all hosts in the pool.

Signature:

```
UPDATE_ALREADY_APPLIED_IN_POOL(update)
```

UPDATEALREADYEXISTS

The uploaded update already exists

Signature:

```
UPDATE_ALREADY_EXISTS(uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATEAPPLYFAILED

The update failed to apply. Please see attached output.

Signature:

```
UPDATE_APPLY_FAILED(output)
```

UPDATEISAPPLIED

The specified update has been applied and cannot be destroyed.

Signature:

```
UPDATE_IS_APPLIED()
```

UPDATEPOOLAPPLYFAILED

The update cannot be applied for the following hosts.

Signature:

```
UPDATE_POOL_APPLY_FAILED(hosts)
```

UPDATEPRECHECKFAILEDCONFLICTPRESENT

The update precheck stage failed: conflicting updates are present.

Signature:

```
UPDATE_PRECHECK_FAILED_CONFLICT_PRESENT(update, conflict_update)
```

UPDATEPRECHECKFAILEDGPGKEYNOTIMPORTED

The update precheck stage failed: RPM package validation requires a GPG key that is not present on the host.

Signature:

```
UPDATE_PRECHECK_FAILED_GPGKEY_NOT_IMPORTED(update)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATEPRECHECKFAILEDOUTOFSPACE

The update precheck stage failed: the server does not have enough space.

Signature:

```
UPDATE_PRECHECK_FAILED_OUT_OF_SPACE(update, available_space,
required_space )
```

UPDATEPRECHECKFAILEDPREREQUISITEMISSING

The update precheck stage failed: prerequisite updates are missing.

Signature:

```
UPDATE_PRECHECK_FAILED_PREREQUISITE_MISSING(update, prerequisite_update)
```

UPDATEPRECHECKFAILEDUNKNOWNERROR

The update precheck stage failed with an unknown error.

Signature:

```
UPDATE_PRECHECK_FAILED_UNKNOWN_ERROR(update, info)
```

UPDATEPRECHECKFAILEDWRONGSERVERVERSION

The update precheck stage failed: the server is of an incorrect version.

Signature:

```
UPDATE_PRECHECK_FAILED_WRONG_SERVER_VERSION(update, installed_version,
required_version )
```

UPDATESYNCINGREMOTEPOOLCOORDINATORCONNECTIONFAILED

There was an error connecting to the remote pool coordinator while syncing updates from it.

Signature:

```
UPDATE_SYNCING_REMOTE_POOL_COORDINATOR_CONNECTION_FAILED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATESYNCINGREMOTEPOOLCOORDINATORSERVICEFAILED

There was an error connecting to the server while syncing updates from it. The service contacted didn't reply properly.

Signature:

```
UPDATE_SYNCING_REMOTE_POOL_COORDINATOR_SERVICE_FAILED()
```

USBALREADYATTACHED

The USB device is currently attached to a VM.

Signature:

```
USB_ALREADY_ATTACHED(PUSB, VM)
```

USBGROUPCONFLICT

USBgroups are currently restricted to contain no more than one VUSB.

Signature:

```
USB_GROUP_CONFLICT(USB_group)
```

USBGROUPCONTAINSNOPUSBS

The USB group does not contain any PUSBs.

Signature:

```
USB_GROUP_CONTAINS_NO_PUSBS(usb_group)
```

USBGROUPCONTAINSPUSB

The USB group contains active PUSBs and cannot be deleted.

Signature:

```
USB_GROUP_CONTAINS_PUSB(pusbs)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

USBGROUPCONTAINSVUSB

The USB group contains active VUSBs and cannot be deleted.

Signature:

```
USB_GROUP_CONTAINS_VUSB(vusbs)
```

USERISNOTLOCALSUPERUSER

Only the local superuser can perform this operation.

Signature:

```
USER_IS_NOT_LOCAL_SUPERUSER(msg)
```

UUIDINVALID

The uuid you supplied was invalid.

Signature:

```
UUID_INVALID(type, uuid)
```

V6DFAILURE

There was a problem with the license daemon v6d.

Signature:

```
V6D_FAILURE()
```

VALUENOTSUPPORTED

You attempted to set a value that is not supported by this implementation. The fullyqualified field name and the value that you tried to set are returned. Also returned is a developeronly diagnostic reason.

Signature:

```
VALUE_NOT_SUPPORTED(field, value, reason)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VBDCDSMUSTBEREADONLY

Read/write CDs are not supported

Signature:

```
VBD_CDS_MUST_BE_READONLY()
```

VBDISEMPTY

Operation could not be performed because the drive is empty

Signature:

```
VBD_IS_EMPTY(vbd)
```

VBDNOTEMPTY

Operation could not be performed because the drive is not empty

Signature:

```
VBD_NOT_EMPTY(vbd)
```

VBDNOTREMOVABLEMEDIA

Media could not be ejected because it is not removable

Signature:

```
VBD_NOT_REMOVABLE_MEDIA(vbd)
```

VBDNOTUNPLUGGABLE

Drive could not be hotunplugged because it is not marked as unpluggable

Signature:

```
VBD_NOT_UNPLUGGABLE(vbd)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VBDTRAYLOCKED

This VM has locked the DVD drive tray, so the disk cannot be ejected

Signature:

```
VBD_TRAY_LOCKED(vbd)
```

VCPUMAXNOTCORESPERSOCKETMULTIPLE

VCPUsmax must be a multiple of corespersocket

Signature:

```
VCPU_MAX_NOT_CORES_PER_SOCKET_MULTIPLE(vcpu_max, cores_per_socket)
```

VDICBTENABLED

The requested operation is not allowed for VDIs with CBT enabled or VMs having such VDIs, and CBT is enabled for the specified VDI.

Signature:

```
VDI_CBT_ENABLED(vdi)
```

VDICONTAINSMETADATAOFTHISPOOL

The VDI could not be opened for metadata recovery as it contains the current pool's metadata.

Signature:

```
VDI_CONTAINS_METADATA_OF_THIS_POOL(vdi, pool)
```

VDICOPYFAILED

The VDI copy action has failed

Signature:

```
VDI_COPY_FAILED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDIHASRRDS

The operation cannot be performed because this VDI has rrd stats

Signature:

```
VDI_HAS_RRDS(vdi)
```

VDIINCOMPATIBLETYPE

This operation cannot be performed because the specified VDI is of an incompatible type eg: an HA statefile cannot be attached to a guest

Signature:

```
VDI_INCOMPATIBLE_TYPE(vdi, type)
```

VDIINUSE

This operation cannot be performed because this VDI is in use by some other operation

Signature:

```
VDI_IN_USE(vdi, operation)
```

VDIISAPHYSICALDEVICE

The operation cannot be performed on physical device

Signature:

```
VDI_IS_A_PHYSICAL_DEVICE(vdi)
```

VDIISENCRYPTED

The requested operation is not allowed because the specified VDI is encrypted.

Signature:

```
VDI_IS_ENCRYPTED(vdi)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDIISNOTISO

This operation can only be performed on CD VDIs iso files or CDROM drives

Signature:

```
VDI_IS_NOT_ISO(vdi, type)
```

VDILOCATIONMISSING

This operation cannot be performed because the specified VDI could not be found in the specified SR

Signature:

```
VDI_LOCATION_MISSING(sr, location)
```

VDIMISSING

This operation cannot be performed because the specified VDI could not be found on the storage substrate

Signature:

```
VDI_MISSING(sr, vdi)
```

VDINEEDSVMFORMIGRATE

Cannot migrate a VDI which is not attached to a running VM.

Signature:

```
VDI_NEEDS_VM_FOR_MIGRATE(vdi)
```

VDINOTAVAILABLE

This operation cannot be performed because this VDI could not be properly attached to the VM.

Signature:

```
VDI_NOT_AVAILABLE(vdi)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDINOTINMAP

This VDI was not mapped to a destination SR in VM.migratesend operation

Signature:

```
VDI_NOT_IN_MAP(vdi)
```

VDINOTMANAGED

This operation cannot be performed because the system does not manage this VDI

Signature:

```
VDI_NOT_MANAGED(vdi)
```

VDINOTSPARSE

The VDI is not stored using a sparse format. It is not possible to query and manipulate only the changed blocks or 'block differences' or 'disk deltas' between two VDIs. Please select a VDI which uses a sparseaware technology such as VHD.

Signature:

```
VDI_NOT_SPARSE(vdi)
```

VDINOCBTMETADATA

The requested operation is not allowed because the specified VDI does not have changed block tracking metadata.

Signature:

```
VDI_NO_CBT_METADATA(vdi)
```

VDIONBOOTMODEINCOMPATIBLEWITHOPERATION

This operation is not permitted on VDIs in the 'onboot=reset' mode, or on VMs having such VDIs.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VDI_ON_BOOT_MODE_INCOMPATIBLE_WITH_OPERATION()
```

VDIREADONLY

The operation required write access but this VDI is readonly

Signature:

```
VDI_READONLY(vdi)
```

VDITOOLARGE

The VDI is too large.

Signature:

```
VDI_TOO_LARGE(vdi, maximum size)
```

VDITOOSMALL

The VDI is too small. Please resize it to at least the minimum size.

Signature:

```
VDI_TOO_SMALL(vdi, minimum size)
```

VGPUDESTINATIONINCOMPATIBLE

The VGPU is not compatible with any PGPU in the destination.

Signature:

```
VGPU_DESTINATION_INCOMPATIBLE(reason, vgpu, host)
```

VGPUGUESTDRIVERLIMIT

The guest driver does not support VGPU migration.

Signature:

```
VGPU_GUEST_DRIVER_LIMIT(reason, vm, host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VGPUSUSPENSIONNOTSUPPORTED

The VGPU configuration does not support suspension.

Signature:

```
VGPU_SUSPENSION_NOT_SUPPORTED(reason, vgpu, host)
```

VGPUTYPENOTCOMPATIBLE

Cannot create a virtual GPU that is incompatible with the existing types on the VM.

Signature:

```
VGPU_TYPE_NOT_COMPATIBLE(type)
```

VGPUTYPENOTCOMPATIBLEWITHRUNNINGTYPE

The VGPU type is incompatible with one or more of the VGPU types currently running on this PGPU

Signature:

```
VGPU_TYPE_NOT_COMPATIBLE_WITH_RUNNING_TYPE(pgpu, type, running_type)
```

VGPUTYPENOTENABLED

VGPU type is not one of the PGPU's enabled types.

Signature:

```
VGPU_TYPE_NOT_ENABLED(type, enabled_types)
```

VGPUTYPENOTSUPPORTED

VGPU type is not one of the PGPU's supported types.

Signature:

```
VGPU_TYPE_NOT_SUPPORTED(type, supported_types)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VGPUTYPENOLONGERSUPPORTED

VGPU type is no longer supported

Signature:

```
VGPU_TYPE_NO_LONGER_SUPPORTED(type)
```

VIFINUSE

Network has active VIFs

Signature:

```
VIF_IN_USE(network, VIF)
```

VIFNOTINMAP

This VIF was not mapped to a destination Network in VM.migratesend operation

Signature:

```
VIF_NOT_IN_MAP(vif)
```

VLANINUSE

Operation cannot be performed because this VLAN is already in use. Please check your network configuration.

Signature:

```
VLAN_IN_USE(device, vlan)
```

VLANTAGINVALID

You tried to create a VLAN, but the tag you gave was invalid it must be between 0 and 4094. The parameter echoes the VLAN tag you gave.

Signature:

```
VLAN_TAG_INVALID(VLAN)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMPPARCHIVEMOREFREQUENTTHANBACKUP

Archive more frequent than backup.

Signature:

```
VMPP_ARCHIVE_MORE_FREQUENT_THAN_BACKUP()
```

VMPPHASVM

There is at least one VM assigned to this protection policy.

Signature:

```
VMPP_HAS_VM()
```

VMSSHASVM

There is at least one VM assigned to snapshot schedule.

Signature:

```
VMSS_HAS_VM()
```

VMSFAILEDTOCOOPERATE

The given VMs failed to release memory when instructed to do so

Signature:

```
VMS_FAILED_TO_COOPERATE()
```

VMASSIGNEDTOPROTECTIONPOLICY

This VM is assigned to a protection policy.

Signature:

```
VM_ASSIGNED_TO_PROTECTION_POLICY(vm, vmpp)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMASSIGNEDTOSNAPSHOTSCHEDULE

This VM is assigned to a snapshot schedule.

Signature:

```
VM_ASSIGNED_TO_SNAPSHOT_SCHEDULE(vm, vmss)
```

VMATTACHEDTOMORETHANONEVDIWITHTIMEOFFSETMARKEDASRESETONBOOT

You attempted to start a VM that's attached to more than one VDI with a timeoffset marked as resetonboot.

Signature:

```
VM_ATTACHED_TO_MORE_THAN_ONE_VDI_WITH_TIMEOFFSET_MARKED_AS_RESET_ON_BOOT(v
m)
```

VMBADPOWERSTATE

You attempted an operation on a VM that was not in an appropriate power state at the time; for example, you attempted to start a VM that was already running. The parameters returned are the VM's handle, and the expected and actual VM state at the time of the call.

Signature:

```
VM_BAD_POWER_STATE(vm, expected, actual)
```

VMBIOSSTRINGSALREADYSET

The BIOS strings for this VM have already been set and cannot be changed.

Signature:

```
VM_BIOS_STRINGS_ALREADY_SET()
```

VMCALLPLUGINRATELIMIT

There is a minimal interval required between consecutive plugin calls made on the same VM, please wait before retry.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_CALL_PLUGIN_RATE_LIMIT(VM, interval, wait)
```

VMCANNOTDELETEDEFAULTTEMPLATE

You cannot delete the specified default template.

Signature:

```
VM_CANNOT_DELETE_DEFAULT_TEMPLATE(vm)
```

VMCHECKPOINTRESUMEFAILED

An error occured while restoring the memory image of the specified virtual machine

Signature:

```
VM_CHECKPOINT_RESUME_FAILED(vm)
```

VMCHECKPOINTSUSPENDFAILED

An error occured while saving the memory image of the specified virtual machine

Signature:

```
VM_CHECKPOINT_SUSPEND_FAILED(vm)
```

VMCRASHED

The VM crashed

Signature:

```
VM_CRASHED(vm)
```

VMDUPLICATEVBDDEVICE

The specified VM has a duplicate VBD device and cannot be started.

Signature:

```
VM_DUPLICATE_VBD_DEVICE(vm, vbd, device)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMFAILEDSHUTDOWNACKNOWLEDGMENT

VM didn't acknowledge the need to shutdown.

Signature:

```
VM_FAILED_SHUTDOWN_ACKNOWLEDGMENT(vm)
```

VMFAILEDSUSPENDACKNOWLEDGMENT

VM didn't acknowledge the need to suspend.

Signature:

```
VM_FAILED_SUSPEND_ACKNOWLEDGMENT(vm)
```

VMHALTED

The VM unexpectedly halted

Signature:

```
VM_HALTED(vm)
```

VMHASCHECKPOINT

Cannot migrate a VM which has a checkpoint.

Signature:

```
VM_HAS_CHECKPOINT(vm)
```

VMHASNOSUSPENDVDI

VM cannot be resumed because it has no suspend VDI

Signature:

```
VM_HAS_NO_SUSPEND_VDI(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMHASPCIATTACHED

This operation could not be performed, because the VM has one or more PCI devices passed through.

Signature:

```
VM_HAS_PCI_ATTACHED(vm)
```

VMHASSRIOVVIF

This operation could not be performed, because the VM has one or more SRIOV VIFs.

Signature:

```
VM_HAS_SRIOV_VIF(vm)
```

VMHASTOOMANYSNAPSHOTS

Cannot migrate a VM with more than one snapshot.

Signature:

```
VM_HAS_TOO_MANY_SNAPSHOTS(vm)
```

VMHASVGPU

This operation could not be performed, because the VM has one or more virtual GPUs.

Signature:

```
VM_HAS_VGPU(vm)
```

VMHASVUSBS

The operation is not allowed when the VM has VUSBs.

Signature:

```
VM_HAS_VUSBS(VM)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMHOSTINCOMPATIBLEVERSION

This VM operation cannot be performed on an olderversioned host during an upgrade.

Signature:

```
VM_HOST_INCOMPATIBLE_VERSION(host, vm)
```

VMHOSTINCOMPATIBLEVERSIONMIGRATE

Cannot migrate a VM to a destination host which is older than the source host.

Signature:

```
VM_HOST_INCOMPATIBLE_VERSION_MIGRATE(host, vm)
```

VMHOSTINCOMPATIBLEVIRTUALHARDWAREPLATFORMVERSION

You attempted to run a VM on a host that cannot provide the VM's required Virtual Hardware Platform version.

Signature:

```
VM_HOST_INCOMPATIBLE_VIRTUAL_HARDWARE_PLATFORM_VERSION(host,
host_versions, vm, vm_version)
```

VMHVMREQUIRED

HVM is required for this operation

Signature:

```
VM_HVM_REQUIRED(vm)
```

VMINCOMPATIBLEWITHTHISHOST

The VM is incompatible with the CPU features of this host.

Signature:

```
VM_INCOMPATIBLE_WITH_THIS_HOST(vm, host, reason)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMISIMMOBILE

The VM is configured in a way that prevents it from being mobile.

Signature:

```
VM_IS_IMMOBILE(VM)
```

VMISPARTOFANAPPLIANCE

This operation is not allowed as the VM is part of an appliance.

Signature:

```
VM_IS_PART_OF_AN_APPLIANCE(vm, appliance)
```

VMISPROTECTED

This operation cannot be performed because the specified VM is protected by HA

Signature:

```
VM_IS_PROTECTED(vm)
```

VMISTEMPLATE

The operation attempted is not valid for a template VM

Signature:

```
VM_IS_TEMPLATE(vm)
```

VMISUSINGNESTEDVIRT

This operation is illegal because the VM is using nested virtualization.

Signature:

```
VM_IS_USING_NESTED_VIRT(VM)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMLACKSFEATURE

You attempted an operation on a VM which lacks the feature.

Signature:

```
VM_LACKS_FEATURE(vm)
```

VMMEMORYSIZETOOLOW

The specified VM has too little memory to be started.

Signature:

```
VM_MEMORY_SIZE_TOO_LOW(vm)
```

VMMIGRATECONTACTREMOTESERVICEFAILED

Failed to contact service on the destination host.

Signature:

```
VM_MIGRATE_CONTACT_REMOTE_SERVICE_FAILED()
```

VMMIGRATEFAILED

An error occurred during the migration process.

Signature:

```
VM_MIGRATE_FAILED(vm, source, destination, msg)
```

VMMISSINGPVDRIVERS

You attempted an operation on a VM which requires PV drivers to be installed but the drivers were not detected.

Signature:

```
VM_MISSING_PV_DRIVERS(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMNONSUSPENDABLE

You attempted an operation on a VM which is not suspendable.

Signature:

```
VM_NON_SUSPENDABLE(vm, reason)
```

VMNOTRESIDENTHERE

The specified VM is not currently resident on the specified server.

Signature:

```
VM_NOT_RESIDENT_HERE(vm, host)
```

VMNOCRASHDUMPSR

This VM does not have a crash dump SR specified.

Signature:

```
VM_NO_CRASHDUMP_SR(vm)
```

VMNOEMPTYCDVBD

The VM has no empty CD drive VBD.

Signature:

```
VM_NO_EMPTY_CD_VBD(vm)
```

VMNOSUSPENDSR

This VM does not have a suspend SR specified.

Signature:

```
VM_NO_SUSPEND_SR(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMNOVCPUS

You need at least 1 VCPU to start a VM

Signature:

```
VM_NO_VCPUS(vm)
```

VMOLDPVDRIVERS

You attempted an operation on a VM which requires a more recent version of the PV drivers. Please upgrade your PV drivers.

Signature:

```
VM_OLD_PV_DRIVERS(vm, major, minor)
```

VMPCIBUSFULL

The VM does not have any free PCI slots

Signature:

```
VM_PCI_BUS_FULL(VM)
```

VMPVDRIVERSINUSE

VM PV drivers still in use

Signature:

```
VM_PV_DRIVERS_IN_USE(vm)
```

VMREBOOTED

The VM unexpectedly rebooted

Signature:

```
VM_REBOOTED(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMREQUIRESGPU

You attempted to run a VM on a host which doesn't have a pGPU available in the GPU group needed by the VM. The VM has a vGPU attached to this GPU group.

Signature:

```
VM_REQUIRES_GPU(vm, GPU_group)
```

VMREQUIRESIOMMU

You attempted to run a VM on a host which doesn't have I/O virtualization IOMMU/VTd enabled, which is needed by the VM.

Signature:

```
VM_REQUIRES_IOMMU(host)
```

VMREQUIRESNETWORK

You attempted to run a VM on a host which doesn't have a PIF on a Network needed by the VM. The VM has at least one VIF attached to the Network.

Signature:

```
VM_REQUIRES_NETWORK(vm, network)
```

VMREQUIRESSR

You attempted to run a VM on a host which doesn't have access to an SR needed by the VM. The VM has at least one VBD attached to a VDI in the SR.

Signature:

```
VM_REQUIRES_SR(vm, sr)
```

VMREQUIRESVDI

VM cannot be started because it requires a VDI which cannot be attached

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_REQUIRES_VDI(vm, vdi)
```

VMREQUIRESVGPU

You attempted to run a VM on a host on which the vGPU required by the VM cannot be allocated on any pGPUs in the GPUgroup needed by the VM.

Signature:

```
VM_REQUIRES_VGPU(vm, GPU_group, vGPU_type)
```

VMREQUIRESVUSB

You attempted to run a VM on a host on which the VUSB required by the VM cannot be allocated on any PUSBs in the USBgroup needed by the VM.

Signature:

```
VM_REQUIRES_VUSB(vm, USB_group)
```

VMREVERTFAILED

An error occured while reverting the specified virtual machine to the specified snapshot

Signature:

```
VM_REVERT_FAILED(vm, snapshot)
```

VMSHUTDOWNTIMEOUT

VM failed to shutdown before the timeout expired

Signature:

```
VM_SHUTDOWN_TIMEOUT(vm, timeout)
```

VMSNAPSHOTWITHQUIESCEFAILED

The quiescedsnapshot operation failed for an unexpected reason

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_SNAPSHOT_WITH_QUIESCE_FAILED(vm)
```

VMSNAPSHOTWITHQUIESCENOTSUPPORTED

The VSS plugin is not installed on this virtual machine

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_NOT_SUPPORTED(vm, error)
```

VMSNAPSHOTWITHQUIESCEPLUGINDEOSNOTRESPOND

The VSS plugin cannot be contacted

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_PLUGIN_DEOS_NOT_RESPOND(vm)
```

VMSNAPSHOTWITHQUIESCETIMEOUT

The VSS plugin has timed out

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_TIMEOUT(vm)
```

VMSUSPENDTIMEOUT

VM failed to suspend before the timeout expired

Signature:

```
VM_SUSPEND_TIMEOUT(vm, timeout)
```

VMTOOMANYVCPUS

Too many VCPUs to start this VM

Signature:

```
VM_TOO_MANY_VCPUS(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMTOIMPORTISNOTNEWERVERSION

The VM cannot be imported unforced because it is either the same version or an older version of an existing VM.

Signature:

```
VM_TO_IMPORT_IS_NOT_NEWER_VERSION(vm, existing_version, version_to_import)
```

VMUNSAFEBOOT

You attempted an operation on a VM that was judged to be unsafe by the server. This can happen if the VM would run on a CPU that has a potentially incompatible set of feature flags to those the VM requires. If you want to override this warning then use the 'force' option.

Signature:

```
VM_UNSAFE_BOOT(vm)
```

VTPMMAXAMOUNTREACHED

The VM cannot be associated with more VTPMs.

Signature:

```
VTPM_MAX_AMOUNT_REACHED(amount)
```

WLBAUTHENTICATIONFAILED

Failed to authenticate with the WLB server, the provided credentials are invalid.

Signature:

```
WLB_AUTHENTICATION_FAILED()
```

WLBCONNECTIONREFUSED

WLB refused a connection to the server.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
WLB_CONNECTION_REFUSED()
```

WLBCONNECTIONRESET

The connection to the WLB server was reset.

Signature:

```
WLB_CONNECTION_RESET()
```

WLBDISABLED

This pool has wlbenabled set to false.

Signature:

```
WLB_DISABLED()
```

WLBINTERNALERROR

WLB reported an internal error.

Signature:

```
WLB_INTERNAL_ERROR()
```

WLBMALFORMEDREQUEST

WLB rejected the server's request as malformed.

Signature:

```
WLB_MALFORMED_REQUEST()
```

WLBMALFORMEDRESPONSE

WLB said something that the server wasn't expecting or didn't understand. The method called on WLB, a diagnostic reason, and the response from WLB are returned.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
WLB_MALFORMED_RESPONSE(method, reason, response)
```

WLBNOTINITIALIZED

No WLB connection is configured.

Signature:

```
WLB_NOT_INITIALIZED()
```

WLBTIMEOUT

The communication with the WLB server timed out.

Signature:

```
WLB_TIMEOUT(configured_timeout)
```

WLBUNKNOWNHOST

The configured WLB server name failed to resolve in DNS.

Signature:

```
WLB_UNKNOWN_HOST()
```

WLBURLINVALID

The WLB URL is invalid. Ensure it is in the format: <ipaddress>:<port>. The configured/given URL is returned.

Signature:

```
WLB_URL_INVALID(url)
```

WLBXENSERVERAUTHENTICATIONFAILED

WLB reported that the server rejected its configured authentication details.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
WLB_XENSERVER_AUTHENTICATION_FAILED()
```

WLBXENSERVERCONNECTIONREFUSED

WLB reported that the server refused to let it connect even though we're connecting perfectly fine in the other direction.

Signature:

```
WLB_XENSERVER_CONNECTION_REFUSED()
```

WLBXENSERVERMALFORMEDRESPONSE

WLB reported that the server said something to it that WLB wasn't expecting or didn't understand.

Signature:

```
WLB_XENSERVER_MALFORMED_RESPONSE()
```

WLBXENSERVERTIMEOUT

WLB reported that communication with the server timed out.

Signature:

```
WLB_XENSERVER_TIMEOUT()
```

WLBXENSERVERUNKNOWNHOST

WLB reported that its configured server name for this server instance failed to resolve in DNS.

Signature:

```
WLB_XENSERVER_UNKNOWN_HOST()
```

XAPIHOOKFAILED

3rd party xapi hook failed

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
XAPI_HOOK_FAILED(hook_name, reason, stdout, exit_code)
```

XENAPIMISSINGPLUGIN

The requested plugin could not be found.

Signature:

```
XENAPI_MISSING_PLUGIN(name)
```

XENAPIPLUGINFAILURE

There was a failure communicating with the plugin.

Signature:

```
XENAPI_PLUGIN_FAILURE(status, stdout, stderr)
```

XENINCOMPATIBLE

The current version of Xen or its control libraries is incompatible with the Toolstack.

Signature:

```
XEN_INCOMPATIBLE()
```

XENVSSREQERRORADDINGVOLUMETOSNAPSETFAILED

Some volumes to be snapshot could not be added to the VSS snapshot set

Signature:

```
XEN_VSS_REQ_ERROR_ADDING_VOLUME_TO_SNAPSET_FAILED(vm, error_code)
```

XENVSSREQERRORCREATINGSNAPSHOT

An attempt to create the snapshots failed

Signature:

```
XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT(vm, error_code)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XENVSSREQERRORCREATINGSNAPSHOTXMLSTRING

Could not create the XML string generated by the transportable snapshot

Signature:

```
XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT_XML_STRING(vm, error_code)
```

XENVSSREQERRORINITFAILED

Initialization of the VSS requester failed

Signature:

```
XEN_VSS_REQ_ERROR_INIT_FAILED(vm, error_code)
```

XENVSSREQERRORNOVOLUMESSUPPORTED

Could not find any volumes supported by the VSS Provider

Signature:

```
XEN_VSS_REQ_ERROR_NO_VOLUMES_SUPPORTED(vm, error_code)
```

XENVSSREQERRORPREPARINGWRITERS

An attempt to prepare VSS writers for the snapshot failed

Signature:

```
XEN_VSS_REQ_ERROR_PREPARING_WRITERS(vm, error_code)
```

XENVSSREQERRORPROVNOTLOADED

The VSS Provider is not loaded

Signature:

```
XEN_VSS_REQ_ERROR_PROV_NOT_LOADED(vm, error_code)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XENVSSREQERRORSTARTSNAPSHOTSETFAILED

An attempt to start a new VSS snapshot failed

Signature:

```
XEN_VSS_REQ_ERROR_START_SNAPSHOT_SET_FAILED(vm, error_code)
```

XMLRPCUNMARSHALFAILURE

The server failed to unmarshal the XMLRPC message; it was expecting one element and received something else.

Signature:

```
XMLRPC_UNMARSHAL_FAILURE(expected, received)
```

XenServer 8.4 Management API

Version **26.1.16**

QUICK LINKS:

- Classes
- Relationships Between Classes
- Types
- ErrorHandling

This document defines the XenServer® Management API - an interface for remotely configuring and controlling virtualized guests running on a Xen-enabled host.

The API is presented here as a set of Remote Procedure Calls (RPCs). There are two supported wire formats, one based upon XML-RPC and one based upon JSON-RPC (v1.0 and v2.0 are both recognized). No specific language bindings are prescribed, although examples are given in the Python programming language.

Although we adopt some terminology from object-oriented programming, future client language bindings may or may not be object-oriented. The API reference uses the terminology *classes* and

© 1997–2026 Citrix Systems, Inc. All rights reserved.

objects. For our purposes a *class* is simply a hierarchical namespace; an *object* is an instance of a class with its fields set to specific values. Objects are persistent and exist on the server-side. Clients may obtain opaque references to these server-side objects and then access their fields via get/set RPCs.

For each class we specify a list of fields along with their *types* and *qualifiers*. A qualifier is one of:

- *RO/runtime*: the field is Read Only. Furthermore, its value is automatically computed at runtime. For example, current CPU load and disk IO throughput.
- *RO/constructor*: the field must be manually set when a new object is created, but is then Read Only for the duration of the object's life. For example, the maximum memory addressable by a guest is set before the guest boots.
- *RW*: the field is Read/Write. For example, the name of a VM.

Types

The following types are used to specify methods and fields in the API Reference:

- `string`: Text strings.
- `int`: 64-bit integers.
- `float`: IEEE double-precision floating-point numbers.
- `bool`: Boolean.
- `datetime`: Date and timestamp.
- `c ref`: Reference to an object of class `c`.
- `t set`: Arbitrary-length set of values of type `t`.
- `(k -> v) map`: Mapping from values of type `k` to values of type `v`.
- `e enum`: Enumeration type with name `e`. Enums are defined in the API reference together with classes that use them.

Note that there are a number of cases where `ref`s are *doubly linked*. For example, a `VM` has a field called `VIFs` of type `VIF ref set`; this field lists the network interfaces attached to a particular VM. Similarly, the `VIF` class has a field called `VM` of type `VM ref` which references the VM to which the interface is connected. These two fields are *bound together*, in the sense that creating a new VIF causes the `VIFs` field of the corresponding VM object to be updated automatically.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The API reference lists explicitly the fields that are bound together in this way. It also contains a diagram that shows relationships between classes. In this diagram an edge signifies the existence of a pair of fields that are bound together, using standard crows-foot notation to signify the type of relationship (for example, one-many, many-many).

RPCs associated with fields

Each field, `f`, has an RPC accessor associated with it that returns `f`'s value:

- `get_f (r)`: takes a `ref`, `r` that refers to an object and returns the value of `f`.

Each field, `f`, with qualifier *RW* and whose outermost type is `set` has the following additional RPCs associated with it:

- `add_f(r, v)`: adds a new element `v` to the set. Note that sets cannot contain duplicate values, hence this operation has no action in the case that `v` is already in the set.
- `remove_f(r, v)`: removes element `v` from the set.

Each field, `f`, with qualifier *RW* and whose outermost type is `map` has the following additional RPCs associated with it:

- `add_to_f(r, k, v)`: adds new pair `k -> v` to the mapping stored in `f` in object `r`. Attempting to add a new pair for duplicate key, `k`, fails with a `MAP_DUPLICATE_KEY` error.
- `remove_from_f(r, k)`: removes the pair with key `k` from the mapping stored in `f` in object `r`.

Each field whose outermost type is neither `set` nor `map`, but whose qualifier is *RW* has an RPC accessor associated with it that sets its value:

- `set_f(r, v)`: sets the field `f` on object `r` to value `v`.

RPCs associated with classes

- Most classes have a *constructor* RPC named `create` that takes as parameters all fields marked *RW* and *RO/constructor*. The result of this RPC is that a new *persistent* object is created on the server-side with the specified field values.
- Each class has a `get_by_uuid(uuid)` RPC that returns the object of that class that has the specified `uuid`.
- Each class that has a `name_label` field has a `get_by_name_label(name_label)` RPC that returns a set of objects of that class that have the specified `name_label`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Most classes have a `destroy(r)` RPC that explicitly deletes the persistent object specified by `r` from the system. This is a non-cascading delete - if the object being removed is referenced by another object then the `destroy` call will fail.

Apart from the RPCs enumerated above, most classes have additional RPCs associated with them. For example, the `VM` class has RPCs for cloning, suspending, starting etc. Such additional RPCs are described explicitly in the API reference.

Wire Protocol for Remote API Calls

API calls are sent over a network to a Xen-enabled host using an RPC protocol. Here we describe how the higher-level types used in our API Reference are mapped to primitive RPC types, covering the two supported wire formats XML-RPC and JSON-RPC.

XML-RPC Protocol

We specify the signatures of API functions in the following style:

```
(VM ref set)  VM.get_all()
```

This specifies that the function with name `VM.get_all` takes no parameters and returns a `set` of `VM ref`. These types are mapped onto XML-RPC types in a straight-forward manner:

- the types `float`, `bool`, `datetime`, and `string` map directly to the XML-RPC `<double>`, `<boolean>`, `<dateTime.iso8601>`, and `<string>` elements.
- all `ref` types are opaque references, encoded as the XML-RPC's `<string>` type. Users of the API should not make assumptions about the concrete form of these strings and should not expect them to remain valid after the client's session with the server has terminated.
- fields named `uuid` of type `string` are mapped to the XML-RPC `<string>` type. The string itself is the OSF DCE UUID presentation format (as output by `uuidgen`).
- `int` is assumed to be 64-bit in our API and is encoded as a string of decimal digits (rather than using XML-RPC's built-in 32-bit `<i4>` type).
- values of `enum` types are encoded as strings. For example, the value `destroy` of `enum on_normal_exit`, would be conveyed as:

```
<value><string>destroy</string></value>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- for all our types, `t`, our type `t set` simply maps to XML-RPC's `<array>` type, so, for example, a value of type `string set` would be transmitted like this:

```
<array>

  <data>

    <value><string>CX8</string></value>

    <value><string>PSE36</string></value>

    <value><string>FPU</string></value>

  </data>

</array>
```

- for types `k` and `v`, our type `(k -> v) map` maps onto an XML-RPC `<struct>`, with the key as the name of the struct. Note that the `(k -> v) map` type is only valid when `k` is a `string`, `ref`, or `int`, and in each case the keys of the maps are stringified as above. For example, the `(string -> float) map` containing the mappings *Mike > 2.3* and *John > 1.2* would be represented as:

```
<value>

  <struct>

    <member>

      <name>Mike</name>

      <value><double>2.3</double></value>

    </member>

    <member>

      <name>John</name>

      <value><double>1.2</double></value>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    </member>
  </struct>
</value>
```

- our `void` type is transmitted as an empty string.

XML-RPC Return Values and Status Codes

The return value of an RPC call is an XML-RPC `<struct>`.

- The first element of the struct is named `Status`; it contains a string value indicating whether the result of the call was a `Success` or a `Failure`.

If the `Status` is `Success` then the struct contains a second element named `Value`:

- The element of the struct named `Value` contains the function's return value.

If the `Status` is `Failure` then the struct contains a second element named `ErrorDescription`:

- The element of the struct named `ErrorDescription` contains an array of string values. The first element of the array is an error code; the rest of the elements are strings representing error parameters relating to that code.

For example, an XML-RPC return value from the `host.get_resident_VMs` function may look like this:

```
<struct>
   <member>
     <name>Status</name>
     <value>Success</value>
   </member>
   <member>
      <name>Value</name>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
      <value>
        <array>
           <data>
             <value>81547a35-205c-a551-c577-00b982c5fe00</value>
             <value>61c85a22-05da-b8a2-2e55-06b0847da503</value>
             <value>1d401ec4-3c17-35a6-fc79-cee6bd9811fe</value>
           </data>
        </array>
     </value>
   </member>
</struct>
```

JSON-RPC Protocol

We specify the signatures of API functions in the following style:

```
(VM ref set)  VM.get_all()
```

This specifies that the function with name `VM.get_all` takes no parameters and returns a `set` of `VM ref`. These types are mapped onto JSON-RPC types in the following manner:

- the types `float` and `bool` map directly to the JSON types `number` and `boolean`, while `datetime` and `string` are represented as the JSON `string` type.
- all `ref` types are opaque references, encoded as the JSON `string` type. Users of the API should not make assumptions about the concrete form of these strings and should not expect them to remain valid after the client's session with the server has terminated.
- fields named `uuid` of type `string` are mapped to the JSON `string` type. The string itself is the OSF DCE UUID presentation format (as output by `uuidgen`).

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `int` is assumed to be 64-bit in our API and is encoded as a JSON `number` without decimal point or exponent, preserved as a string.
- values of `enum` types are encoded as the JSON `string` type. For example, the value `destroy` of `enum on_normal_exit`, would be conveyed as:

```
"destroy"
```

- for all our types, `t`, our type `t set` simply maps to the JSON `array` type, so, for example, a value of type `string set` would be transmitted like this:

```
[ "CX8", "PSE36", "FPU" ]
```

- for types `k` and `v`, our type `(k -> v) map` maps onto a JSON object which contains members with name `k` and value `v`. Note that the `(k -> v) map` type is only valid when `k` is a `string`, `ref`, or `int`, and in each case the keys of the maps are stringified as above. For example, the `(string -> float) map` containing the mappings *Mike > 2.3* and *John > 1.2* would be represented as:

```
{

  "Mike": 2.3,

  "John": 1.2

}
```

- our `void` type is transmitted as an empty string.

Both versions 1.0 and 2.0 of the JSON-RPC wire format are recognised and, depending on your client library, you can use either of them.

JSON-RPC v1.0

JSON-RPC V1.0 REQUESTS

An API call is represented by sending a single JSON object to the server, which contains the members `method`, `params`, and `id`.

- `method`: A JSON `string` containing the name of the function to be invoked.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- `params`: A JSON `array` of values, which represents the parameters of the function to be invoked.
- `id`: A JSON `string` or `integer` representing the call id. Note that, diverging from the JSON-RPC v1.0 specification the API does not accept *notification* requests (requests without responses), i.e. the id cannot be `null`.

For example, the body of a JSON-RPC v1.0 request to retrieve the resident VMs of a host may look like this:

```
{

  "method": "host.get_resident_VMs",

  "params": [

    "OpaqueRef:74f1a19cd-b660-41e3-a163-10f03e0eae67",

    "OpaqueRef:08c34fc9-f418-4f09-8274-b9cb25cd8550"

  ],

  "id": "xyz"

}
```

In the above example, the first element of the `params` array is the reference of the open session to the host, while the second is the host reference.

JSON-RPC v1.0 return values

The return value of a JSON-RPC v1.0 call is a single JSON object containing the members `result`, `error`, and `id`.

- `result`: If the call is successful, it is a JSON value (`string`, `array` etc.) representing the return value of the invoked function. If an error has occurred, it is `null`.
- `error`: If the call is successful, it is `null`. If the call has failed, it a JSON `array` of `string` values. The first element of the array is an error code; the remainder of the array are strings representing error parameters relating to that code.
- `id`: The call id. It is a JSON `string` or `integer` and it is the same id as the request it is responding to.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

For example, a JSON-RPC v1.0 return value from the `host.get_resident_VMs` function may look like this:

```
{
  "result": [
      "OpaqueRef:604f51e7-630f-4412-83fa-b11c6cf008ab",
      "OpaqueRef:670d08f5-cbeb-4336-8420-ccd56390a65f"
  ],
  "error": null,
  "id": "xyz"
}
```

while the return value of the same call made on a logged out session may look like this:

```
{
  "result": null,
  "error": [
      "SESSION_INVALID",
      "OpaqueRef:93f1a23cd-a640-41e3-b163-10f86e0eae67"
  ],
  "id": "xyz"
}
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

JSON-RPC v2.0

JSON-RPC V2.0 REQUESTS

An API call is represented by sending a single JSON object to the server, which contains the members `jsonrpc`, `method`, `params`, and `id`.

- `jsonrpc`: A JSON `string` specifying the version of the JSON-RPC protocol. It is exactly "2.0".
- `method`: A JSON `string` containing the name of the function to be invoked.
- `params`: A JSON `array` of values, which represents the parameters of the function to be invoked. Although the JSON-RPC v2.0 specification allows this member to be omitted, in practice all API calls accept at least one parameter.
- `id`: A JSON `string` or `integer` representing the call id. Note that, diverging from the JSON-RPC v2.0 specification it cannot be null. Neither can it be omitted because the API does not accept *notification* requests (requests without responses).

For example, the body of a JSON-RPC v2.0 request to retrieve the VMs resident on a host may look like this:

```
{
  "jsonrpc": "2.0",
  "method": "host.get_resident_VMs",
  "params": [
    "OpaqueRef:c90cd28f-37ec-4dbf-88e6-f697ccb28b39",
    "OpaqueRef:08c34fc9-f418-4f09-8274-b9cb25cd8550"
  ],
  "id": 3
}
```

As before, the first element of the `parameter` array is the reference of the open session to the host, while the second is the host reference.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

JSON-RPC v2.0 return values

The return value of a JSON-RPC v2.0 call is a single JSON object containing the members `jsonrpc`, either `result` or `error` depending on the outcome of the call, and `id`.

- `jsonrpc`: A JSON `string` specifying the version of the JSON-RPC protocol. It is exactly "2.0".
- `result`: If the call is successful, it is a JSON value (`string`, `array` etc.) representing the return value of the invoked function. If an error has occurred, it does not exist.
- `error`: If the call is successful, it does not exist. If the call has failed, it is a single structured JSON object (see below).
- `id`: The call id. It is a JSON `string` or `integer` and it is the same id as the request it is responding to.

The `error` object contains the members `code`, `message`, and `data`.

- `code`: The API does not make use of this member and only retains it for compliance with the JSON-RPC v2.0 specification. It is a JSON `integer` which has a non-zero value.
- `message`: A JSON `string` representing an API error code.
- `data`: A JSON array of `string` values representing error parameters relating to the aforementioned API error code.

For example, a JSON-RPC v2.0 return value from the `host.get_resident_VMs` function may look like this:

```
{
  "jsonrpc": "2.0",
  "result": [
      "OpaqueRef:604f51e7-630f-4412-83fa-b11c6cf008ab",
      "OpaqueRef:670d08f5-cbeb-4336-8420-ccd56390a65f"
  ],
  "id": 3
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
}
```

while the return value of the same call made on a logged out session may look like this:

```
{
  "jsonrpc": "2.0",
  "error": {
      "code": 1,
      "message": "SESSION_INVALID",
      "data": [
          "OpaqueRef:c90cd28f-37ec-4dbf-88e6-f697ccb28b39"
      ]
  },
  "id": 3
}
```

Errors

When a low-level transport error occurs, or a request is malformed at the HTTP or RPC level, the server may send an HTTP 500 error response, or the client may simulate the same. The client must be prepared to handle these errors, though they may be treated as fatal.

For example, the following malformed request when using the XML-RPC protocol:

```
$curl -D - -X POST https://server -H 'Content-Type: application/xml' \
  -d '<?xml version="1.0"?>
  <methodCall>
    <methodName>session.logout</methodName>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  </methodCall>'
```

results to the following response:

```
HTTP/1.1 500 Internal Error

content-length: 297

content-type:text/html

connection:close

cache-control:no-cache, no-store



<html><body><h1>HTTP 500 internal server error</h1>An unexpected error
occurred;

 please wait a while and try again. If the problem persists, please
contact your

 support representative.<h1> Additional information </
h1>Xmlrpc.Parse_error(&quo

t;close_tag&quot;, &quot;open_tag&quot;, _)</body></html>
```

When using the JSON-RPC protocol:

```
$curl -D - -X POST https://server/jsonrpc -H 'Content-Type: application/
json' \

  -d '{

      "jsonrpc": "2.0",

      "method": "session.login_with_password",

      "id": 0
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  }'
```

the response is:

```
HTTP/1.1 500 Internal Error

content-length: 308

content-type:text/html

connection:close

cache-control:no-cache, no-store



<html><body><h1>HTTP 500 internal server error</h1>An unexpected error
occurred;

 please wait a while and try again. If the problem persists, please
contact your

 support representative.<h1> Additional information </
h1>Jsonrpc.Malformed_metho

d_request(&amp;quot;{jsonrpc=...,method=...,id=...}&amp;quot;)</body></html>
```

All other failures are reported with a more structured error response, to allow better automatic response to failures, proper internationalization of any error message, and easier debugging.

On the wire, these are transmitted like this when using the XML-RPC protocol:

```
<struct>

    <member>

        <name>Status</name>

        <value>Failure</value>

    </member>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    <member>
        <name>ErrorDescription</name>
        <value>
            <array>
                <data>
                    <value>MAP_DUPLICATE_KEY</value>
                    <value>Customer</value>
                    <value>eSpiel Inc.</value>
                    <value>eSpiel Incorporated</value>
                </data>
            </array>
        </value>
    </member>
</struct>
```

Note that `ErrorDescription` value is an array of string values. The first element of the array is an error code; the remainder of the array are strings representing error parameters relating to that code. In this case, the client has attempted to add the mapping *Customer* > *eSpiel Incorporated* to a Map, but it already contains the mapping *Customer* > *eSpiel Inc.*, hence the request has failed.

When using the JSON-RPC protocol v2.0, the above error is transmitted as:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": 1,
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        "message": "MAP_DUPLICATE_KEY",
        "data": [
            "Customer",
            "eSpiel Inc.",
            "eSpiel Incorporated"
        ]
    },
    "id": 3
}
```

Finally, when using the JSON-RPC protocol v1.0:

```
{
    "result": null,
    "error": [
        "MAP_DUPLICATE_KEY",
        "Customer",
        "eSpiel Inc.",
        "eSpiel Incorporated"
    ],
    "id": "xyz"
}
```

Each possible error code is documented in the last section of the API reference.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Note on References vs UUIDs

References are opaque types - encoded as XML-RPC and JSON-RPC strings on the wire - understood only by the particular server which generated them. Servers are free to choose any concrete representation they find convenient; clients should not make any assumptions or attempt to parse the string contents. References are not guaranteed to be permanent identifiers for objects; clients should not assume that references generated during one session are valid for any future session. References do not allow objects to be compared for equality. Two references to the same object are not guaranteed to be textually identical.

UUIDs are intended to be permanent identifiers for objects. They are guaranteed to be in the OSF DCE UUID presentation format (as output by `uuidgen`). Clients may store UUIDs on disk and use them to look up objects in subsequent sessions with the server. Clients may also test equality on objects by comparing UUID strings.

The API provides mechanisms for translating between UUIDs and opaque references. Each class that contains a UUID field provides:

- A `get_by_uuid` method that takes a UUID and returns an opaque reference to the server-side object that has that UUID;
- A `get_uuid` function (a regular "field getter" RPC) that takes an opaque reference and returns the UUID of the server-side object that is referenced by it.

Making RPC Calls

Transport Layer

The following transport layers are currently supported:

- HTTP/HTTPS for remote administration
- HTTP over Unix domain sockets for local administration

Session Layer

The RPC interface is session-based; before you can make arbitrary RPC calls you must login and initiate a session. For example:

```
(session ref) session.login_with_password(string uname, string pwd,

                string version, string originator)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

where `uname` and `password` refer to your username and password, as defined by the Xen administrator, while `version` and `originator` are optional. The `session ref` returned by `session.login_with_password` is passed to subsequent RPC calls as an authentication token. Note that a session reference obtained by a login request to the XML-RPC backend can be used in subsequent requests to the JSON-RPC backend, and vice versa.

A session can be terminated with the `session.logout` function:

```
void  session.logout(session ref session_id)
```

Synchronous and Asynchronous Invocation

Each method call (apart from methods on the `Session` and `Task` objects and "getters" and "setters" derived from fields) can be made either synchronously or asynchronously. A synchronous RPC call blocks until the return value is received; the return value of a synchronous RPC call is exactly as specified above.

Only synchronous API calls are listed explicitly in this document. All their asynchronous counterparts are in the special `Async` namespace. For example, the synchronous call `VM.clone(...)` has an asynchronous counterpart, `Async.VM.clone(...)`, that is non-blocking.

Instead of returning its result directly, an asynchronous RPC call returns an identifier of type `task ref` which is subsequently used to track the status of a running asynchronous RPC.

Note that an asynchronous call may fail immediately, before a task has even been created. When using the XML-RPC wire protocol, this eventuality is represented by wrapping the returned `task ref` in an XML-RPC struct with a `Status`, `ErrorDescription`, and `Value` fields, exactly as specified above; the `task ref` is provided in the `Value` field if `Status` is set to `Success`. When using the JSON-RPC protocol, the `task ref` is wrapped in a response JSON object as specified above and it is provided by the value of the `result` member of a successful call.

The RPC call

```
(task ref set)  Task.get_all(session ref session_id)
```

returns a set of all task identifiers known to the system. The status (including any returned result and error codes) of these can then be queried by accessing the fields of the `Task` object in the usual way. Note that, in order to get a consistent snapshot of a task's state, it is advisable to call the `get_record` function.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Example interactive session

This section describes how an interactive session might look, using python XML-RPC and JSON-RPC client libraries.

First, initialise python:

```
$ python3

>>>
```

Using the XML-RPC Protocol

Import the library `xmlrpc.client` and create a python object referencing the remote server as shown below:

```
>>> import xmlrpc.client

>>> xen = xmlrpc.client.ServerProxy("https://localhost:443")
```

Note that you may need to disable SSL certificate validation to establish the connection, this can be done as follows:

```
>>> import ssl

>>> ctx = ssl._create_unverified_context()

>>> xen = xmlrpc.client.ServerProxy("https://localhost:443", context=ctx)
```

Acquire a session reference by logging in with a username and password; the session reference is returned under the key `Value` in the resulting dictionary (error-handling omitted for brevity):

```
>>> session = xen.session.login_with_password("user", "passwd",

...                                            "version", "originator")
['Value']
```

This is what the call looks like when serialized

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
<?xml version='1.0'?>

<methodCall>

    <methodName>session.login_with_password</methodName>

    <params>

        <param><value><string>user</string></value></param>

        <param><value><string>passwd</string></value></param>

        <param><value><string>version</string></value></param>

        <param><value><string>originator</string></value></param>

    </params>

</methodCall>
```

Next, the user may acquire a list of all the VMs known to the system (note the call takes the session reference as the only parameter):

```
>>> all_vms = xen.VM.get_all(session)['Value']

>>> all_vms

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]
```

The VM references here have the form `OpaqueRef:X` (though they may not be that simple in reality) and you should treat them as opaque strings. *Templates* are VMs with the `is_a_template` field set to `true`. We can find the subset of template VMs using a command like the following:

```
>>> all_templates = filter(lambda x: xen.VM.get_is_a_template(session, x)
['Value'],

                            all_vms)
```

Once a reference to a VM has been acquired, a lifecycle operation may be invoked:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
>>> xen.VM.start(session, all_templates[0], False, False)

{'Status': 'Failure', 'ErrorDescription': ['VM_IS_TEMPLATE',
'OpaqueRef:X']}
```

In this case the `start` message has been rejected, because the VM is a template, and so an error response has been returned. These high-level errors are returned as structured data (rather than as XML-RPC faults), allowing them to be internationalized.

Rather than querying fields individually, whole *records* may be returned at once. To retrieve the record of a single object as a python dictionary:

```
>>> record = xen.VM.get_record(session, all_templates[0])['Value']

>>> record['power_state']

'Halted'

>>> record['name_label']

'Windows 10 (64-bit)'
```

To retrieve all the VM records in a single call:

```
>>> records = xen.VM.get_all_records(session)['Value']

>>> list(records.keys())

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]

>>> records['OpaqueRef:1']['name_label']

'Red Hat Enterprise Linux 7'
```

Using the JSON-RPC Protocol

For this example we are making use of the package `jsonrpcclient` and the `requests` library due to their simplicity, although other packages can also be used.

First, import the `requests` and `jsonrpcclient` libraries:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
>>> import requests

>>> import jsonrpcclient
```

Now we construct a utility method to make using these libraries easier:

```
>>> def jsonrpccall(method, params):

...     r = requests.post("https://localhost:443/jsonrpc",

...                       json=jsonrpcclient.request(method,
params=params),

...                       verify=False)

...     p = jsonrpcclient.parse(r.json())

...     if isinstance(p, jsonrpcclient.Ok):

...         return p.result

...     raise Exception(p.message, p.data)
```

Acquire a session reference by logging in with a username and password:

```
>>> session = jsonrpccall("session.login_with_password",

...                       ("user", "password", "version", "originator"))
```

`jsonrpcclient` uses the JSON-RPC protocol v2.0, so this is what the serialized request looks like:

```
{

  "jsonrpc": "2.0",

  "method": "session.login_with_password",

  "params": ["user", "passwd", "version", "originator"],
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
  "id": 0

}
```

Next, the user may acquire a list of all the VMs known to the system (note the call takes the session reference as the only parameter):

```
>>> all_vms = jsonrpccall("VM.get_all", (session,))

>>> all_vms

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]
```

The VM references here have the form `OpaqueRef:X` (though they may not be that simple in reality) and you should treat them as opaque strings. *Templates* are VMs with the `is_a_template` field set to `true`. We can find the subset of template VMs using a command like the following:

```
>>> all_templates = filter(

...     lambda x: jsonrpccall("VM.get_is_a_template", (session, x)),

...     all_vms)
```

Once a reference to a VM has been acquired, a lifecycle operation may be invoked:

```
>>> try:

...     jsonrpccall("VM.start", (session, next(all_templates), False,
False))

... except Exception as e:

...     e

...

Exception('VM_IS_TEMPLATE', ['OpaqueRef:1', 'start'])
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

In this case the `start` message has been rejected because the VM is a template, hence an error response has been returned. These high-level errors are returned as structured data, allowing them to be internationalized.

Rather than querying fields individually, whole *records* may be returned at once. To retrieve the record of a single object as a python dictionary:

```
>>> record = jsonrpccall("VM.get_record", (session, next(all_templates)))

>>> record['power_state']

'Halted'

>>> record['name_label']

'Windows 10 (64-bit)'
```

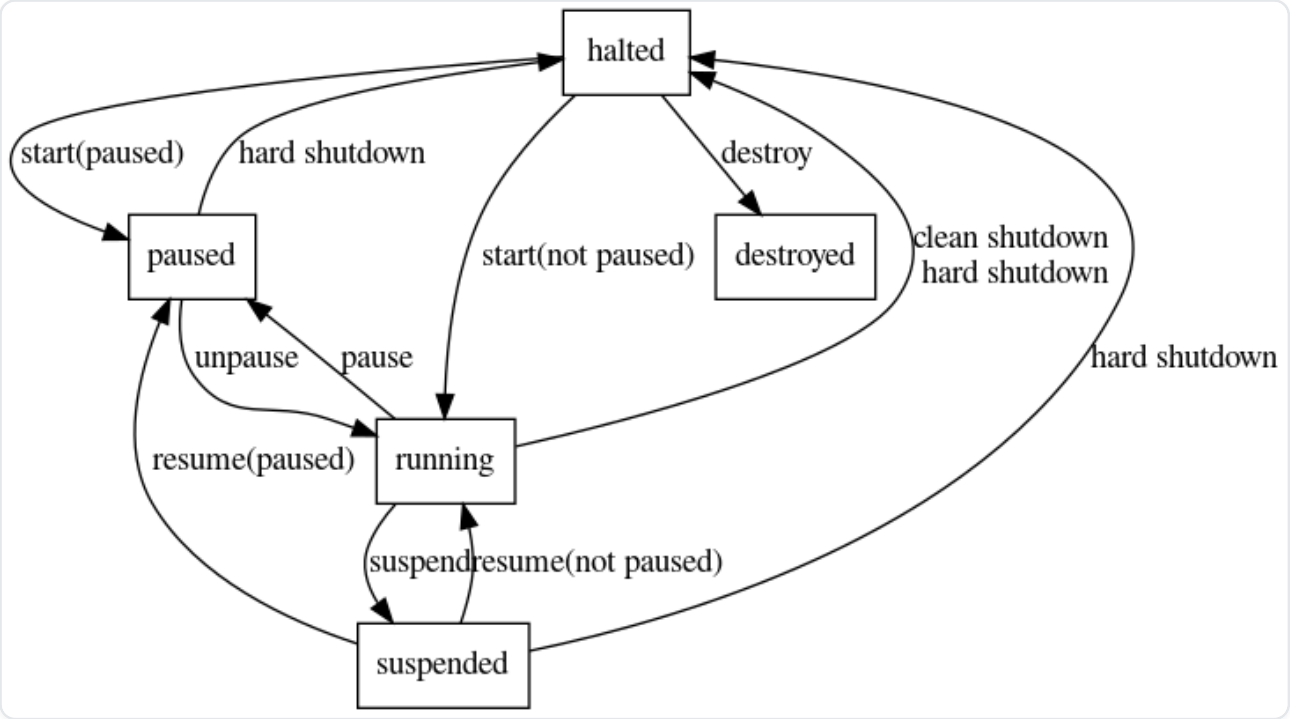
To retrieve all the VM records in a single call:

```
>>> records = jsonrpccall("VM.get_all_records", (session,))

>>> records.keys()

['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]

>>> records['OpaqueRef:1']['name_label']

'Red Hat Enterprise Linux 7'
```

VM Lifecycle

The following figure shows the states that a VM can be in and the API calls that can be used to move the VM between these states.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



VM boot parameters

The `VM` class contains a number of fields that control the way in which the VM is booted. With reference to the fields defined in the VM class (see later in this document), this section outlines the boot options available and the mechanisms provided for controlling them.

VM booting is controlled by setting one of the two mutually exclusive groups: "PV" and "HVM". If `HVM.boot_policy` is an empty string, then paravirtual domain building and booting will be used; otherwise the VM will be loaded as a HVM domain, and booted using an emulated BIOS.

When paravirtual booting is in use, the `PV_bootloader` field indicates the bootloader to use. It may be "pygrub", in which case the platform's default installation of pygrub will be used, or a full path within the control domain to some other bootloader. The other fields, `PV_kernel`, `PV_ramdisk`, `PV_args`, and `PV_bootloader_args` will be passed to the bootloader unmodified, and interpretation of those fields is then specific to the bootloader itself, including the possibility that the bootloader will ignore some or all of those given values. Finally the paths of all bootable disks are added to the bootloader commandline (a disk is bootable if its VBD has the bootable flag set). There may be zero, one, or many bootable disks; the bootloader decides which disk (if any) to boot from.

If the bootloader is pygrub, then the menu.lst is parsed, if present in the guest's filesystem, otherwise the specified kernel and ramdisk are used, or an autodetected kernel is used if nothing is specified and autodetection is possible. `PV_args` is appended to the kernel command line, no matter which mechanism is used for finding the kernel.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If `PV_bootloader` is empty but `PV_kernel` is specified, then the kernel and ramdisk values will be treated as paths within the control domain. If both `PV_bootloader` and `PV_kernel` are empty, then the behaviour is as if `PV_bootloader` were specified as "pygrub".

When using HVM booting, `HVM_boot_policy` and `HVM_boot_params` specify the boot handling. Only one policy is currently defined, "BIOS order". In this case, `HVM_boot_params` should contain one key-value pair "order" = "N" where N is the string that will be passed to QEMU. Optionally `HVM_boot_params` can contain another key-value pair "firmware" with values "bios" or "uefi" (default is "bios" if absent). By default Secure Boot is not enabled, it can be enabled when "uefi" is enabled by setting `VM.platform["secureboot"]` to true.

Classes

The following classes are defined:

NAME	DESCRIPTION
auth	Management of remote authentication services
blob	A placeholder for a binary blob
Bond	A Network bond that combines physical network interfaces, also known as link aggregation
Certificate	An X509 certificate used for TLS connections
Cluster	Cluster-wide Cluster metadata
Cluster_host	Cluster member metadata
console	A console
crashdump	**Deprecated**. A VM crashdump
data_source	Data sources for logging in RRDs
DR_task	DR task
Driver_variant	UNSUPPORTED. Variant of a host driver
event	Asynchronous event registration and handling

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
Feature	A new piece of functionality
GPU_group	A group of compatible GPUs across the resource pool
host	A physical host
host_cpu	**Deprecated**. A physical CPU
host_crashdump	Represents a host crash dump
Host_driver	UNSUPPORTED. A multi-version driver on a host
host_metrics	The metrics associated with a host
host_patch	**Deprecated**. Represents a patch stored on a server
LVHD	LVHD SR specific operations
message	An message for the attention of the administrator
network	A virtual network
network_sriov	network-sriov which connects logical pif and physical pif
Observer	Describes an observer which will control observability activity in the Toolstack
PBD	The physical block devices through which hosts access SRs
PCI	A PCI device
PGPU	A physical GPU (pGPU)
PIF	A physical network interface (note separate VLANs are represented as several PIFs)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
PIF_metrics	The metrics associated with a physical network interface
pool	Pool-wide information
pool_patch	**Deprecated**. Pool-wide patches
pool_update	Pool-wide updates to the host software
probe_result	A set of properties that describe one result element of SR.probe. Result elements and properties can change dynamically based on changes to the the SR.probe input-parameters or the target.
PUSB	A physical USB device
PVS_cache_storage	Describes the storage that is available to a PVS site for caching purposes
PVS_proxy	a proxy connects a VM/VIF with a PVS site
PVS_server	individual machine serving provisioning (block) data
PVS_site	machines serving blocks of data for provisioning VMs
Repository	Repository for updates
role	A set of permissions associated with a subject
SDN_controller	Describes the SDN controller that is to connect with the pool
secret	A secret
session	A session
SM	A storage manager plugin
SR	A storage repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
sr_stat	A set of high-level properties associated with an SR.
subject	A user or group that can log in xapi
task	A long-running asynchronous task
tunnel	A tunnel for network traffic
USB_group	A group of compatible USBs across the resource pool
user	**Deprecated**. A user of the system
VBD	A virtual block device
VBD_metrics	**Removed**. The metrics associated with a virtual block device
VDI	A virtual disk image
vdi_nbd_server_info	Details for connecting to a VDI using the Network Block Device protocol
VGPU	A virtual GPU (vGPU)
VGPU_type	A type of virtual GPU
VIF	A virtual network interface
VIF_metrics	**Removed**. The metrics associated with a virtual network device
VLAN	A VLAN mux/demux
VM	A virtual machine (or 'guest').
VM_appliance	VM appliance
VM_group	A VM group

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAME	DESCRIPTION
VM_guest_metrics	The metrics reported by the guest (as opposed to inferred from outside)
VM_metrics	The metrics associated with a VM
VMPP	**Removed**. VM Protection Policy
VMSS	VM Snapshot Schedule
VTPM	A virtual TPM device
VUSB	Describes the vusb device

Relationships Between Classes

Fields that are bound together are shown in the following table:

OBJECT.FIELD	*OBJECT.FIELD*	*RELATIONSHIP*
`VM.snapshot_of`	`VM.snapshots`	one-to-many
`VDI.snapshot_of`	`VDI.snapshots`	one-to-many
`VM.parent`	`VM.children`	one-to-many
`task.subtask_of`	`task.subtasks`	one-to-many
`PIF.bond_slave_of`	`Bond.slaves`	one-to-many
`Bond.master`	`PIF.bond_master_of`	one-to-many
`VLAN.tagged_PIF`	`PIF.VLAN_slave_of`	one-to-many
`tunnel.access_PIF`	`PIF.tunnel_access_PIF_of`	one-to-many
`tunnel.transport_PIF`	`PIF.tunnel_transport_PIF_of`	one-to-many
`PBD.host`	`host.PBDs`	one-to-many

© 1997–2026 Citrix Systems, Inc. All rights reserved.

OBJECT.FIELD	OBJECT.FIELD	RELATIONSHIP
`PBD.SR`	`SR.PBDs`	one-to-many
`VBD.VDI`	`VDI.VBDs`	one-to-many
`crashdump.VDI`	`VDI.crash_dumps`	one-to-many
`VBD.VM`	`VM.VBDs`	one-to-many
`crashdump.VM`	`VM.crash_dumps`	one-to-many
`VIF.VM`	`VM.VIFs`	one-to-many
`VIF.network`	`network.VIFs`	one-to-many
`Cluster_host.cluster`	`Cluster.cluster_hosts`	one-to-many
`PIF.host`	`host.PIFs`	one-to-many
`PIF.network`	`network.PIFs`	one-to-many
`VDI.SR`	`SR.VDIs`	one-to-many
`VTPM.VM`	`VM.VTPMs`	one-to-many
`console.VM`	`VM.consoles`	one-to-many
`VM.resident_on`	`host.resident_VMs`	one-to-many
`host_cpu.host`	`host.host_CPUs`	one-to-many
`host_crashdump.host`	`host.crashdumps`	one-to-many
`host_patch.host`	`host.patches`	one-to-many
`host_patch.pool_patch`	`pool_patch.host_patches`	one-to-many
`host.updates`	`pool_update.hosts`	many-to-many
`subject.roles`	`subject.roles`	unknown type
`role.subroles`	`role.subroles`	many-to-many

© 1997–2026 Citrix Systems, Inc. All rights reserved.

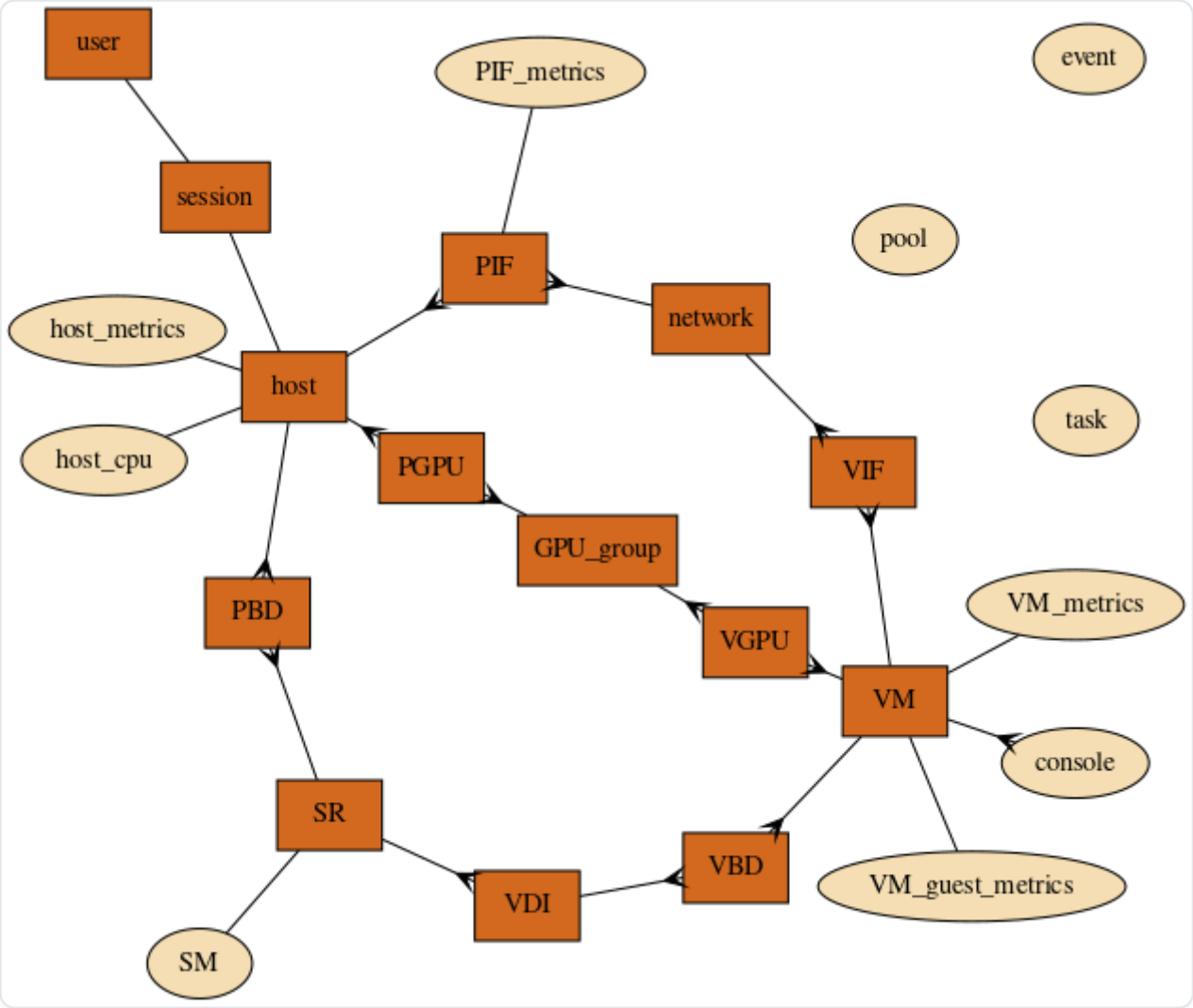
OBJECT.FIELD	OBJECT.FIELD	RELATIONSHIP
`VM.protection_policy`	`VMPP.VMs`	one-to-many
`VM.snapshot_schedule`	`VMSS.VMs`	one-to-many
`VM.appliance`	`VM_appliance.VMs`	one-to-many
`PGPU.GPU_group`	`GPU_group.PGPUs`	one-to-many
`VGPU.GPU_group`	`GPU_group.VGPUs`	one-to-many
`VGPU.type`	`VGPU_type.VGPUs`	one-to-many
`VGPU.VM`	`VM.VGPUs`	one-to-many
`VGPU.resident_on`	`PGPU.resident_VGPUs`	one-to-many
`PGPU.supported_VGPU_types`	`VGPU_type.supported_on_PGPUs`	many-to-many
`PGPU.enabled_VGPU_types`	`VGPU_type.enabled_on_PGPUs`	many-to-many
`GPU_group.supported_VGPU_types`	`VGPU_type.supported_on_GPU_groups`	many-to-many
`GPU_group.enabled_VGPU_types`	`VGPU_type.enabled_on_GPU_groups`	many-to-many
`PCI.host`	`host.PCIs`	one-to-many
`PGPU.host`	`host.PGPUs`	one-to-many
`VDI.metadata_of_pool`	`pool.metadata_VDIs`	one-to-many
`SR.introduced_by`	`DR_task.introduced_SRs`	one-to-many
`PVS_server.site`	`PVS_site.servers`	one-to-many
`PVS_proxy.site`	`PVS_site.proxies`	one-to-many
`PVS_cache_storage.site`	`PVS_site.cache_storage`	one-to-many

© 1997–2026 Citrix Systems, Inc. All rights reserved.

OBJECT.FIELD	*OBJECT.FIELD*	*RELATIONSHIP*
`PUSB.host`	`host.PUSBs`	one-to-many
`PUSB.USB_group`	`USB_group.PUSBs`	one-to-many
`VUSB.USB_group`	`USB_group.VUSBs`	one-to-many
`VUSB.VM`	`VM.VUSBs`	one-to-many
`Feature.host`	`host.features`	one-to-many
`network_sriov.physical_PIF`	`PIF.sriov_physical_PIF_of`	one-to-many
`network_sriov.logical_PIF`	`PIF.sriov_logical_PIF_of`	one-to-many
`Certificate.host`	`host.certificates`	one-to-many
`VM.groups`	`VM_group.VMs`	many-to-many
`Driver_variant.driver`	`Host_driver.variants`	one-to-many

The following figure represents bound fields (as specified above) using crow's foot notation to specify one-to-one, one-to-many, or many-to-many relationships:

© 1997–2026 Citrix Systems, Inc. All rights reserved.



Types

Primitives

The following primitive types are used to specify methods and fields in the API Reference:

TYPE	DESCRIPTION
`string`	text strings
`int`	64-bit integers
`float`	IEEE double-precision floating-point numbers
`bool`	boolean
`datetime`	date and timestamp

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Higher-order types

The following type constructors are used:

TYPE	DESCRIPTION
`c ref`	reference to an object of class `c`
`t set`	a set of elements of type `t`
`(a -> b) map`	a table mapping values of type `a` to values of type `b`

Enumeration types

The following enumeration types are used:

Enum after_apply_guidance

NAMED VALUE	DESCRIPTION
`restartHost`	This patch requires the host to be restarted once applied.
`restartHVM`	This patch requires HVM guests to be restarted once applied.
`restartPV`	This patch requires PV guests to be restarted once applied.
`restartXAPI`	This patch requires XAPI to be restarted once applied.

Enum allocation_algorithm

NAMED VALUE	DESCRIPTION
`breadth_first`	vGPUs of a given type are allocated evenly across supporting pGPUs.
`depth_first`	vGPUs of a given type are allocated on supporting pGPUs until they are full.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum bond_mode

NAMED VALUE	DESCRIPTION
`active-backup`	Active/passive bonding: only one NIC is carrying traffic
`balance-slb`	Source-level balancing
`lacp`	Link aggregation control protocol

Enum certificate_purpose

NAMED VALUE	DESCRIPTION
`licensing`	Trusted certificates that are for licensing purpose.

Enum certificate_type

NAMED VALUE	DESCRIPTION
`ca`	Certificate that is trusted by the whole pool
`host`	Certificate that identifies a single host to entities outside the pool
`host_internal`	Certificate that identifies a single host to other pool members
`pinned`	Pinned leaf certificate that is trusted by the whole pool

Enum cls

NAMED VALUE	DESCRIPTION
`Certificate`	Certificate
`Host`	Host
`Pool`	Pool
`PVS_proxy`	PVS_proxy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`SR`	SR
`VDI`	VDI
`VM`	VM
`VMPP`	VMPP
`VMSS`	VMSS

Enum cluster_host_operation

NAMED VALUE	DESCRIPTION
`destroy`	completely destroying a cluster host
`disable`	disabling cluster membership on a particular host
`enable`	enabling cluster membership on a particular host

Enum cluster_operation

NAMED VALUE	DESCRIPTION
`add`	adding a new member to the cluster
`destroy`	completely destroying a cluster
`disable`	disabling any cluster member
`enable`	enabling any cluster member
`remove`	removing a member from the cluster

Enum console_protocol

NAMED VALUE	DESCRIPTION
`rdp`	Remote Desktop Protocol

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`rfb`	Remote FrameBuffer protocol (as used in VNC)
`vt100`	VT100 terminal

Enum domain_type

NAMED VALUE	DESCRIPTION
`hvm`	HVM; Fully Virtualised
`pv`	PV: Paravirtualised
`pv_in_pvh`	PV inside a PVH container
`pvh`	PVH
`unspecified`	Not specified or unknown domain type

Enum event_operation

NAMED VALUE	DESCRIPTION
`add`	An object has been created
`del`	An object has been deleted
`mod`	An object has been modified

Enum host_allowed_operations

NAMED VALUE	DESCRIPTION
`apply_updates`	Indicates this host is being updated
`enable`	Indicates this host is in the process of enabling
`evacuate`	Indicates this host is evacuating
`power_on`	Indicates this host is in the process of being powered on

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`provision`	Indicates this host is able to provision another VM
`reboot`	Indicates this host is in the process of rebooting
`shutdown`	Indicates this host is in the process of shutting itself down
`vm_migrate`	This host is the migration target of a VM
`vm_resume`	This host is resuming a VM
`vm_start`	This host is starting a VM

Enum host_display

NAMED VALUE	DESCRIPTION
`disable_on_reboot`	The host will stop outputting its console to a physical display device on next boot
`disabled`	This host is not outputting its console to a physical display device
`enable_on_reboot`	The host will start outputting its console to a physical display device on next boot
`enabled`	This host is outputting its console to a physical display device

Enum host_ntp_mode

NAMED VALUE	DESCRIPTION
`Custom`	Using custom NTP servers configured by user to sync time
`DHCP`	Using NTP servers assigned by DHCP to sync time

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`Disabled`	NTP is disabled on the host
`Factory`	Using built-in NTP servers to sync time

Enum host_sched_gran

NAMED VALUE	DESCRIPTION
`core`	core scheduling
`cpu`	CPU scheduling
`socket`	socket scheduling

Enum image_format_type

NAMED VALUE	DESCRIPTION
`qcow2`	Qemu Copy-On-Write version 2
`raw`	Plain disk image
`vhd`	Virtual Hard Disk

Enum ip_configuration_mode

NAMED VALUE	DESCRIPTION
`DHCP`	Acquire an IP address by DHCP
`None`	Do not acquire an IP address
`Static`	Static IP address configuration

Enum ipv6_configuration_mode

NAMED VALUE	DESCRIPTION
`Autoconf`	Router assigned prefix delegation IPv6 allocation

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`DHCP`	Acquire an IPv6 address by DHCP
`None`	Do not acquire an IPv6 address
`Static`	Static IPv6 address configuration

Enum latest_synced_updates_applied_state

NAMED VALUE	DESCRIPTION
`no`	The host is outdated with the latest updates synced from remote CDN
`unknown`	If the host is up to date with the latest updates synced from remote CDN is unknown
`yes`	The host is up to date with the latest updates synced from remote CDN

Enum livepatch_status

NAMED VALUE	DESCRIPTION
`ok`	There is no applicable live patch
`ok_livepatch_complete`	An applicable live patch exists for every required component
`ok_livepatch_incomplete`	An applicable live patch exists but it is not sufficient

Enum network_default_locking_mode

NAMED VALUE	DESCRIPTION
`disabled`	Treat all VIFs on this network with locking_mode = 'default' as if they have locking_mode = 'disabled'

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`unlocked`	Treat all VIFs on this network with locking_mode = 'default' as if they have locking_mode = 'unlocked'

Enum network_operations

NAMED VALUE	DESCRIPTION
`attaching`	Indicates this network is attaching to a VIF or PIF

Enum network_purpose

NAMED VALUE	DESCRIPTION
`insecure_nbd`	Network Block Device service without integrity or confidentiality: NOT RECOMMENDED
`nbd`	Network Block Device service using TLS

Enum on_boot

NAMED VALUE	DESCRIPTION
`persist`	Standard behaviour.
`reset`	When a VM containing this VDI is started, the contents of the VDI are reset to the state they were in when this flag was last set.

Enum on_crash_behaviour

NAMED VALUE	DESCRIPTION
`coredump_and_destroy`	record a coredump and then destroy the VM state
`coredump_and_restart`	record a coredump and then restart the VM
`destroy`	destroy the VM state

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`preserve`	leave the crashed VM paused
`rename_restart`	rename the crashed VM and start a new copy
`restart`	restart the VM

Enum on_normal_exit

NAMED VALUE	DESCRIPTION
`destroy`	destroy the VM state
`restart`	restart the VM

Enum on_softreboot_behavior

NAMED VALUE	DESCRIPTION
`destroy`	destroy the VM state
`preserve`	leave the VM paused
`restart`	restart the VM
`soft_reboot`	perform soft-reboot

Enum origin

NAMED VALUE	DESCRIPTION
`bundle`	The origin of the repository is a local bundle file
`remote`	The origin of the repository is a remote one
`remote_pool`	The origin of the repository is a remote pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum pci_dom0_access

NAMED VALUE	DESCRIPTION
`disable_on_reboot`	On host reboot dom0 will be blocked from accessing this device
`disabled`	dom0 cannot access this device
`enable_on_reboot`	On host reboot dom0 will be allowed to access this device
`enabled`	dom0 can access this device as normal

Enum persistence_backend

NAMED VALUE	DESCRIPTION
`xapi`	This VTPM is persisted in XAPI's DB

Enum pif_igmp_status

NAMED VALUE	DESCRIPTION
`disabled`	IGMP Snooping is disabled in the corresponding backend bridge.'
`enabled`	IGMP Snooping is enabled in the corresponding backend bridge.'
`unknown`	IGMP snooping status is unknown. If this is a VLAN master, then please consult the underlying VLAN slave PIF.

Enum placement_policy

NAMED VALUE	DESCRIPTION
`anti_affinity`	Anti-affinity placement policy
`normal`	Default placement policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum pool_allowed_operations

NAMED VALUE	DESCRIPTION
`apply_updates`	Indicates this pool is in the process of applying updates
`cert_refresh`	A certificate refresh and distribution is in progress
`cluster_create`	Indicates this pool is in the process of creating a cluster
`configure_repositories`	Indicates this pool is in the process of configuring repositories
`copy_primary_host_certs`	Indicates the primary host is sending its certificates to another host
`designate_new_master`	Indicates this pool is in the process of changing master
`eject`	Ejection of a host from the pool is under way
`exchange_ca_certificates_on_join`	Indicates this pool is exchanging ca certificates with a new joiner
`exchange_certificates_on_join`	Indicates this pool is exchanging internal certificates with a new joiner
`exchange_crls_on_join`	Indicates this pool is exchanging CRLs with a new joiner
`get_updates`	Indicates this pool is in the process of getting updates
`ha_disable`	Indicates this pool is in the process of disabling HA
`ha_enable`	Indicates this pool is in the process of enabling HA
`sync_bundle`	Indicates this pool is in the process of syncing bundle

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`sync_updates`	Indicates this pool is in the process of syncing updates
`tls_verification_enable`	Indicates this pool is in the process of enabling TLS verification

Enum pool_guest_secureboot_readiness

NAMED VALUE	DESCRIPTION
`not_ready`	Pool is not ready for SecureBoot, mandatory auth files are missing
`ready`	Pool is ready for SecureBoot, all auth files are present
`ready_no_dbx`	Pool is ready for SecureBoot, but there is no dbx auth file

Enum primary_address_type

NAMED VALUE	DESCRIPTION
`IPv4`	Primary address is the IPv4 address
`IPv6`	Primary address is the IPv6 address

Enum pvs_proxy_status

NAMED VALUE	DESCRIPTION
`caching`	The proxy is currently caching data
`incompatible_protocol_version`	The PVS protocol in use is not compatible with the PVS proxy
`incompatible_write_cache_mode`	The PVS device is configured to use an incompatible write-cache mode
`initialised`	The proxy is setup but has not yet cached anything

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`stopped`	The proxy is not currently running

Enum sdn_controller_protocol

NAMED VALUE	DESCRIPTION
`pssl`	Passive ssl connection
`ssl`	Active ssl connection

Enum sr_health

NAMED VALUE	DESCRIPTION
`healthy`	Storage is fully available
`recovering`	Storage is busy recovering, for example, rebuilding mirrors.
`unavailable`	Storage is unavailable, a host reboot will be required
`unreachable`	Storage is unreachable but may be recoverable with admin intervention

Enum sriov_configuration_mode

NAMED VALUE	DESCRIPTION
`manual`	Configure network sriov manually
`modprobe`	Configure network sriov by modprobe, need reboot
`sysfs`	Configure network sriov by sysfs, do not need reboot
`unknown`	Unknown mode

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum storage_operations

NAMED VALUE	DESCRIPTION
`destroy`	Destroying the SR
`forget`	Forgetting about SR
`pbd_create`	Creating a PBD for this SR
`pbd_destroy`	Destroying one of this SR's PBDs
`plug`	Plugging a PBD into this SR
`scan`	Scanning backends for new or deleted VDIs
`unplug`	Unplugging a PBD from this SR
`update`	Refresh the fields on the SR
`vdi_blocked`	Blocking other operations for a VDI
`vdi_clone`	Cloneing a VDI
`vdi_copy`	Copying the VDI
`vdi_create`	Creating a new VDI
`vdi_data_destroy`	Deleting the data of the VDI
`vdi_destroy`	Destroying a VDI
`vdi_disable_cbt`	Disabling changed block tracking for a VDI
`vdi_enable_cbt`	Enabling changed block tracking for a VDI
`vdi_force_unlock`	Forcefully unlocking the VDI
`vdi_forget`	Forgetting about the VDI
`vdi_generate_config`	Generating the configuration of the VDI
`vdi_introduce`	Introducing a new VDI

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`vdi_list_changed_blocks`	Exporting a bitmap that shows the changed blocks between two VDIs
`vdi_mirror`	Mirroring a VDI
`vdi_resize`	Resizing a VDI
`vdi_resize_online`	Resizing the VDI online
`vdi_revert`	Reverting a VDI to the snapshot
`vdi_set_on_boot`	Setting the on_boot field of the VDI
`vdi_snapshot`	Snapshotting a VDI
`vdi_update`	Refreshing the fields on the VDI

Enum task_allowed_operations

NAMED VALUE	DESCRIPTION
`cancel`	refers to the operation "cancel"
`destroy`	refers to the operation "destroy"

Enum task_status_type

NAMED VALUE	DESCRIPTION
`cancelled`	task has been cancelled
`cancelling`	task is being cancelled
`failure`	task has failed
`pending`	task is in progress
`success`	task was completed successfully

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum telemetry_frequency

NAMED VALUE	DESCRIPTION
`daily`	Run telemetry task daily
`monthly`	Run telemetry task monthly
`weekly`	Run telemetry task weekly

Enum tristate_type

NAMED VALUE	DESCRIPTION
`no`	Known to be false
`unspecified`	Unknown or unspecified
`yes`	Known to be true

Enum tunnel_protocol

NAMED VALUE	DESCRIPTION
`gre`	GRE protocol
`vxlan`	VxLAN Protocol

Enum update_after_apply_guidance

NAMED VALUE	DESCRIPTION
`restartHost`	This update requires the host to be restarted once applied.
`restartHVM`	This update requires HVM guests to be restarted once applied.
`restartPV`	This update requires PV guests to be restarted once applied.
`restartXAPI`	This update requires XAPI to be restarted once applied.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum update_guidances

NAMED VALUE	DESCRIPTION
`reboot_host`	Indicates the updated host should reboot as soon as possible
`reboot_host_on_kernel_livepatch_failure`	Indicates the updated host should reboot as soon as possible since one or more kernel livepatch(es) failed to be applied.
`reboot_host_on_livepatch_failure`	Indicates the updated host should reboot as soon as possible since one or more livepatch(es) failed to be applied.
`reboot_host_on_xen_livepatch_failure`	Indicates the updated host should reboot as soon as possible since one or more xen livepatch(es) failed to be applied.
`restart_device_model`	Indicates the device model of a running VM should restart as soon as possible
`restart_toolstack`	Indicates the Toolstack running on the updated host should restart as soon as possible
`restart_vm`	Indicates the VM should restart as soon as possible

Enum update_sync_frequency

NAMED VALUE	DESCRIPTION
`daily`	The update synchronizations happen every day
`weekly`	The update synchronizations happen every week on the chosen day

Enum vbd_mode

NAMED VALUE	DESCRIPTION
`RO`	only read-only access will be allowed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`RW`	read-write access will be allowed

Enum vbd_operations

NAMED VALUE	DESCRIPTION
`attach`	Attempting to attach this VBD to a VM
`eject`	Attempting to eject the media from this VBD
`insert`	Attempting to insert new media into this VBD
`pause`	Attempting to pause a block device backend
`plug`	Attempting to hotplug this VBD
`unpause`	Attempting to unpause a block device backend
`unplug`	Attempting to hot unplug this VBD
`unplug_force`	Attempting to forcibly unplug this VBD

Enum vbd_type

NAMED VALUE	DESCRIPTION
`CD`	VBD will appear to guest as CD
`Disk`	VBD will appear to guest as disk
`Floppy`	VBD will appear as a floppy

Enum vdi_operations

NAMED VALUE	DESCRIPTION
`blocked`	Operations on this VDI are temporarily blocked
`clone`	Cloning the VDI
`copy`	Copying the VDI

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`data_destroy`	Deleting the data of the VDI
`destroy`	Destroying the VDI
`disable_cbt`	Disabling changed block tracking for a VDI
`enable_cbt`	Enabling changed block tracking for a VDI
`force_unlock`	Forcibly unlocking the VDI
`forget`	Forget about the VDI
`generate_config`	Generating static configuration
`list_changed_blocks`	Exporting a bitmap that shows the changed blocks between two VDIs
`mirror`	Mirroring the VDI
`resize`	Resizing the VDI
`resize_online`	Resizing the VDI which may or may not be online
`revert_from`	Reverting this VDI to a clone of a snapshot
`revert_to`	Reverting a VDI to a clone of this snapshot
`set_on_boot`	Setting the on_boot field of the VDI
`snapshot`	Snapshotting the VDI
`update`	Refreshing the fields of the VDI

Enum vdi_type

NAMED VALUE	DESCRIPTION
`cbt_metadata`	Metadata about a snapshot VDI that has been deleted: the set of blocks that changed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
	between some previous version of the disk and the version tracked by the snapshot.
`crashdump`	a disk that stores VM crashdump information
`ephemeral`	a disk that may be reformatted on upgrade
`ha_statefile`	a disk used for HA storage heartbeating
`metadata`	a disk used for HA Pool metadata
`pvs_cache`	a disk that stores PVS cache data
`redo_log`	a disk used for a general metadata redo-log
`rrd`	a disk that stores SR-level RRDs
`suspend`	a disk that stores a suspend image
`system`	a disk that may be replaced on upgrade
`user`	a disk that is always preserved on upgrade

Enum vgpu_type_implementation

NAMED VALUE	DESCRIPTION
`gvt_g`	vGPU using Intel GVT-g
`mxgpu`	vGPU using AMD MxGPU
`nvidia`	vGPU using NVIDIA hardware
`nvidia_sriov`	vGPU using NVIDIA hardware with SR-IOV
`passthrough`	Pass through an entire physical GPU to a guest

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vif_ipv4_configuration_mode

NAMED VALUE	DESCRIPTION
`DHCP`	Acquire an IP address by DHCP
`None`	Follow the default IPv4 configuration of the guest (this is guest-dependent)
`Static`	Static IPv4 address configuration

Enum vif_ipv6_configuration_mode

NAMED VALUE	DESCRIPTION
`Autoconf`	Acquire an IPv6 address automatically
`None`	Follow the default IPv6 configuration of the guest (this is guest-dependent)
`Static`	Static IPv6 address configuration

Enum vif_locking_mode

NAMED VALUE	DESCRIPTION
`disabled`	No traffic is permitted
`locked`	Only traffic to a specific MAC and a list of IPv4 or IPv6 addresses is permitted
`network_default`	No specific configuration set - default network policy applies
`unlocked`	All traffic is permitted

Enum vif_operations

NAMED VALUE	DESCRIPTION
`attach`	Attempting to attach this VIF to a VM
`plug`	Attempting to hotplug this VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`unplug`	Attempting to hot unplug this VIF

Enum vm_appliance_operation

NAMED VALUE	DESCRIPTION
`clean_shutdown`	Clean shutdown
`hard_shutdown`	Hard shutdown
`shutdown`	Shutdown
`start`	Start

Enum vm_operations

NAMED VALUE	DESCRIPTION
`assert_operation_valid`	
`awaiting_memory_live`	Waiting for the memory settings to change
`call_plugin`	refers to the operation "call_plugin"
`changing_dynamic_range`	Changing the memory dynamic range
`changing_memory_limits`	Changing the memory limits
`changing_memory_live`	Changing the memory settings
`changing_NVRAM`	Changing NVRAM for a halted VM.
`changing_shadow_memory`	Changing the shadow memory for a halted VM.
`changing_shadow_memory_live`	Changing the shadow memory for a running VM.
`changing_static_range`	Changing the memory static range
`changing_VCPUs`	Changing VCPU settings for a halted VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`changing_VCPUs_live`	Changing VCPU settings for a running VM.
`checkpoint`	refers to the operation "checkpoint"
`clean_reboot`	refers to the operation "clean_reboot"
`clean_shutdown`	refers to the operation "clean_shutdown"
`clone`	refers to the operation "clone"
`copy`	refers to the operation "copy"
`create_template`	refers to the operation "create_template"
`create_vtpm`	Creating and adding a VTPM to this VM
`csvm`	refers to the operation "csvm"
`data_source_op`	Add, remove, query or list data sources
`destroy`	refers to the act of uninstalling the VM
`export`	exporting a VM to a network stream
`get_boot_record`	refers to the operation "get_boot_record"
`hard_reboot`	refers to the operation "hard_reboot"
`hard_shutdown`	refers to the operation "hard_shutdown"
`import`	importing a VM from a network stream
`make_into_template`	Turning this VM into a template
`metadata_export`	exporting VM metadata to a network stream
`migrate_send`	refers to the operation "migrate_send"
`pause`	refers to the operation "pause"
`pool_migrate`	refers to the operation "pool_migrate"

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`power_state_reset`	refers to the operation "power_state_reset"
`provision`	refers to the operation "provision"
`query_services`	refers to the operation "query_services"
`resume`	refers to the operation "resume"
`resume_on`	refers to the operation "resume_on"
`revert`	refers to the operation "revert"
`reverting`	Reverting the VM to a previous snapshotted state
`send_sysrq`	refers to the operation "send_sysrq"
`send_trigger`	refers to the operation "send_trigger"
`shutdown`	refers to the operation "shutdown"
`snapshot`	refers to the operation "snapshot"
`snapshot_with_quiesce`	refers to the operation "snapshot_with_quiesce"
`start`	refers to the operation "start"
`start_on`	refers to the operation "start_on"
`suspend`	refers to the operation "suspend"
`sysprep`	Performing a Windows sysprep on this VM
`unpause`	refers to the operation "unpause"
`update_allowed_operations`	

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vm_power_state

NAMED VALUE	DESCRIPTION
`Halted`	VM is offline and not using any resources
`Paused`	All resources have been allocated but the VM itself is paused and its vCPUs are not running
`Running`	Running
`Suspended`	VM state has been saved to disk and it is nolonger running. Note that disks remain in-use while the VM is suspended.

Enum vm_secureboot_certificates_state

NAMED VALUE	DESCRIPTION
`ok`	The VM's certificates do not need to be updated (including the case where Secure Boot does not apply to this VM, for example, BIOS VM).
`update_available`	The Secure Boot certificates are due to expire or have already expired.
`update_on_boot`	An update of the certificates will be triggered whenever the VM boots. This includes VM.start, VM.reboot and a guest-triggered reboot.

Enum vm_secureboot_readiness

NAMED VALUE	DESCRIPTION
`certs_incomplete`	Secured boot is enabled on this VM and the certificates defined in its EFI variables are incomplete
`disabled`	Secureboot is disabled on this VM
`first_boot`	Secured boot is enabled on this VM and its NVRAM.EFI-variables are empty

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`not_supported`	VM's firmware is not UEFI
`ready`	Secured boot is enabled on this VM and PK, KEK, db and dbx are defined in its EFI variables
`ready_no_dbx`	Secured boot is enabled on this VM and PK, KEK, db but not dbx are defined in its EFI variables
`setup_mode`	Secured boot is enabled on this VM and PK is not defined in its EFI variables

Enum vm_uefi_mode

NAMED VALUE	DESCRIPTION
`setup`	clears a VM's EFI variables related to Secure Boot and places it into Setup Mode
`user`	resets a VM's EFI variables related to Secure Boot to the defaults, placing it into User Mode

Enum vmpp_archive_frequency

NAMED VALUE	DESCRIPTION
`always_after_backup`	Archive after backup
`daily`	Daily archives
`never`	Never archive
`weekly`	Weekly backups

Enum vmpp_archive_target_type

NAMED VALUE	DESCRIPTION
`cifs`	CIFS target config

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NAMED VALUE	DESCRIPTION
`nfs`	NFS target config
`none`	No target config

Enum vmpp_backup_frequency

NAMED VALUE	DESCRIPTION
`daily`	Daily backups
`hourly`	Hourly backups
`weekly`	Weekly backups

Enum vmpp_backup_type

NAMED VALUE	DESCRIPTION
`checkpoint`	The backup is a checkpoint
`snapshot`	The backup is a snapshot

Enum vmss_frequency

NAMED VALUE	DESCRIPTION
`daily`	Daily snapshots
`hourly`	Hourly snapshots
`weekly`	Weekly snapshots

Enum vmss_type

NAMED VALUE	DESCRIPTION
`checkpoint`	The snapshot is a checkpoint
`snapshot`	The snapshot is a disk snapshot
`snapshot_with_quiesce`	Support for VSS has been removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enum vtpm_operations

NAMED VALUE	DESCRIPTION
`destroy`	Destroy a VTPM

Enum vusb_operations

NAMED VALUE	DESCRIPTION
`attach`	Attempting to attach this VUSB to a VM
`plug`	Attempting to plug this VUSB into a VM
`unplug`	Attempting to hot unplug this VUSB

Class: auth

Management of remote authentication services

Fields for class: auth

Class auth has no fields.

RPCs associated with class: auth

RPC name: getgroupmembership

Overview:

This calls queries the external directory service to obtain the transitivelyclosed set of groups that the the subjectidentifier is member of.

Signature:

```
string set get_group_membership (session ref session_ref, string
subject_identifier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`subject_identifier`	A string containing the subject_identifier, unique in the external directory service

Minimum Role: read-only

Return Type: `string set`

set of subjectidentifiers that provides the group membership of subjectidentifier passed as argument, it contains, recursively, all groups a subjectidentifier is member of.

RPC name: getsubjectidentifier

Overview:

This call queries the external directory service to obtain the subjectidentifier as a string from the humanreadable subjectname

Signature:

```
string get_subject_identifier (session ref session_ref, string
subject_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`subject_name`	The human-readable subject_name, such as a username or a groupname

Minimum Role: read-only

Return Type: `string`

the subjectidentifier obtained from the external directory service

RPC name: getsubjectinformationfromidentifier

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

This call queries the external directory service to obtain the user information for example, username, organization etc from the specified subjectidentifier

Signature:

```
(string -> string) map get_subject_information_from_identifier (session ref session_ref, string subject_identifier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`subject_identifier`	A string containing the subject_identifier, unique in the external directory service

Minimum Role: read-only

Return Type: `(string -> string) map`

keyvalue pairs containing at least a key called subjectname

Class: blob

A placeholder for a binary blob

Fields for class: blob

FIELD	TYPE	QUALIFIER	DESCRIPTION
`last_updated`	`datetime`	*RO/constructor*	Time at which the data in the blob was last updated
`mime_type`	`string`	*RO/constructor*	The mime type associated with this object. Defaults to 'application/octet-stream' if the empty string is supplied

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`public`	`bool`	*RW*	True if the blob is publicly accessible
`size`	`int`	*RO/runtime*	Size of the binary data, in bytes
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: blob

RPC name: create

Overview:

Create a placeholder for a binary blob

Signature:

```
blob ref create (session ref session_ref, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`mime_type`	The mime-type of the blob. Defaults to 'application/octet-stream' if the empty string is supplied
`bool`	`public`	True if the blob should be publicly available

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the created blob

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	The reference of the blob to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the blobs known to the system.

Signature:

```
blob ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `blob ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of blob references to blob records for all blobs known to the system.

Signature:

```
(blob ref -> blob record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(blob ref -> blob record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the blob instances with the given label.

Signature:

```
blob ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `blob ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the blob instance with the specified UUID.

Signature:

```
blob ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `blob ref`

reference to the object

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given blob.

Signature:

```
datetime get_last_updated (session ref session_ref, blob ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getmimetype

Overview:

Get the mimetype field of the given blob.

Signature:

```
string get_mime_type (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given blob.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_name_description (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given blob.

Signature:

```
string get_name_label (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpublic

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the public field of the given blob.

Signature:

```
bool get_public (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given blob.

Signature:

```
blob record get_record (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `blob record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsize

Overview:

Get the size field of the given blob.

Signature:

```
int get_size (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given blob.

Signature:

```
string get_uuid (session ref session_ref, blob ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: setnamedescription

Overview:

Set the name/description field of the given blob.

Signature:

```
void set_name_description (session ref session_ref, blob ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given blob.

Signature:

```
void set_name_label (session ref session_ref, blob ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setpublic

Overview:

Set the public field of the given blob.

Signature:

```
void set_public (session ref session_ref, blob ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`blob ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: Bond

A Network bond that combines physical network interfaces, also known as link aggregation

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Bond

FIELD	TYPE	QUALIFIER	DESCRIPTION
`auto_update_mac`	`bool`	*RO/runtime*	true if the MAC was taken from the primary slave when the bond was created, and false if the client specified the MAC
`links_up`	`int`	*RO/runtime*	Number of links up in this bond
`master`	`PIF ref`	*RO/constructor*	The bonded interface
`mode`	`bond_mode`	*RO/runtime*	The algorithm used to distribute traffic among the bonded NICs
`other_config`	`(string -> string) map`	*RW*	additional configuration
`primary_slave`	`PIF ref`	*RO/runtime*	The PIF of which the IP configuration and MAC were copied to the bond, and which will receive all configuration/VLANs/VIFs on the bond if the bond is destroyed
`properties`	`(string -> string) map`	*RO/runtime*	Additional configuration properties specific to the bond mode.
`slaves`	`PIF ref set`	*RO/runtime*	The interfaces which are part of this bond

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: Bond

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given Bond.

Signature:

```
void add_to_other_config (session ref session_ref, Bond ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create an interface bond

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Bond ref create (session ref session_ref, network ref network, PIF ref set members, string MAC, bond_mode mode, (string -> string) map properties)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	Network to add the bonded PIF to
`PIF ref set`	`members`	PIFs to add to this bond
`string`	`MAC`	The MAC address to use on the bond itself. If this parameter is the empty string then the bond will inherit its MAC address from the primary slave.
`bond_mode`	`mode`	Bonding mode to use for the new bond
`(string -> string) map`	`properties`	Additional configuration parameters specific to the bond mode

Minimum Role: pool-operator

Return Type: `Bond ref`

The reference of the created Bond object

RPC name: destroy

Overview:

Destroy an interface bond

Signature:

```
void destroy (session ref session_ref, Bond ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	Bond to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the Bonds known to the system.

Signature:

```
Bond ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Bond ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Bond references to Bond records for all Bonds known to the system.

Signature:

```
(Bond ref -> Bond record) map get_all_records (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Bond ref -> Bond record) map`

records of all objects

RPC name: getautoupdatemac

Overview:

Get the autoupdatemac field of the given Bond.

Signature:

```
bool get_auto_update_mac (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the Bond instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Bond ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Bond ref`

reference to the object

RPC name: getlinksup

Overview:

Get the linksup field of the given Bond.

Signature:

```
int get_links_up (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmaster

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the master field of the given Bond.

Signature:

```
PIF ref get_master (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getmode

Overview:

Get the mode field of the given Bond.

Signature:

```
bond_mode get_mode (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bond_mode`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given Bond.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprimaryslave

Overview:

Get the primaryslave field of the given Bond.

Signature:

```
PIF ref get_primary_slave (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getproperties

Overview:

Get the properties field of the given Bond.

Signature:

```
(string -> string) map get_properties (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Bond.

Signature:

```
Bond record get_record (session ref session_ref, Bond ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Bond record`

all fields from the object

RPC name: getslaves

Overview:

Get the slaves field of the given Bond.

Signature:

```
PIF ref set get_slaves (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Bond.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, Bond ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given Bond. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, Bond ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setmode

Overview:

Change the bond mode

Signature:

```
void set_mode (session ref session_ref, Bond ref self, bond_mode value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	The bond
`bond_mode`	`value`	The new bond mode

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given Bond.

Signature:

```
void set_other_config (session ref session_ref, Bond ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setproperty

Overview:

Set the value of a property of the bond

Signature:

```
void set_property (session ref session_ref, Bond ref self, string name,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Bond ref`	`self`	The bond
`string`	`name`	The property name
`string`	`value`	The property value

Minimum Role: pool-operator

Return Type: `void`

Class: Certificate

An X509 certificate used for TLS connections

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Certificate

FIELD	TYPE	QUALIFIER	DESCRIPTION
`fingerprint`	`string`	*RO/constructor*	**Deprecated.** Use fingerprint_sha256 instead
`fingerprint_sha1`	`string`	*RO/constructor*	The certificate's SHA1 fingerprint / hash
`fingerprint_sha256`	`string`	*RO/constructor*	The certificate's SHA256 fingerprint / hash
`host`	`host ref`	*RO/constructor*	The host where the certificate is installed
`name`	`string`	*RO/runtime*	The name of the certificate, only present on certificates of type 'ca'
`not_after`	`datetime`	*RO/constructor*	Date before which the certificate is valid
`not_before`	`datetime`	*RO/constructor*	Date after which the certificate is valid
`purpose`	`certificate_purpose set`	*RO/constructor*	The purposes of the certificate
`type`	`certificate_type`	*RO/runtime*	The type of the certificate, either 'ca', 'host' or 'host_internal'
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: Certificate

RPC name: getall

Overview:

Return a list of all the Certificates known to the system.

Signature:

```
Certificate ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Certificate ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Certificate references to Certificate records for all Certificates known to the system.

Signature:

```
(Certificate ref -> Certificate record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(Certificate ref -> Certificate record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the Certificate instance with the specified UUID.

Signature:

```
Certificate ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Certificate ref`

reference to the object

RPC name: getfingerprint

This message is deprecated.

Overview:

Get the fingerprint field of the given Certificate.

Signature:

```
string get_fingerprint (session ref session_ref, Certificate ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfingerprintsha1

Overview:

Get the fingerprintsha1 field of the given Certificate.

Signature:

```
string get_fingerprint_sha1 (session ref session_ref, Certificate ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfingerprintsha256

Overview:

Get the fingerprintsha256 field of the given Certificate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_fingerprint_sha256 (session ref session_ref, Certificate ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given Certificate.

Signature:

```
host ref get_host (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getname

Overview:

Get the name field of the given Certificate.

Signature:

```
string get_name (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnotafter

Overview:

Get the notafter field of the given Certificate.

Signature:

```
datetime get_not_after (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `datetime`

value of the field

RPC name: getnotbefore

Overview:

Get the notbefore field of the given Certificate.

Signature:

```
datetime get_not_before (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getpurpose

Overview:

Get the purpose field of the given Certificate.

Signature:

```
certificate_purpose set get_purpose (session ref session_ref, Certificate
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `certificate_purpose set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Certificate.

Signature:

```
Certificate record get_record (session ref session_ref, Certificate ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Certificate record`

all fields from the object

RPC name: gettype

Overview:

Get the type field of the given Certificate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
certificate_type get_type (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `certificate_type`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Certificate.

Signature:

```
string get_uuid (session ref session_ref, Certificate ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Certificate ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: Cluster

Clusterwide Cluster metadata

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Cluster

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`cluster_operation set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`cluster_config`	`(string -> string) map`	*RO/constructor*	Contains read-only settings for the cluster, such as timeouts and other options. It can only be set at cluster create time
`cluster_hosts`	`Cluster_host ref set`	*RO/runtime*	A list of the cluster_host objects associated with the Cluster
`cluster_stack`	`string`	*RO/constructor*	Simply the string 'corosync'. No other cluster stacks are currently supported
`cluster_stack_version`	`int`	*RO/constructor*	Version of cluster stack, not writable via the API. Defaulting to 2 for backwards compatibility when upgrading from a cluster without this field, which means it is necessarily running version 2 of corosync, the only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			cluster stack supported so far.
`cluster_token`	`string`	*RO/constructor*	The secret key used by xapi-clusterd when it talks to itself on other hosts
`current_operations`	`(string -> cluster_operation) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`expected_hosts`	`int`	*RO/runtime*	Total number of hosts expected by the cluster stack
`is_quorate`	`bool`	*RO/runtime*	Whether the cluster stack thinks the cluster is quorate
`live_hosts`	`int`	*RO/runtime*	Current number of live hosts, according to the cluster stack
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`pending_forget`	`string set`	*RO/runtime*	Internal field used by Host.destroy to store the IP of cluster members marked as permanently dead but not yet removed
`pool_auto_join`	`bool`	*RO/constructor*	True if automatically joining new pool members to the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			cluster. This will be `true` in the first release
`quorum`	`int`	*RO/runtime*	Number of live hosts in order to be quorate
`token_timeout`	`float`	*RO/constructor*	The corosync token timeout in seconds
`token_timeout_coefficient`	`float`	*RO/constructor*	The corosync token timeout coefficient in seconds
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: Cluster

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given Cluster.

Signature:

```
void add_to_other_config (session ref session_ref, Cluster ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Creates a Cluster object and one Clusterhost object as its first member

Signature:

```
Cluster ref create (session ref session_ref, PIF ref PIF, string
cluster_stack, bool pool_auto_join, float token_timeout, float
token_timeout_coefficient)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`PIF`	The PIF to connect the cluster's first cluster_host to
`string`	`cluster_stack`	simply the string 'corosync'. No other cluster stacks are currently supported
`bool`	`pool_auto_join`	true if xapi is automatically joining new pool members to the cluster
`float`	`token_timeout`	Corosync token timeout in seconds
`float`	`token_timeout_coefficient`	Corosync token timeout coefficient in seconds

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `Cluster ref`

the new Cluster

Possible Error Codes: `INVALID_CLUSTER_STACK`, `INVALID_VALUE`, `PIF_ALLOWS_UNPLUG`, `REQUIRED_PIF_IS_UNPLUGGED`

RPC name: destroy

Overview:

Destroys a Cluster object and the one remaining Clusterhost member

Signature:

```
void destroy (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	the Cluster to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_DOES_NOT_HAVE_ONE_NODE`, `CLUSTER_STACK_IN_USE`

RPC name: getall

Overview:

Return a list of all the Clusters known to the system.

Signature:

```
Cluster ref set get_all (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Cluster ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Cluster references to Cluster records for all Clusters known to the system.

Signature:

```
(Cluster ref -> Cluster record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Cluster ref -> Cluster record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given Cluster.

Signature:

```
cluster_operation set get_allowed_operations (session ref session_ref,
Cluster ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `cluster_operation set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the Cluster instance with the specified UUID.

Signature:

```
Cluster ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Cluster ref`

reference to the object

RPC name: getclusterconfig

Overview:

Get the clusterconfig field of the given Cluster.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_cluster_config (session ref session_ref,
Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getclusterhosts

Overview:

Get the clusterhosts field of the given Cluster.

Signature:

```
Cluster_host ref set get_cluster_hosts (session ref session_ref, Cluster
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster_host ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getclusterstack

Overview:

Get the clusterstack field of the given Cluster.

Signature:

```
string get_cluster_stack (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getclusterstackversion

Overview:

Get the clusterstackversion field of the given Cluster.

Signature:

```
int get_cluster_stack_version (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getclustertoken

Overview:

Get the clustertoken field of the given Cluster.

Signature:

```
string get_cluster_token (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given Cluster.

Signature:

```
(string -> cluster_operation) map get_current_operations (session ref
session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> cluster_operation) map`

value of the field

RPC name: getexpectedhosts

Overview:

Get the expectedhosts field of the given Cluster.

Signature:

```
int get_expected_hosts (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getisquorate

Overview:

Get the isquorate field of the given Cluster.

Signature:

```
bool get_is_quorate (session ref session_ref, Cluster ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlivehosts

Overview:

Get the livehosts field of the given Cluster.

Signature:

```
int get_live_hosts (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnetwork

Overview:

Returns the network used by the cluster for interhost communication, i.e. the network shared by all cluster host PIFs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
network ref get_network (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	the Cluster with the network

Minimum Role: read-only

Return Type: `network ref`

network of cluster

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given Cluster.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, Cluster
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpendingforget

Overview:

Get the pendingforget field of the given Cluster.

Signature:

```
string set get_pending_forget (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getpoolautojoin

Overview:

Get the poolautojoin field of the given Cluster.

Signature:

```
bool get_pool_auto_join (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getquorum

Overview:

Get the quorum field of the given Cluster.

Signature:

```
int get_quorum (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Cluster.

Signature:

```
Cluster record get_record (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster record`

all fields from the object

RPC name: gettokentimeout

Overview:

Get the tokentimeout field of the given Cluster.

Signature:

```
float get_token_timeout (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: gettokentimeoutcoefficient

Overview:

Get the tokentimeoutcoefficient field of the given Cluster.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
float get_token_timeout_coefficient (session ref session_ref, Cluster ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Cluster.

Signature:

```
string get_uuid (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: poolcreate

Overview:

Attempt to create a Cluster from the entire pool

Signature:

```
Cluster ref pool_create (session ref session_ref, network ref network,
string cluster_stack, float token_timeout, float
token_timeout_coefficient)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	the single network on which corosync carries out its inter-host communications
`string`	`cluster_stack`	simply the string 'corosync'. No other cluster stacks are currently supported
`float`	`token_timeout`	Corosync token timeout in seconds
`float`	`token_timeout_coefficient`	Corosync token timeout coefficient in seconds

Minimum Role: pool-operator

Return Type: `Cluster ref`

the new Cluster

RPC name: pooldestroy

Overview:

Attempt to destroy the Clusterhost objects for all hosts in the pool and then destroy the Cluster.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void pool_destroy (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	The cluster to destroy.

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_STACK_IN_USE`, `CLUSTERING_DISABLED`, `CLUSTER_HOST_IS_LAST`

RPC name: poolforcedestroy

Overview:

Attempt to force destroy the Clusterhost objects, and then destroy the Cluster.

Signature:

```
void pool_force_destroy (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	The cluster to force destroy.

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_FORCE_DESTROY_FAILED`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: poolresync

Overview:

Resynchronise the clusterhost objects across the pool. Creates them where they need creating and then plugs them

Signature:

```
void pool_resync (session ref session_ref, Cluster ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	The cluster to resync

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given Cluster. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, Cluster ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given Cluster.

Signature:

```
void set_other_config (session ref session_ref, Cluster ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: Clusterhost

Cluster member metadata

Fields for class: Clusterhost

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`cluster_host_operation set`	*RO/runtime*	list of the operations allowed in this state.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			This list is advisory only and the server state may have changed by the time this field is read by a client.
`cluster`	`Cluster ref`	*RO/constructor*	Reference to the Cluster object
`current_operations`	`(string -> cluster_host_operation) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`enabled`	`bool`	*RO/runtime*	Whether the cluster host believes that clustering should be enabled on this host. This field can be altered by calling the enable/disable message on a cluster host. Only enabled members run the underlying cluster stack. Disabled members are still considered a member of the cluster (see joined), and can be re-enabled by the user.
`host`	`host ref`	*RO/constructor*	Reference to the Host object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`joined`	`bool`	*RO/runtime*	Whether the cluster host has joined the cluster. Contrary to enabled, a host that is not joined is not considered a member of the cluster, and hence enable and disable operations cannot be performed on this host.
`last_update_live`	`datetime`	*RO/runtime*	Time when the live field was last updated based on information from the cluster stack
`live`	`bool`	*RO/runtime*	Whether the underlying cluster stack thinks we are live. This field is set automatically based on updates from the cluster stack and cannot be altered by the user.
`other_config`	`(string -> string) map`	*RO/constructor*	Additional configuration
`PIF`	`PIF ref`	*RO/constructor*	Reference to the PIF object
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: Clusterhost

RPC name: create

Overview:

Add a new host to an existing cluster.

Signature:

```
Cluster_host ref create (session ref session_ref, Cluster ref cluster,
host ref host, PIF ref pif)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster ref`	`cluster`	Cluster to join
`host ref`	`host`	new cluster member
`PIF ref`	`pif`	Network interface to use for communication

Minimum Role: pool-operator

Return Type: `Cluster_host ref`

the newly created clusterhost object

Possible Error Codes: `PIF_NOT_ATTACHED_TO_HOST`, `NO_CLUSTER_HOSTS_REACHABLE`

RPC name: destroy

Overview:

Remove the host from an existing cluster. This operation is allowed even if a cluster host is not enabled.

Signature:

```
void destroy (session ref session_ref, Cluster_host ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to remove from the cluster

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_STACK_IN_USE`, `CLUSTERING_DISABLED`, `CLUSTER_HOST_IS_LAST`

RPC name: disable

Overview:

Disable cluster membership for an enabled cluster host.

Signature:

```
void disable (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to disable

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTER_STACK_IN_USE`

RPC name: enable

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable cluster membership for a disabled cluster host.

Signature:

```
void enable (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to enable

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `PIF_ALLOWS_UNPLUG`, `REQUIRED_PIF_IS_UNPLUGGED`

RPC name: forcedestroy

Overview:

Remove a host from an existing cluster forcefully.

Signature:

```
void force_destroy (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	the cluster_host to remove from the cluster

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `CLUSTER_STACK_IN_USE`

RPC name: getall

Overview:

Return a list of all the Clusterhosts known to the system.

Signature:

```
Cluster_host ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Cluster_host ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Clusterhost references to Clusterhost records for all Clusterhosts known to the system.

Signature:

```
(Cluster_host ref -> Cluster_host record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(Cluster_host ref -> Cluster_host record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given Clusterhost.

Signature:

```
cluster_host_operation set get_allowed_operations (session ref
session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `cluster_host_operation set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the Clusterhost instance with the specified UUID.

Signature:

```
Cluster_host ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Cluster_host ref`

reference to the object

RPC name: getcluster

Overview:

Get the cluster field of the given Clusterhost.

Signature:

```
Cluster ref get_cluster (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster ref`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given Clusterhost.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> cluster_host_operation) map get_current_operations (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> cluster_host_operation) map`

value of the field

RPC name: getenabled

Overview:

Get the enabled field of the given Clusterhost.

Signature:

```
bool get_enabled (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given Clusterhost.

Signature:

```
host ref get_host (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getjoined

Overview:

Get the joined field of the given Clusterhost.

Signature:

```
bool get_joined (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getlastupdatelive

Overview:

Get the lastupdatelive field of the given Clusterhost.

Signature:

```
datetime get_last_update_live (session ref session_ref, Cluster_host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlive

Overview:

Get the live field of the given Clusterhost.

Signature:

```
bool get_live (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given Clusterhost.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPIF

Overview:

Get the PIF field of the given Clusterhost.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PIF ref get_PIF (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Clusterhost.

Signature:

```
Cluster_host record get_record (session ref session_ref, Cluster_host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Cluster_host record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given Clusterhost.

Signature:

```
string get_uuid (session ref session_ref, Cluster_host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Cluster_host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: console

A console

Fields for class: console

FIELD	TYPE	QUALIFIER	DESCRIPTION
`location`	`string`	*RO/runtime*	URI for the console service
`other_config`	`(string -> string) map`	*RW*	additional configuration
`protocol`	`console_protocol`	*RO/runtime*	the protocol used by this console
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VM`	`VM ref`	*RO/runtime*	VM to which this console is attached

RPCs associated with class: console

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given console.

Signature:

```
void add_to_other_config (session ref session_ref, console ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: create

Overview:

Create a new console instance, and return its handle.

Signature:

```
console ref create (session ref session_ref, console record args)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `console ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified console instance.

Signature:

```
void destroy (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the consoles known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
console ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `console ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of console references to console records for all consoles known to the system.

Signature:

```
(console ref -> console record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(console ref -> console record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the console instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
console ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `console ref`

reference to the object

RPC name: getlocation

Overview:

Get the location field of the given console.

Signature:

```
string get_location (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the otherconfig field of the given console.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprotocol

Overview:

Get the protocol field of the given console.

Signature:

```
console_protocol get_protocol (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `console_protocol`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given console.

Signature:

```
console record get_record (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `console record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given console.

Signature:

```
string get_uuid (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given console.

Signature:

```
VM ref get_VM (session ref session_ref, console ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given console. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, console ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given console.

Signature:

```
void set_other_config (session ref session_ref, console ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`console ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: crashdump

This class is deprecated.

A VM crashdump

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: crashdump

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference
`VDI`	`VDI ref`	*RO/constructor*	**Deprecated.** the virtual disk
`VM`	`VM ref`	*RO/constructor*	**Deprecated.** the virtual machine

RPCs associated with class: crashdump

RPC name: addtootherconfig

This message is deprecated.

Overview:

Add the given keyvalue pair to the otherconfig field of the given crashdump.

Signature:

```
void add_to_other_config (session ref session_ref, crashdump ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

This message is deprecated.

Overview:

Destroy the specified crashdump

Signature:

```
void destroy (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	The crashdump to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the crashdumps known to the system.

Signature:

```
crashdump ref set get_all (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `crashdump ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

Return a map of crashdump references to crashdump records for all crashdumps known to the system.

Signature:

```
(crashdump ref -> crashdump record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(crashdump ref -> crashdump record) map`

records of all objects

RPC name: getbyuuid

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a reference to the crashdump instance with the specified UUID.

Signature:

```
crashdump ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `crashdump ref`

reference to the object

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given crashdump.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given crashdump.

Signature:

```
crashdump record get_record (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `crashdump record`

all fields from the object

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given crashdump.

Signature:

```
string get_uuid (session ref session_ref, crashdump ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDI

This message is deprecated.

Overview:

Get the VDI field of the given crashdump.

Signature:

```
VDI ref get_VDI (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getVM

This message is deprecated.

Overview:

Get the VM field of the given crashdump.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM ref get_VM (session ref session_ref, crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given crashdump. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, crashdump ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object
`string`	`key`	Key to remove

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given crashdump.

Signature:

```
void set_other_config (session ref session_ref, crashdump ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`crashdump ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: datasource

Data sources for logging in RRDs

Fields for class: datasource

FIELD	TYPE	QUALIFIER	DESCRIPTION
`enabled`	`bool`	*RO/runtime*	true if the data source is being logged

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`max`	`float`	*RO/runtime*	the maximum value of the data source
`min`	`float`	*RO/runtime*	the minimum value of the data source
`name_description`	`string`	*RO/runtime*	a notes field containing human-readable description
`name_label`	`string`	*RO/runtime*	a human-readable name
`standard`	`bool`	*RO/runtime*	true if the data source is enabled by default. Non-default data sources cannot be disabled
`units`	`string`	*RO/runtime*	the units of the value
`value`	`float`	*RO/runtime*	current value of the data source

RPCs associated with class: datasource

Class datasource has no RPCs associated with it.

Class: DRtask

DR task

Fields for class: DRtask

FIELD	TYPE	QUALIFIER	DESCRIPTION
`introduced_SRs`	`SR ref set`	*RO/runtime*	All SRs introduced by this appliance
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: DRtask

RPC name: create

Overview:

Create a disaster recovery task which will query the supplied list of devices

Signature:

```
DR_task ref create (session ref session_ref, string type, (string ->
string) map device_config, string set whitelist)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`type`	The SR driver type of the SRs to introduce
`(string -> string) map`	`device_config`	The device configuration of the SRs to introduce
`string set`	`whitelist`	The devices to use for disaster recovery

Minimum Role: pool-operator

Return Type: `DR_task ref`

The reference of the created DRtask

RPC name: destroy

Overview:

Destroy the disaster recovery task, detaching and forgetting any SRs introduced which are no longer required

Signature:

```
void destroy (session ref session_ref, DR_task ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	The disaster recovery task to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the DRtasks known to the system.

Signature:

```
DR_task ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `DR_task ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of DRtask references to DRtask records for all DRtasks known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(DR_task ref -> DR_task record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(DR_task ref -> DR_task record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the DRtask instance with the specified UUID.

Signature:

```
DR_task ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `DR_task ref`

reference to the object

RPC name: getintroducedSRs

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the introducedSRs field of the given DRtask.

Signature:

```
SR ref set get_introduced_SRs (session ref session_ref, DR_task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given DRtask.

Signature:

```
DR_task record get_record (session ref session_ref, DR_task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `DR_task record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given DRtask.

Signature:

```
string get_uuid (session ref session_ref, DR_task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`DR_task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: Drivervariant

UNSUPPORTED. Variant of a host driver

Fields for class: Drivervariant

FIELD	TYPE	QUALIFIER	DESCRIPTION
`driver`	`Host_driver ref`	*RO/runtime*	Driver this variant is a part of
`hardware_present`	`bool`	*RO/runtime*	True if the hardware for this variant is present on the host
`name`	`string`	*RO/runtime*	Name identifying the driver variant within the driver

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`priority`	`float`	*RO/runtime*	Priority; this needs an explanation how this is ordered
`status`	`string`	*RO/runtime*	Development and release status of this variant, like 'alpha'
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`version`	`string`	*RO/runtime*	Unique version of this driver variant

RPCs associated with class: Drivervariant

RPC name: getall

Overview:

Return a list of all the Drivervariants known to the system.

Signature:

```
Driver_variant ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Driver_variant ref set`

references to all objects

RPC name: getallrecords

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return a map of Drivervariant references to Drivervariant records for all Drivervariants known to the system.

Signature:

```
(Driver_variant ref -> Driver_variant record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Driver_variant ref -> Driver_variant record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the Drivervariant instance with the specified UUID.

Signature:

```
Driver_variant ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Driver_variant ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getdriver

Overview:

Get the driver field of the given Drivervariant.

Signature:

```
Host_driver ref get_driver (session ref session_ref, Driver_variant ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Host_driver ref`

value of the field

RPC name: gethardwarepresent

Overview:

Get the hardwarepresent field of the given Drivervariant.

Signature:

```
bool get_hardware_present (session ref session_ref, Driver_variant ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getname

Overview:

Get the name field of the given Drivervariant.

Signature:

```
string get_name (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpriority

Overview:

Get the priority field of the given Drivervariant.

Signature:

```
float get_priority (session ref session_ref, Driver_variant ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Drivervariant.

Signature:

```
Driver_variant record get_record (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant record`

all fields from the object

RPC name: getstatus

Overview:

Get the status field of the given Drivervariant.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_status (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Drivervariant.

Signature:

```
string get_uuid (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the version field of the given Drivervariant.

Signature:

```
string get_version (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: select

Overview:

UNSUPPORTED Select this variant of a driver to become active after reboot or immediately if currently no version is active

Signature:

```
void select (session ref session_ref, Driver_variant ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Driver_variant ref`	`self`	Driver variant to become active (after reboot).

Minimum Role: pool-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: event

Asynchronous event registration and handling

Fields for class: event

FIELD	TYPE	QUALIFIER	DESCRIPTION
`class`	`string`	*RO/constructor*	The name of the class of the object that changed
`id`	`int`	*RO/constructor*	An ID, monotonically increasing, and local to the current session
`obj_uuid`	`string`	*RO/constructor*	**Deprecated.** The uuid of the object that changed
`operation`	`event_operation`	*RO/constructor*	The operation that was performed
`ref`	`string`	*RO/constructor*	A reference to the object that changed
`timestamp`	`datetime`	*RO/constructor*	**Deprecated.** The time at which the event occurred
snapshot	object record	*RO/runtime*	The record of the database object that was added, changed or deleted

RPCs associated with class: event

RPC name: from

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Blocking call which returns a new token and a possibly empty batch of events. The returned token can be used in subsequent calls to this function. It eliminates redundant events for example, same field updated multiple times.

Signature:

```
event batch from (session ref session_ref, string set classes, string
token, float timeout)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`classes`	register for events for the indicated classes
`string`	`token`	A token representing the point from which to generate database events. The empty string represents the beginning.
`float`	`timeout`	Return after this many seconds if no events match

Minimum Role: read-only

Return Type: `event batch`

a structure consisting of a token 'token', a map of valid references per object type 'validrefcounts', and a set of event records 'events'.

Possible Error Codes: `EVENT_FROM_TOKEN_PARSE_FAILURE`, `EVENT_SUBSCRIPTION_PARSE_FAILURE`

RPC name: getcurrentid

Overview:

Return the ID of the next event to be generated by the system

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
int get_current_id (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `int`

the event ID

RPC name: inject

Overview:

Injects an artificial event on the given object and returns the corresponding ID in the form of a token, which can be used as a point of reference for database events. For example, to check whether an object has reached the right state before attempting an operation, one can inject an artificial event on the object and wait until the token returned by consecutive event.from calls is lexicographically greater than the one returned by event.inject.

Signature:

```
string inject (session ref session_ref, string class, string ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`class`	class of the object
`string`	`ref`	A reference to the object that will be changed.

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

the event ID in the form of a token

RPC name: next

This message is deprecated.

Overview:

Blocking call which returns a possibly empty batch of events. This method is only recommended for legacy use.It stores events in a buffer of limited size, raising EVENTSLOST if too many events got generated. New development should use event.from which supersedes this method.

Signature:

```
event record set next (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `event record set`

A set of events

Possible Error Codes: `SESSION_NOT_REGISTERED`, `EVENTS_LOST`, `EVENT_SUBSCRIPTION_PARSE_FAILURE`

RPC name: register

This message is deprecated.

Overview:

Registers this session with the event system for a set of given classes. This method is only recommended for legacy use in conjunction with event.next.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void register (session ref session_ref, string set classes)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`classes`	the classes for which the session will register with the event system; specifying * as the desired class will register for all classes

Minimum Role: read-only

Return Type: `void`

RPC name: unregister

This message is deprecated.

Overview:

Removes this session's registration with the event system for a set of given classes. This method is only recommended for legacy use in conjunction with event.next.

Signature:

```
void unregister (session ref session_ref, string set classes)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`classes`	the classes for which the session's registration with the event system will be removed

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Class: Feature

A new piece of functionality

Fields for class: Feature

FIELD	TYPE	QUALIFIER	DESCRIPTION
`enabled`	`bool`	*RO/runtime*	Indicates whether the feature is enabled
`experimental`	`bool`	*RO/constructor*	Indicates whether the feature is experimental (as opposed to stable and fully supported)
`host`	`host ref`	*RO/runtime*	The host where this feature is available
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`version`	`string`	*RO/constructor*	The version of this feature

RPCs associated with class: Feature

RPC name: getall

Overview:

Return a list of all the Features known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Feature ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Feature ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Feature references to Feature records for all Features known to the system.

Signature:

```
(Feature ref -> Feature record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Feature ref -> Feature record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the Feature instances with the given label.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Feature ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `Feature ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the Feature instance with the specified UUID.

Signature:

```
Feature ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Feature ref`

reference to the object

RPC name: getenabled

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the enabled field of the given Feature.

Signature:

```
bool get_enabled (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getexperimental

Overview:

Get the experimental field of the given Feature.

Signature:

```
bool get_experimental (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given Feature.

Signature:

```
host ref get_host (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given Feature.

Signature:

```
string get_name_description (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given Feature.

Signature:

```
string get_name_label (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Feature.

Signature:

```
Feature record get_record (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Feature record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given Feature.

Signature:

```
string get_uuid (session ref session_ref, Feature ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

Overview:

Get the version field of the given Feature.

Signature:

```
string get_version (session ref session_ref, Feature ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Feature ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: GPUgroup

A group of compatible GPUs across the resource pool

Fields for class: GPUgroup

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allocation_algorithm`	`allocation_algorithm`	*RW*	Current allocation of vGPUs to pGPUs for this group
`enabled_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	vGPU types supported on at least one of the pGPUs in this group
`GPU_types`	`string set`	*RO/runtime*	List of GPU types (vendor+device ID) that can be in this group
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PGPUs`	`PGPU ref set`	*RO/runtime*	List of pGPUs in the group
`supported_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	vGPU types supported on at least one of the pGPUs in this group
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VGPUs`	`VGPU ref set`	*RO/runtime*	List of vGPUs using the group

RPCs associated with class: GPUgroup

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given GPUgroup.

Signature:

```
void add_to_other_config (session ref session_ref, GPU_group ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Signature:

```
GPU_group ref create (session ref session_ref, string name_label, string
name_description, (string -> string) map other_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	
`string`	`name_description`	
`(string -> string) map`	`other_config`	

Minimum Role: pool-operator

Return Type: `GPU_group ref`

The reference of the created GPUgroup

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, GPU_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	The GPU group to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the GPUgroups known to the system.

Signature:

```
GPU_group ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `GPU_group ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of GPUgroup references to GPUgroup records for all GPUgroups known to the system.

Signature:

```
(GPU_group ref -> GPU_group record) map get_all_records (session ref
session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(GPU_group ref -> GPU_group record) map`

records of all objects

RPC name: getallocationalgorithm

Overview:

Get the allocationalgorithm field of the given GPUgroup.

Signature:

```
allocation_algorithm get_allocation_algorithm (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `allocation_algorithm`

value of the field

RPC name: getbynamelabel

Overview:

Get all the GPUgroup instances with the given label.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
GPU_group ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `GPU_group ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the GPUgroup instance with the specified UUID.

Signature:

```
GPU_group ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `GPU_group ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getenabledVGPUtypes

Overview:

Get the enabledVGPUtypes field of the given GPUgroup.

Signature:

```
VGPU_type ref set get_enabled_VGPU_types (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getGPUtypes

Overview:

Get the GPUtypes field of the given GPUgroup.

Signature:

```
string set get_GPU_types (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given GPUgroup.

Signature:

```
string get_name_description (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given GPUgroup.

Signature:

```
string get_name_label (session ref session_ref, GPU_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given GPUgroup.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPGPUs

Overview:

Get the PGPUs field of the given GPUgroup.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PGPU ref set get_PGPUs (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given GPUgroup.

Signature:

```
GPU_group record get_record (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group record`

all fields from the object

RPC name: getremainingcapacity

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
int get_remaining_capacity (session ref session_ref, GPU_group ref self,
VGPU_type ref vgpu_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	The GPU group to query
`VGPU_type ref`	`vgpu_type`	The VGPU_type for which the remaining capacity will be calculated

Minimum Role: read-only

Return Type: `int`

The number of VGPUs of the given type which can still be started on the PGPUs in the group

RPC name: getsupportedVGPUtypes

Overview:

Get the supportedVGPUtypes field of the given GPUgroup.

Signature:

```
VGPU_type ref set get_supported_VGPU_types (session ref session_ref,
GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given GPUgroup.

Signature:

```
string get_uuid (session ref session_ref, GPU_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVGPUs

Overview:

Get the VGPUs field of the given GPUgroup.

Signature:

```
VGPU ref set get_VGPUs (session ref session_ref, GPU_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given GPUgroup. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, GPU_group ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setallocationalgorithm

Overview:

Set the allocationalgorithm field of the given GPUgroup.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_allocation_algorithm (session ref session_ref, GPU_group ref
self, allocation_algorithm value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`allocation_algorithm`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given GPUgroup.

Signature:

```
void set_name_description (session ref session_ref, GPU_group ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given GPUgroup.

Signature:

```
void set_name_label (session ref session_ref, GPU_group ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`GPU_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given GPUgroup.

Signature:

```
void set_other_config (session ref session_ref, GPU_group ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`GPU_group ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: host

A physical host

Fields for class: host

FIELD	TYPE	QUALIFIER	DESCRIPTION
`address`	`string`	*RW*	The address by which this host can be contacted from any other host in the pool
`allowed_operations`	`host_allowed_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`API_version_major`	`int`	*RO/runtime*	major version number
`API_version_minor`	`int`	*RO/runtime*	minor version number
`API_version_vendor`	`string`	*RO/runtime*	identification of vendor

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`API_version_vendor_implementation`	`(string -> string) map`	*RO/runtime*	details of vendor implementation
`bios_strings`	`(string -> string) map`	*RO/runtime*	BIOS strings
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this host
`capabilities`	`string set`	*RO/constructor*	Xen capabilities
`certificates`	`Certificate ref set`	*RO/runtime*	List of certificates installed in the host
`chipset_info`	`(string -> string) map`	*RO/runtime*	Information about chipset features
`console_idle_timeout`	`int`	*RO/runtime*	The timeout in seconds after which idle console will be automatically terminated (0 means never)
`control_domain`	`VM ref`	*RO/runtime*	The control domain (domain 0)
`cpu_configuration`	`(string -> string) map`	*RO/runtime*	The CPU configuration on this host. May contain keys such as "nr_nodes", "sockets_per_node", "cores_per_socket", or "threads_per_core"
`cpu_info`	`(string -> string) map`	*RO/runtime*	Details about the physical CPUs on this host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`crash_dump_sr`	`SR ref`	*RW*	The SR in which VDIs for crash dumps are created
`crashdumps`	`host_crashdump ref set`	*RO/runtime*	Set of host crash dumps
`current_operations`	`(string -> host_allowed_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`display`	`host_display`	*RW*	indicates whether the host is configured to output its console to a physical display device
`edition`	`string`	*RO/runtime*	Product edition
`editions`	`string set`	*RO/runtime*	List of all available product editions
`enabled`	`bool`	*RO/runtime*	True if the host is currently enabled
`external_auth_configuration`	`(string -> string) map`	*RO/runtime*	configuration specific to external authentication service
`external_auth_service_name`	`string`	*RO/runtime*	name of external authentication service configured; empty if none configured.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`external_auth_type`	`string`	*RO/runtime*	type of external authentication service configured; empty if none configured.
`features`	`Feature ref set`	*RO/runtime*	List of features available on this host
`guest_VCPUs_params`	`(string -> string) map`	*RW*	VCPUs params to apply to all resident guests
`ha_network_peers`	`string set`	*RO/runtime*	The set of hosts visible via the network from this host
`ha_statefiles`	`string set`	*RO/runtime*	The set of statefiles accessible from this host
`host_CPUs`	`host_cpu ref set`	*RO/runtime*	The physical CPUs on this host
`hostname`	`string`	*RW*	The hostname of this host
`https_only`	`bool`	*RO/runtime*	Reflects whether port 80 is open (false) or not (true)
`iscsi_iqn`	`string`	*RO/constructor*	The initiator IQN for the host
`last_software_update`	`datetime`	*RO/runtime*	Date and time when the last software update was applied
`last_update_hash`	`string`	*RO/runtime*	The SHA256 checksum of updateinfo of the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			most recently applied update on the host
`latest_synced_updates_applied`	`latest_synced_updates_applied_state`	*RO/runtime*	Default as 'unknown', 'yes' if the host is up to date with updates synced from remote CDN, otherwise 'no'
`license_params`	`(string -> string) map`	*RO/runtime*	State of the current license
`license_server`	`(string -> string) map`	*RW*	Contact information of the license server
`local_cache_sr`	`SR ref`	*RO/constructor*	The SR that is used as a local cache
`logging`	`(string -> string) map`	*RW*	logging configuration
`max_cstate`	`string`	*RO/runtime*	The maximum C-state that the host is allowed to enter, "" means unlimited; "N" means limit to CN; "N,M" means limit to CN with max sub cstate M.
`memory_overhead`	`int`	*RO/runtime*	Virtualization memory overhead (bytes).
`metrics`	`host_metrics ref`	*RO/runtime*	metrics associated with this host
`multipathing`	`bool`	*RO/constructor*	Specifies whether multipathing is enabled

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`ntp_custom_servers`	`string set`	*RO/runtime*	Custom NTP servers configured by users, used in Custom NTP mode
`ntp_mode`	`host_ntp_mode`	*RO/runtime*	Indicates NTP servers are assigned by DHCP, or configured by user, or the factory servers, or NTP is disabled
`other_config`	`(string -> string) map`	*RW*	additional configuration
`patches`	`host_patch ref set`	*RO/runtime*	**Deprecated.** Set of host patches
`PBDs`	`PBD ref set`	*RO/runtime*	physical blockdevices
`PCIs`	`PCI ref set`	*RO/runtime*	List of PCI devices in the host
`pending_guidances`	`update_guidances set`	*RO/runtime*	The set of pending mandatory guidances after applying updates, which must be applied, as otherwise there may be for example, VM failures

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`pending_guidances_full`	`update_guidances set`	*RO/runtime*	The set of pending full guidances after applying updates, which a user should follow to make some updates, for example, specific hardware drivers or CPU features, fully effective, but the 'average user' doesn't need to
`pending_guidances_recommended`	`update_guidances set`	*RO/runtime*	The set of pending recommended guidances after applying updates, which most users should follow to make the updates effective, but if not followed, will not cause a failure
`PGPUs`	`PGPU ref set`	*RO/runtime*	List of physical GPUs in the host
`PIFs`	`PIF ref set`	*RO/runtime*	physical network interfaces
`power_on_config`	`(string -> string) map`	*RO/runtime*	The power on config
`power_on_mode`	`string`	*RO/runtime*	The power on mode
`PUSBs`	`PUSB ref set`	*RO/runtime*	List of physical USBs in the host
`resident_VMs`	`VM ref set`	*RO/runtime*	list of VMs currently resident on host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`sched_policy`	`string`	*RO/runtime*	Scheduler policy currently in force on this host
`secure_boot`	`bool`	*RO/runtime*	Whether the host has booted in secure boot mode
`software_version`	`(string -> string) map`	*RO/constructor*	version strings
`ssh_auto_mode`	`bool`	*RO/runtime*	Reflects whether SSH auto mode is enabled for the host
`ssh_enabled`	`bool`	*RO/runtime*	True if SSH access is enabled for the host
`ssh_enabled_timeout`	`int`	*RO/runtime*	The timeout in seconds after which SSH access will be automatically disabled (0 means never), this setting will be applied every time the SSH is enabled by XAPI
`ssh_expiry`	`datetime`	*RO/runtime*	The time in UTC after which the SSH access will be automatically disabled
`ssl_legacy`	`bool`	*RO/constructor*	**Deprecated.** Allow SSLv3 protocol and ciphersuites as used by older server versions. This controls both

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			incoming and outgoing connections. When this is set to a different value, the host immediately restarts its SSL/TLS listening service; typically this takes less than a second but existing connections to it will be broken. API login sessions will remain valid.
`supported_bootloaders`	`string set`	*RO/runtime*	a list of the bootloaders installed on the machine
`suspend_image_sr`	`SR ref`	*RW*	The SR in which VDIs for suspend images are created
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`timezone`	`string`	*RO/runtime*	The time zone identifier as defined in the IANA Time Zone Database
`tls_verification_enabled`	`bool`	*RO/runtime*	True if this host has TLS verifcation enabled
`uefi_certificates`	`string`	*RO/constructor*	**Deprecated.** The UEFI certificates allowing Secure Boot

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`updates`	`pool_update ref set`	*RO/runtime*	Set of updates
`updates_requiring_reboot`	`pool_update ref set`	*RO/runtime*	List of updates which require reboot
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`virtual_hardware_platform_versions`	`int set`	*RO/runtime*	The set of versions of the virtual hardware platform that the host can offer to its guests

RPCs associated with class: host

RPC name: addtags

Overview:

Add the given value to the tags field of the given host. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: addtoguestVCPUsparams

Overview:

Add the given keyvalue pair to the guestVCPUsparams field of the given host.

Signature:

```
void add_to_guest_VCPUs_params (session ref session_ref, host ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtolicenseserver

Overview:

Add the given keyvalue pair to the licenseserver field of the given host.

Signature:

```
void add_to_license_server (session ref session_ref, host ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtologging

Overview:

Add the given keyvalue pair to the logging field of the given host.

Signature:

```
void add_to_logging (session ref session_ref, host ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void add_to_other_config (session ref session_ref, host ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: applyedition

Overview:

Change to another edition, or reactivate the current edition after a license has expired. This may be subject to the successful checkout of an appropriate license.

Signature:

```
void apply_edition (session ref session_ref, host ref host, string
edition, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`edition`	The requested edition

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`force`	Update the license params even if the apply call fails

Minimum Role: pool-operator

Return Type: `void`

RPC name: applyrecommendedguidances

This message is removed.

Overview:

apply all recommended guidances both on the host and on all HVM VMs on the host after updates are applied on the host

Signature:

```
void apply_recommended_guidances (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host whose recommended guidances will be applied

Minimum Role: pool-operator

Return Type: `void`

RPC name: applyupdates

Overview:

apply updates from current enabled repository on a host

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set set apply_updates (session ref session_ref, host ref self,
string hash)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host where updates will be applied
`string`	`hash`	The hash of updateinfo to be applied which is returned by previous pool.sync_udpates

Minimum Role: client-cert

Return Type: `string set set`

The list of results after applying updates, including livepatch apply failures and recommended guidances

RPC name: assertcanevacuate

Overview:

Check this host can be evacuated.

Signature:

```
void assert_can_evacuate (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to evacuate

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: backuprrds

Overview:

This causes the RRDs to be backed up to the master

Signature:

```
void backup_rrds (session ref session_ref, host ref host, float delay)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Schedule a backup of the RRDs of this host
`float`	`delay`	Delay in seconds from when the call is received to perform the backup

Minimum Role: pool-admin

Return Type: `void`

RPC name: bugreportupload

Overview:

Run xenbugtool yestoall and upload the output to support

Signature:

```
void bugreport_upload (session ref session_ref, host ref host, string url,
(string -> string) map options)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host on which to run xen-bugtool
`string`	`url`	The URL to upload to
`(string -> string) map`	`options`	Extra configuration operations

Minimum Role: pool-operator

Return Type: `void`

RPC name: callextension

Overview:

Call an API extension on this host

Signature:

```
string call_extension (session ref session_ref, host ref host, string
call)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`call`	Rpc call for the extension

Minimum Role: pool-admin

Return Type: `string`

Result from the extension

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: callplugin

Overview:

Call an API plugin on this host

Signature:

```
string call_plugin (session ref session_ref, host ref host, string plugin,
string fn, (string -> string) map args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`plugin`	The name of the plugin
`string`	`fn`	The name of the function within the plugin
`(string -> string) map`	`args`	Arguments for the function

Minimum Role: pool-admin

Return Type: `string`

Result from the plugin

RPC name: computefreememory

Overview:

Computes the amount of free memory on the host.

Signature:

```
int compute_free_memory (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to send the request to

Minimum Role: read-only

Return Type: `int`

the amount of free memory on the host.

RPC name: computememoryoverhead

Overview:

Computes the virtualization memory overhead of a host.

Signature:

```
int compute_memory_overhead (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host for which to compute the memory overhead

Minimum Role: read-only

Return Type: `int`

the virtualization memory overhead of the host.

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
blob ref create_new_blob (session ref session_ref, host ref host, string
name, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: declaredead

Overview:

Declare that a host is dead. This is a dangerous operation, and should only be called if the administrator is absolutely sure the host is definitely dead

Signature:

```
void declare_dead (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to declare is dead

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

Overview:

Destroy specified host record in database

Signature:

```
void destroy (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host record to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: disable

Overview:

Puts the host into a state in which no new VMs can be started. Currently active VMs on the host continue to execute.

Signature:

```
void disable (session ref session_ref, host ref host, bool auto_enable)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to disable
`bool`	`auto_enable`	If true (default), the host will be re-enabled after a toolstack restart automatically. If false, the host will be disabled indefinitely, across toolstack restarts and host reboots, until re-enabled explicitly with Host.enable.

Minimum Role: client-cert

Return Type: `void`

RPC name: disabledisplay

Overview:

Disable console output to the physical display device next time this host boots

Signature:

```
host_display disable_display (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `host_display`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

This host's physical display usage

RPC name: disableexternalauth

Overview:

This call disables external authentication on the local host

Signature:

```
void disable_external_auth (session ref session_ref, host ref host,
(string -> string) map config, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose external authentication should be disabled
`(string -> string) map`	`config`	Optional parameters as a list of key-values containing the configuration data
`bool`	`force`	Disable external auth even when not enabled

Minimum Role: pool-admin

Return Type: `void`

RPC name: disablelocalstoragecaching

Overview:

Disable the use of a local SR for caching purposes

Signature:

```
void disable_local_storage_caching (session ref session_ref, host ref
host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `void`

RPC name: disablessh

Overview:

Disable SSH access on the host. It will stop the service sshd only if it is running. It will also disable the service sshd only if it is enabled. A newly joined host in the pool or an ejected host from the pool would keep the original status.

Signature:

```
void disable_ssh (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: dmesg

Overview:

Get the host xen dmesg.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string dmesg (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to query

Minimum Role: pool-operator

Return Type: `string`

dmesg string

RPC name: dmesgclear

Overview:

Get the host xen dmesg, and clear the buffer.

Signature:

```
string dmesg_clear (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to query

Minimum Role: pool-operator

Return Type: `string`

dmesg string

RPC name: emergencyclearmandatoryguidance

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Clear the pending mandatory guidance on this host

Signature:

```
void emergency_clear_mandatory_guidance (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Return Type: `void`

RPC name: emergencydisabletlsverification

Overview:

Disable TLS verification for this host only

Signature:

```
void emergency_disable_tls_verification (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Return Type: `void`

RPC name: emergencyhadisable

Overview:

This call disables HA on the local host. This should only be used with extreme care.

Signature:

```
void emergency_ha_disable (session ref session_ref, bool soft)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`bool`	`soft`	Disable HA temporarily, revert upon host reboot or further changes, idempotent

Minimum Role: pool-operator

Return Type: `void`

RPC name: emergencyreenabletlsverification

Overview:

Reenable TLS verification for this host only, and only after it was emergency disabled

Signature:

```
void emergency_reenable_tls_verification (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Return Type: `void`

RPC name: emergencyresetservercertificate

Overview:

Delete the current TLS server certificate and replace by a new, selfsigned one. This should only be used with extreme care.

Signature:

```
void emergency_reset_server_certificate (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Return Type: `void`

RPC name: enable

Overview:

Puts the host into a state in which new VMs can be started.

Signature:

```
void enable (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to enable

Minimum Role: client-cert

Return Type: `void`

RPC name: enabledisplay

Overview:

Enable console output to the physical display device next time this host boots

Signature:

```
host_display enable_display (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `host_display`

This host's physical display usage

RPC name: enableexternalauth

Overview:

This call enables external authentication on a host

Signature:

```
void enable_external_auth (session ref session_ref, host ref host, (string
-> string) map config, string service_name, string auth_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose external authentication should be enabled
`(string -> string) map`	`config`	A list of key-values containing the configuration data
`string`	`service_name`	The name of the service
`string`	`auth_type`	The type of authentication (for example, AD for Active Directory)

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: enablelocalstoragecaching

Overview:

Enable the use of a local SR for caching purposes

Signature:

```
void enable_local_storage_caching (session ref session_ref, host ref host,
SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`SR ref`	`sr`	The SR to use as a local cache

Minimum Role: pool-operator

Return Type: `void`

RPC name: enablessh

Overview:

Enable SSH access on the host. It will start the service sshd only if it is not running. It will also enable the service sshd only if it is not enabled. A newly joined host in the pool or an ejected host from the pool would keep the original status.

Signature:

```
void enable_ssh (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: evacuate

Overview:

Migrate all VMs off of this host, where possible.

Signature:

```
void evacuate (session ref session_ref, host ref host, network ref
network, int evacuate_batch_size)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to evacuate
`network ref`	`network`	Optional preferred network for migration
`int`	`evacuate_batch_size`	The maximum number of VMs to be migrated per batch 0 will use the value `evacuation-batch-size` defined in xapi.conf

Minimum Role: client-cert

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: forgetdatasourcearchives

Overview:

Forget the recorded statistics related to the specified data source

Signature:

```
void forget_data_source_archives (session ref session_ref, host ref host,
string data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`data_source`	The data source whose archives are to be forgotten

Minimum Role: pool-operator

Return Type: `void`

RPC name: getaddress

Overview:

Get the address field of the given host.

Signature:

```
string get_address (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getall

Overview:

Return a list of all the hosts known to the system.

Signature:

```
host ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of host references to host records for all hosts known to the system.

Signature:

```
(host ref -> host record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(host ref -> host record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given host.

Signature:

```
host_allowed_operations set get_allowed_operations (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_allowed_operations set`

value of the field

RPC name: getAPIversionmajor

Overview:

Get the APIversion/major field of the given host.

Signature:

```
int get_API_version_major (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getAPIversionminor

Overview:

Get the APIversion/minor field of the given host.

Signature:

```
int get_API_version_minor (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getAPIversionvendor

Overview:

Get the APIversion/vendor field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_API_version_vendor (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getAPIversionvendorimplementation

Overview:

Get the APIversion/vendorimplementation field of the given host.

Signature:

```
(string -> string) map get_API_version_vendor_implementation (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbiosstrings

Overview:

Get the biosstrings field of the given host.

Signature:

```
(string -> string) map get_bios_strings (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given host.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getbynamelabel

Overview:

Get all the host instances with the given label.

Signature:

```
host ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `host ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the host instance with the specified UUID.

Signature:

```
host ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host ref`

reference to the object

RPC name: getcapabilities

Overview:

Get the capabilities field of the given host.

Signature:

```
string set get_capabilities (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getcertificates

Overview:

Get the certificates field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Certificate ref set get_certificates (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Certificate ref set`

value of the field

RPC name: getchipsetinfo

Overview:

Get the chipsetinfo field of the given host.

Signature:

```
(string -> string) map get_chipset_info (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getconsoleidletimeout

Overview:

Get the consoleidletimeout field of the given host.

Signature:

```
int get_console_idle_timeout (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getcontroldomain

Overview:

Get the controldomain field of the given host.

Signature:

```
VM ref get_control_domain (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VM ref`

value of the field

RPC name: getcpuconfiguration

Overview:

Get the cpuconfiguration field of the given host.

Signature:

```
(string -> string) map get_cpu_configuration (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcpuinfo

Overview:

Get the cpuinfo field of the given host.

Signature:

```
(string -> string) map get_cpu_info (session ref session_ref, host ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcrashdumpsr

Overview:

Get the crashdumpsr field of the given host.

Signature:

```
SR ref get_crash_dump_sr (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getcrashdumps

Overview:

Get the crashdumps field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host_crashdump ref set get_crashdumps (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_crashdump ref set`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given host.

Signature:

```
(string -> host_allowed_operations) map get_current_operations (session
ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> host_allowed_operations) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getdatasources

Overview:

Signature:

```
data_source record set get_data_sources (session ref session_ref, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to interrogate

Minimum Role: read-only

Return Type: `data_source record set`

A set of data sources

RPC name: getdisplay

Overview:

Get the display field of the given host.

Signature:

```
host_display get_display (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `host_display`

value of the field

RPC name: getedition

Overview:

Get the edition field of the given host.

Signature:

```
string get_edition (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: geteditions

Overview:

Get the editions field of the given host.

Signature:

```
string set get_editions (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getenabled

Overview:

Get the enabled field of the given host.

Signature:

```
bool get_enabled (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getexternalauthconfiguration

Overview:

Get the externalauthconfiguration field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_external_auth_configuration (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getexternalauthservicename

Overview:

Get the externalauthservicename field of the given host.

Signature:

```
string get_external_auth_service_name (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getexternalauthtype

Overview:

Get the externalauthtype field of the given host.

Signature:

```
string get_external_auth_type (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfeatures

Overview:

Get the features field of the given host.

Signature:

```
Feature ref set get_features (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `Feature ref set`

value of the field

RPC name: getguestVCPUsparams

Overview:

Get the guestVCPUsparams field of the given host.

Signature:

```
(string -> string) map get_guest_VCPUs_params (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethanetworkpeers

Overview:

Get the hanetworkpeers field of the given host.

Signature:

```
string set get_ha_network_peers (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gethastatefiles

Overview:

Get the hastatefiles field of the given host.

Signature:

```
string set get_ha_statefiles (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gethostCPUs

Overview:

Get the hostCPUs field of the given host.

Signature:

```
host_cpu ref set get_host_CPUs (session ref session_ref, host ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_cpu ref set`

value of the field

RPC name: gethostname

Overview:

Get the hostname field of the given host.

Signature:

```
string get_hostname (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethttpsonly

Overview:

Get the httpsonly field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_https_only (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getiscsiiqn

Overview:

Get the iscsiiqn field of the given host.

Signature:

```
string get_iscsi_iqn (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getlastsoftwareupdate

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the lastsoftwareupdate field of the given host.

Signature:

```
datetime get_last_software_update (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlastupdatehash

Overview:

Get the lastupdatehash field of the given host.

Signature:

```
string get_last_update_hash (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getlatestsyncedupdatesapplied

Overview:

Get the latestsyncedupdatesapplied field of the given host.

Signature:

```
latest_synced_updates_applied_state get_latest_synced_updates_applied
(session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `latest_synced_updates_applied_state`

value of the field

RPC name: getlicenseparams

Overview:

Get the licenseparams field of the given host.

Signature:

```
(string -> string) map get_license_params (session ref session_ref, host
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlicenseserver

Overview:

Get the licenseserver field of the given host.

Signature:

```
(string -> string) map get_license_server (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlocalcachesr

Overview:

Get the localcachesr field of the given host.

Signature:

```
SR ref get_local_cache_sr (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getlog

Overview:

Get the host's log file

Signature:

```
string get_log (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to query

Minimum Role: read-only

Return Type: `string`

The contents of the host's primary log file

RPC name: getlogging

Overview:

Get the logging field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_logging (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getmanagementinterface

Overview:

Returns the management interface for the specified host

Signature:

```
PIF ref get_management_interface (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Which host's management interface is required

Minimum Role: pool-operator

Return Type: `PIF ref`

The management interface for the host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getmaxcstate

Overview:

Get the maxcstate field of the given host.

Signature:

```
string get_max_cstate (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getmemoryoverhead

Overview:

Get the memory/overhead field of the given host.

Signature:

```
int get_memory_overhead (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getmetrics

Overview:

Get the metrics field of the given host.

Signature:

```
host_metrics ref get_metrics (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_metrics ref`

value of the field

RPC name: getmultipathing

Overview:

Get the multipathing field of the given host.

Signature:

```
bool get_multipathing (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given host.

Signature:

```
string get_name_description (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given host.

Signature:

```
string get_name_label (session ref session_ref, host ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getntpcustomservers

Overview:

Get the ntpcustomservers field of the given host.

Signature:

```
string set get_ntp_custom_servers (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getntpmode

Overview:

Get the ntpmode field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host_ntp_mode get_ntp_mode (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_ntp_mode`

value of the field

RPC name: getntpserversstatus

Overview:

Get the NTP servers status on the host

Signature:

```
(string -> string) map get_ntp_servers_status (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `(string -> string) map`

The map of NTP server to its status, status may be synced/combined/uncombined/error/variable/unreachable/unknown

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getntpsynchronized

Overview:

Returns true if the system clock on the host is synchronized with the NTP servers.

Signature:

```
bool get_ntp_synchronized (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `bool`

true if the system clock on the host is synchronized with the NTP servers.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given host.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpatches

This message is deprecated.

Overview:

Get the patches field of the given host.

Signature:

```
host_patch ref set get_patches (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_patch ref set`

value of the field

RPC name: getPBDs

Overview:

Get the PBDs field of the given host.

Signature:

```
PBD ref set get_PBDs (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PBD ref set`

value of the field

RPC name: getPCIs

Overview:

Get the PCIs field of the given host.

Signature:

```
PCI ref set get_PCIs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref set`

value of the field

RPC name: getpendingguidances

Overview:

Get the pendingguidances field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
update_guidances set get_pending_guidances (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getpendingguidancesfull

Overview:

Get the pendingguidancesfull field of the given host.

Signature:

```
update_guidances set get_pending_guidances_full (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpendingguidancesrecommended

Overview:

Get the pendingguidancesrecommended field of the given host.

Signature:

```
update_guidances set get_pending_guidances_recommended (session ref
session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getPGPUs

Overview:

Get the PGPUs field of the given host.

Signature:

```
PGPU ref set get_PGPUs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getPIFs

Overview:

Get the PIFs field of the given host.

Signature:

```
PIF ref set get_PIFs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref set`

value of the field

RPC name: getpoweronconfig

Overview:

Get the poweronconfig field of the given host.

Signature:

```
(string -> string) map get_power_on_config (session ref session_ref, host
ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpoweronmode

Overview:

Get the poweronmode field of the given host.

Signature:

```
string get_power_on_mode (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPUSBs

Overview:

Get the PUSBs field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PUSB ref set get_PUSBs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PUSB ref set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given host.

Signature:

```
host record get_record (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host record`

all fields from the object

RPC name: getresidentVMs

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the residentVMs field of the given host.

Signature:

```
VM ref set get_resident_VMs (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: getschedgran

Overview:

Gets xen's schedgran on a host

Signature:

```
host_sched_gran get_sched_gran (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Return Type: `host_sched_gran`

The host's schedgran

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getschedpolicy

Overview:

Get the schedpolicy field of the given host.

Signature:

```
string get_sched_policy (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsecureboot

Overview:

Get the secureboot field of the given host.

Signature:

```
bool get_secure_boot (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getservercertificate

Overview:

Get the installed server public TLS certificate.

Signature:

```
string get_server_certificate (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: read-only

Return Type: `string`

The installed server public TLS certificate, in PEM form.

RPC name: getserverlocaltime

Overview:

This call queries the host's clock for the current time in the host's local timezone

Signature:

```
datetime get_server_localtime (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`host`	The host whose clock should be queried

Minimum Role: read-only

Return Type: `datetime`

The current local time

RPC name: getservertime

Overview:

This call queries the host's clock for the current time

Signature:

```
datetime get_servertime (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose clock should be queried

Minimum Role: read-only

Return Type: `datetime`

The current time

RPC name: getsoftwareversion

Overview:

Get the softwareversion field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_software_version (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsshautomode

Overview:

Get the sshautomode field of the given host.

Signature:

```
bool get_ssh_auto_mode (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsshenabled

Overview:

Get the sshenabled field of the given host.

Signature:

```
bool get_ssh_enabled (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsshenabledtimeout

Overview:

Get the sshenabledtimeout field of the given host.

Signature:

```
int get_ssh_enabled_timeout (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getsshexpiry

Overview:

Get the sshexpiry field of the given host.

Signature:

```
datetime get_ssh_expiry (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getssllegacy

This message is deprecated.

Overview:

Get the ssllegacy field of the given host.

Signature:

```
bool get_ssl_legacy (session ref session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsupportedbootloaders

Overview:

Get the supportedbootloaders field of the given host.

Signature:

```
string set get_supported_bootloaders (session ref session_ref, host ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getsuspendimagesr

Overview:

Get the suspendimagesr field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR ref get_suspend_image_sr (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getsystemstatuscapabilities

Overview:

Signature:

```
string get_system_status_capabilities (session ref session_ref, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to interrogate

Minimum Role: read-only

Return Type: `string`

An XML fragment containing the system status capabilities.

RPC name: gettags

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the tags field of the given host.

Signature:

```
string set get_tags (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettimezone

Overview:

Get the timezone field of the given host.

Signature:

```
string get_timezone (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettlsverificationenabled

Overview:

Get the tlsverificationenabled field of the given host.

Signature:

```
bool get_tls_verification_enabled (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gettrackeduseragents

Overview:

Get the name, version list of tracked user agents on this host. If different versions of the same name are seen, keep the lastseen version. The oldest entry will be removed if reach the max num. Note that the list is cleared after host/XAPI restart

Signature:

```
(string -> string) map get_tracked_user_agents (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `(string -> string) map`

The name, version list of user agents that have been tracked on this host

RPC name: getueficertificates

This message is deprecated.

Overview:

Get the ueficertificates field of the given host.

Signature:

```
string get_uefi_certificates (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuncooperativeresidentVMs

This message is deprecated.

Overview:

Return a set of VMs which are not cooperating with the host's memory control system

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM ref set get_uncooperative_resident_VMs (session ref session_ref, host
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host to query

Minimum Role: read-only

Return Type: `VM ref set`

VMs which are not cooperating

RPC name: getupdates

Overview:

Get the updates field of the given host.

Signature:

```
pool_update ref set get_updates (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getupdatesrequiringreboot

Overview:

Get the updatesrequiringreboot field of the given host.

Signature:

```
pool_update ref set get_updates_requiring_reboot (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given host.

Signature:

```
string get_uuid (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvirtualhardwareplatformversions

Overview:

Get the virtualhardwareplatformversions field of the given host.

Signature:

```
int set get_virtual_hardware_platform_versions (session ref session_ref,
host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int set`

value of the field

RPC name: getvmswhichpreventevacuation

Overview:

Return a set of VMs which prevent the host being evacuated, with perVM error codes

Signature:

```
(VM ref -> string set) map get_vms_which_prevent_evacuation (session ref
session_ref, host ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host to query

Minimum Role: read-only

Return Type: `(VM ref -> string set) map`

VMs which block evacuation together with reasons

RPC name: hasextension

Overview:

Return true if the extension is available on the host

Signature:

```
bool has_extension (session ref session_ref, host ref host, string name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`name`	The name of the API call

Minimum Role: pool-admin

Return Type: `bool`

True if the extension exists, false otherwise

RPC name: installservercertificate

Overview:

Install the TLS server certificate.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void install_server_certificate (session ref session_ref, host ref host,
string certificate, string private_key, string certificate_chain)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`certificate`	The server certificate, in PEM form
`string`	`private_key`	The unencrypted private key used to sign the certificate, in PKCS#8 form
`string`	`certificate_chain`	The certificate chain, in PEM form

Minimum Role: pool-admin

Return Type: `void`

RPC name: licenseadd

Overview:

Apply a new license to a host

Signature:

```
void license_add (session ref session_ref, host ref host, string contents)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`host`	The host to upload the license to
`string`	`contents`	The contents of the license file, base64 encoded

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `LICENSE_PROCESSING_ERROR`

RPC name: licenseapply

This message is removed.

Overview:

Apply a new license to a host

Signature:

```
void license_apply (session ref session_ref, host ref host, string
contents)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to upload the license to
`string`	`contents`	The contents of the license file, base64 encoded

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `LICENSE_PROCESSING_ERROR`

RPC name: licenseremove

Overview:

Remove any license file from the specified host, and switch that host to the unlicensed edition

Signature:

```
void license_remove (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host from which any license will be removed

Minimum Role: pool-operator

Return Type: `void`

RPC name: listmethods

Overview:

List all supported methods

Signature:

```
string set list_methods (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string set`

The name of every supported method.

RPC name: listtimezones

Overview:

List all available timezones on the host.

Signature:

```
string set list_timezones (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

Minimum Role: read-only

Return Type: `string set`

The set of available timezones on the host

RPC name: localmanagementreconfigure

Overview:

Reconfigure the management network interface. Should only be used if Host.managementreconfigure is impossible because the network configuration is broken.

Signature:

```
void local_management_reconfigure (session ref session_ref, string
interface)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`interface`	name of the interface to use as a management interface

Minimum Role: pool-operator

Return Type: `void`

RPC name: managementdisable

Overview:

Disable the management network interface

Signature:

```
void management_disable (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: managementreconfigure

Overview:

Reconfigure the management network interface

Signature:

```
void management_reconfigure (session ref session_ref, PIF ref pif)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`pif`	reference to a PIF object corresponding to the management interface

Minimum Role: pool-operator

Return Type: `void`

RPC name: migratereceive

Overview:

Prepare to receive a VM, returning a token which can be passed to VM.migrate.

Signature:

```
(string -> string) map migrate_receive (session ref session_ref, host ref host, network ref network, (string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The target host
`network ref`	`network`	The network through which migration traffic should be received.
`(string -> string) map`	`options`	Extra configuration operations

Minimum Role: vm-power-admin

Return Type: `(string -> string) map`

A value which should be passed to VM.migrate

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: poweron

Overview:

Attempt to poweron the host if the capability exists.

Signature:

```
void power_on (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to power on

Minimum Role: pool-operator

Return Type: `void`

RPC name: querydatasource

Overview:

Query the latest value of the specified data source

Signature:

```
float query_data_source (session ref session_ref, host ref host, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`data_source`	The data source to query

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `float`

The latest value, averaged over the last 5 seconds

RPC name: reboot

Overview:

Reboot the host. This function can only be called if there are no currently running VMs on the host and it is disabled.

Signature:

```
void reboot (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to reboot

Minimum Role: pool-operator

Return Type: `void`

RPC name: recorddatasource

Overview:

Start recording the specified data source

Signature:

```
void record_data_source (session ref session_ref, host ref host, string
data_source)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`data_source`	The data source to record

Minimum Role: pool-operator

Return Type: `void`

RPC name: refreshpackinfo

This message is deprecated.

Overview:

Refresh the list of installed Supplemental Packs.

Signature:

```
void refresh_pack_info (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host to modify

Minimum Role: pool-operator

Return Type: `void`

RPC name: refreshservercertificate

Overview:

Replace the internal selfsigned host certficate with a new one.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void refresh_server_certificate (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: removefromguestVCPUsparams

Overview:

Remove the given key and its corresponding value from the guestVCPUsparams field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_guest_VCPUs_params (session ref session_ref, host ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromlicenseserver

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the licenseserver field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_license_server (session ref session_ref, host ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromlogging

Overview:

Remove the given key and its corresponding value from the logging field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_logging (session ref session_ref, host ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given host. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given host. If the value is not in that Set, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_tags (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: rescandrivers

Overview:

Scan the host and update its driver information.

Signature:

```
void rescan_drivers (session ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host to be rescanned

Minimum Role: pool-admin

Return Type: `void`

RPC name: resetcpufeatures

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the feature mask, such that after a reboot all features of the CPU are enabled.

Signature:

```
void reset_cpu_features (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-operator

Return Type: `void`

RPC name: resetservercertificate

Overview:

Delete the current TLS server certificate and replace by a new, selfsigned one. This should only be used with extreme care.

Signature:

```
void reset_server_certificate (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: restartagent

Overview:

Restarts the agent after a 10 second pause. WARNING: this is a dangerous operation. Any operations in progress will be aborted, and unrecoverable data loss may occur. The caller is responsible for ensuring that there are no operations in progress when this method is called.

Signature:

```
void restart_agent (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The Host on which you want to restart the agent

Minimum Role: pool-operator

Return Type: `void`

RPC name: retrievewlbevacuaterecommendations

Overview:

Retrieves recommended host migrations to perform when evacuating the host from the wlb server. If a VM cannot be migrated from the host the reason is listed instead of a recommendation.

Signature:

```
(VM ref -> string set) map retrieve_wlb_evacuate_recommendations (session
ref session_ref, host ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	The host to query

Minimum Role: read-only

Return Type: `(VM ref -> string set) map`

VMs and the reasons why they would block evacuation, or their target host recommended by the wlb server

RPC name: senddebugkeys

Overview:

Inject the given string as debugging keys into Xen

Signature:

```
void send_debug_keys (session ref session_ref, host ref host, string keys)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`keys`	The keys to send

Minimum Role: pool-admin

Return Type: `void`

RPC name: setaddress

Overview:

Set the address field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_address (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setconsoleidletimeout

Overview:

Set the console idle timeout for the host

Signature:

```
void set_console_idle_timeout (session ref session_ref, host ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`int`	`value`	The console idle timeout in seconds

Minimum Role: pool-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setcpufeatures

This message is removed.

Overview:

Set the CPU features to be used after a reboot, if the given features string is valid.

Signature:

```
void set_cpu_features (session ref session_ref, host ref host, string
features)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`features`	The features string (32 hexadecimal digits)

Minimum Role: pool-operator

Return Type: `void`

RPC name: setcrashdumpsr

Overview:

Set the crashdumpsr field of the given host.

Signature:

```
void set_crash_dump_sr (session ref session_ref, host ref self, SR ref
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setdisplay

Overview:

Set the display field of the given host.

Signature:

```
void set_display (session ref session_ref, host ref self, host_display
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`host_display`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setguestVCPUsparams

Overview:

Set the guestVCPUsparams field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_guest_VCPUs_params (session ref session_ref, host ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethostname

Overview:

Set the hostname field of the given host.

Signature:

```
void set_hostname (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: sethostnamelive

Overview:

Sets the host name to the specified string. Both the API and lowerlevel system hostname are changed immediately.

Signature:

```
void set_hostname_live (session ref session_ref, host ref host, string hostname)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host whose host name to set
`string`	`hostname`	The new host name

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `HOST_NAME_INVALID`

RPC name: sethttpsonly

Overview:

updates the host firewall to open or close port 80 depending on the value

Signature:

```
void set_https_only (session ref session_ref, host ref self, bool value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The Host
`bool`	`value`	true - http port 80 will be blocked, false - http port 80 will be open

Minimum Role: pool-operator

Return Type: `void`

RPC name: setiscsiiqn

Overview:

Sets the initiator IQN for the host

Signature:

```
void set_iscsi_iqn (session ref session_ref, host ref host, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`value`	The value to which the IQN should be set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlicenseserver

Overview:

Set the licenseserver field of the given host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_license_server (session ref session_ref, host ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlogging

Overview:

Set the logging field of the given host.

Signature:

```
void set_logging (session ref session_ref, host ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setmaxcstate

Overview:

Sets xen's maxcstate on a host. See: https://xenbits.xen.org/docs/unstable/misc/xencommandline.htmlmaxcstatex86. "" means unlimited; "N" means limit to CN; "N,M" means limit to CN with max sub cstate M. Note: Only C0, C1, unlimited are supported currently.

Signature:

```
void set_max_cstate (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`string`	`value`	The max_cstate to apply to a host

Minimum Role: pool-operator

Return Type: `void`

RPC name: setmultipathing

Overview:

Specifies whether multipathing is enabled

Signature:

```
void set_multipathing (session ref session_ref, host ref host, bool value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`bool`	`value`	Whether multipathing should be enabled

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given host.

Signature:

```
void set_name_description (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_name_label (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setntpcustomservers

Overview:

Set the custom NTP servers for the host

Signature:

```
void set_ntp_custom_servers (session ref session_ref, host ref self,
string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`string set`	`value`	The set of custom NTP servers to configure

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setntpmode

Overview:

Set the NTP mode for the host

Signature:

```
void set_ntp_mode (session ref session_ref, host ref self, host_ntp_mode
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`host_ntp_mode`	`value`	The NTP mode to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given host.

Signature:

```
void set_other_config (session ref session_ref, host ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setpoweronmode

Overview:

Set the poweronmode, host, user and password

Signature:

```
void set_power_on_mode (session ref session_ref, host ref self, string
power_on_mode, (string -> string) map power_on_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`string`	`power_on_mode`	power-on-mode can be empty, wake-on-lan, IPMI or other
`(string -> string) map`	`power_on_config`	Power on config

Minimum Role: pool-operator

Return Type: `void`

RPC name: setschedgran

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Sets xen's schedgran on a host. See: https://xenbits.xen.org/docs/unstable/misc/xencommandline.htmlschedgranx86

Signature:

```
void set_sched_gran (session ref session_ref, host ref self,
host_sched_gran value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`host_sched_gran`	`value`	The sched-gran to apply to a host

Return Type: `void`

RPC name: setservertime

Overview:

Set the host's system clock when NTP is disabled.

Signature:

```
void set_servertime (session ref session_ref, host ref self, datetime
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`datetime`	`value`	A date/time to be set. When a timezone offset is missing, UTC is assumed.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsshautomode

Overview:

Set the SSH auto mode for the host

Signature:

```
void set_ssh_auto_mode (session ref session_ref, host ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`bool`	`value`	The SSH auto mode for the host，when set to true, SSH to normally be disabled and SSH to be enabled only in case of emergency for example, xapi is down

Minimum Role: pool-admin

Return Type: `void`

RPC name: setsshenabledtimeout

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the SSH service enabled timeout for the host

Signature:

```
void set_ssh_enabled_timeout (session ref session_ref, host ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`int`	`value`	The SSH enabled timeout in seconds (0 means no timeout, max 2 days)

Minimum Role: pool-admin

Return Type: `void`

RPC name: setssllegacy

Overview:

Enable/disable SSLv3 for interoperability with older server versions. When this is set to a different value, the host immediately restarts its SSL/TLS listening service; typically this takes less than a second but existing connections to it will be broken. API login sessions will remain valid.

Signature:

```
void set_ssl_legacy (session ref session_ref, host ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	True to allow SSLv3 and ciphersuites as used in old XenServer versions

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsuspendimagesr

Overview:

Set the suspendimagesr field of the given host.

Signature:

```
void set_suspend_image_sr (session ref session_ref, host ref self, SR ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given host.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_tags (session ref session_ref, host ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: settimezone

Overview:

Set the host's timezone.

Signature:

```
void set_timezone (session ref session_ref, host ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`self`	The host
`string`	`value`	The time zone identifier as defined in the IANA Time Zone Database

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setueficertificates

This message is deprecated.

Overview:

Sets the UEFI certificates on a host

Signature:

```
void set_uefi_certificates (session ref session_ref, host ref host, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host
`string`	`value`	The certificates to apply to a host

Return Type: `void`

RPC name: shutdown

Overview:

Shutdown the host. This function can only be called if there are no currently running VMs on the host and it is disabled.

Signature:

```
void shutdown (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host ref`	`host`	The Host to shutdown

Minimum Role: pool-operator

Return Type: `void`

RPC name: shutdownagent

Overview:

Shuts the agent down after a 10 second pause. WARNING: this is a dangerous operation. Any operations in progress will be aborted, and unrecoverable data loss may occur. The caller is responsible for ensuring that there are no operations in progress when this method is called.

Signature:

```
void shutdown_agent (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: syncdata

Overview:

This causes the synchronisation of the nondatabase data messages, RRDs and so on stored on the master to be synchronised with the host

Signature:

```
void sync_data (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to whom the data should be sent

Minimum Role: pool-admin

Return Type: `void`

RPC name: syslogreconfigure

Overview:

Reconfigure syslog logging

Signature:

```
void syslog_reconfigure (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Tell the host to reread its Host.logging parameters and reconfigure itself accordingly

Minimum Role: pool-operator

Return Type: `void`

Class: hostcpu

This class is deprecated.

A physical CPU

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: hostcpu

FIELD	TYPE	QUALIFIER	DESCRIPTION
`family`	`int`	*RO/runtime*	**Deprecated.** the family (number) of the physical CPU
`features`	`string`	*RO/runtime*	**Deprecated.** the physical CPU feature bitmap
`flags`	`string`	*RO/runtime*	**Deprecated.** the flags of the physical CPU (a decoded version of the features field)
`host`	`host ref`	*RO/runtime*	**Deprecated.** the host the CPU is in
`model`	`int`	*RO/runtime*	**Deprecated.** the model number of the physical CPU
`modelname`	`string`	*RO/runtime*	**Deprecated.** the model name of the physical CPU
`number`	`int`	*RO/runtime*	**Deprecated.** the number of the physical CPU within the host
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`speed`	`int`	*RO/runtime*	**Deprecated.** the speed of the physical CPU

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`stepping`	`string`	*RO/runtime*	**Deprecated.** the stepping of the physical CPU
`utilisation`	`float`	*RO/runtime*	**Deprecated.** the current CPU utilisation
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference
`vendor`	`string`	*RO/runtime*	**Deprecated.** the vendor of the physical CPU

RPCs associated with class: hostcpu

RPC name: addtootherconfig

This message is deprecated.

Overview:

Add the given keyvalue pair to the otherconfig field of the given hostcpu.

Signature:

```
void add_to_other_config (session ref session_ref, host_cpu ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the hostcpus known to the system.

Signature:

```
host_cpu ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host_cpu ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

Return a map of hostcpu references to hostcpu records for all hostcpus known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(host_cpu ref -> host_cpu record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(host_cpu ref -> host_cpu record) map`

records of all objects

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the hostcpu instance with the specified UUID.

Signature:

```
host_cpu ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host_cpu ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getfamily

This message is deprecated.

Overview:

Get the family field of the given hostcpu.

Signature:

```
int get_family (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getfeatures

This message is deprecated.

Overview:

Get the features field of the given hostcpu.

Signature:

```
string get_features (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getflags

This message is deprecated.

Overview:

Get the flags field of the given hostcpu.

Signature:

```
string get_flags (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

This message is deprecated.

Overview:

Get the host field of the given hostcpu.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref get_host (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getmodel

This message is deprecated.

Overview:

Get the model field of the given hostcpu.

Signature:

```
int get_model (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getmodelname

This message is deprecated.

Overview:

Get the modelname field of the given hostcpu.

Signature:

```
string get_modelname (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnumber

This message is deprecated.

Overview:

Get the number field of the given hostcpu.

Signature:

```
int get_number (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given hostcpu.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given hostcpu.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
host_cpu record get_record (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_cpu record`

all fields from the object

RPC name: getspeed

This message is deprecated.

Overview:

Get the speed field of the given hostcpu.

Signature:

```
int get_speed (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getstepping

This message is deprecated.

Overview:

Get the stepping field of the given hostcpu.

Signature:

```
string get_stepping (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getutilisation

This message is deprecated.

Overview:

Get the utilisation field of the given hostcpu.

Signature:

```
float get_utilisation (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given hostcpu.

Signature:

```
string get_uuid (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendor

This message is deprecated.

Overview:

Get the vendor field of the given hostcpu.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_vendor (session ref session_ref, host_cpu ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given hostcpu. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host_cpu ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given hostcpu.

Signature:

```
void set_other_config (session ref session_ref, host_cpu ref self, (string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_cpu ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: hostcrashdump

Represents a host crash dump

Fields for class: hostcrashdump

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	Host the crashdump relates to
`other_config`	`(string -> string) map`	*RW*	additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`size`	`int`	*RO/runtime*	Size of the crashdump
`timestamp`	`datetime`	*RO/runtime*	Time the crash happened
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: hostcrashdump

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given hostcrashdump.

Signature:

```
void add_to_other_config (session ref session_ref, host_crashdump ref
self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Destroy specified host crash dump, removing it from the disk.

Signature:

```
void destroy (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	The host crashdump to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the hostcrashdumps known to the system.

Signature:

```
host_crashdump ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host_crashdump ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of hostcrashdump references to hostcrashdump records for all hostcrashdumps known to the system.

Signature:

```
(host_crashdump ref -> host_crashdump record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(host_crashdump ref -> host_crashdump record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the hostcrashdump instance with the specified UUID.

Signature:

```
host_crashdump ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `host_crashdump ref`

reference to the object

RPC name: gethost

Overview:

Get the host field of the given hostcrashdump.

Signature:

```
host ref get_host (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given hostcrashdump.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given hostcrashdump.

Signature:

```
host_crashdump record get_record (session ref session_ref, host_crashdump
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_crashdump record`

all fields from the object

RPC name: getsize

Overview:

Get the size field of the given hostcrashdump.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int get_size (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gettimestamp

Overview:

Get the timestamp field of the given hostcrashdump.

Signature:

```
datetime get_timestamp (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the uuid field of the given hostcrashdump.

Signature:

```
string get_uuid (session ref session_ref, host_crashdump ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given hostcrashdump. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host_crashdump ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object
`string`	`key`	Key to remove

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given hostcrashdump.

Signature:

```
void set_other_config (session ref session_ref, host_crashdump ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: upload

Overview:

Upload the specified host crash dump to a specified URL

Signature:

```
void upload (session ref session_ref, host_crashdump ref self, string url,
(string -> string) map options)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_crashdump ref`	`self`	The host crashdump to upload
`string`	`url`	The URL to upload to
`(string -> string) map`	`options`	Extra configuration operations

Minimum Role: pool-operator

Return Type: `void`

Class: Hostdriver

UNSUPPORTED. A multiversion driver on a host

Fields for class: Hostdriver

FIELD	TYPE	QUALIFIER	DESCRIPTION
`active_variant`	`Driver_variant ref`	*RO/runtime*	Currently active variant of this driver, if any, or Null.
`description`	`string`	*RO/runtime*	Description of the driver
`friendly_name`	`string`	*RO/runtime*	Descriptive name, not used for identification
`host`	`host ref`	*RO/constructor*	Host where this driver is installed
`info`	`string`	*RO/runtime*	Information about the driver
`name`	`string`	*RO/runtime*	Name identifying the driver uniquely

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`selected_variant`	`Driver_variant ref`	*RO/runtime*	Variant (if any) selected to become active after reboot. Or Null
`type`	`string`	*RO/runtime*	Device type this driver supports, like network or storage
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`variants`	`Driver_variant ref set`	*RO/runtime*	Variants of this driver available for selection

RPCs associated with class: Hostdriver

RPC name: deselect

Overview:

UNSUPPORTED. Deselect the currently active variant of this driver after reboot. No action will be taken if no variant is currently active.

Signature:

```
void deselect (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	Driver to become inactive (after reboot).

Minimum Role: pool-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getactivevariant

Overview:

Get the activevariant field of the given Hostdriver.

Signature:

```
Driver_variant ref get_active_variant (session ref session_ref,
Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant ref`

value of the field

RPC name: getall

Overview:

Return a list of all the Hostdrivers known to the system.

Signature:

```
Host_driver ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `Host_driver ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Hostdriver references to Hostdriver records for all Hostdrivers known to the system.

Signature:

```
(Host_driver ref -> Host_driver record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Host_driver ref -> Host_driver record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the Hostdriver instance with the specified UUID.

Signature:

```
Host_driver ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Host_driver ref`

reference to the object

RPC name: getdescription

Overview:

Get the description field of the given Hostdriver.

Signature:

```
string get_description (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfriendlyname

Overview:

Get the friendlyname field of the given Hostdriver.

Signature:

```
string get_friendly_name (session ref session_ref, Host_driver ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given Hostdriver.

Signature:

```
host ref get_host (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getinfo

Overview:

Get the info field of the given Hostdriver.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_info (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getname

Overview:

Get the name field of the given Hostdriver.

Signature:

```
string get_name (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a record containing the current state of the given Hostdriver.

Signature:

```
Host_driver record get_record (session ref session_ref, Host_driver ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Host_driver record`

all fields from the object

RPC name: getselectedvariant

Overview:

Get the selectedvariant field of the given Hostdriver.

Signature:

```
Driver_variant ref get_selected_variant (session ref session_ref,
Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: gettype

Overview:

Get the type field of the given Hostdriver.

Signature:

```
string get_type (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given Hostdriver.

Signature:

```
string get_uuid (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvariants

Overview:

Get the variants field of the given Hostdriver.

Signature:

```
Driver_variant ref set get_variants (session ref session_ref, Host_driver ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Driver_variant ref set`

value of the field

RPC name: rescan

Overview:

UNSUPPORTED. Rescan a host's drivers and update information about them. This is mostly for trouble shooting.

Signature:

```
void rescan (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	Update driver information of this host.

Minimum Role: pool-admin

Return Type: `void`

RPC name: select

Overview:

UNSUPPORTED. Select a variant of the driver to become active after reboot or immediately if currently no version is active

Signature:

```
void select (session ref session_ref, Host_driver ref self, Driver_variant ref variant)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Host_driver ref`	`self`	Driver to become active (after reboot).
`Driver_variant ref`	`variant`	Driver version to become active (after reboot).

Minimum Role: pool-admin

Return Type: `void`

Class: hostmetrics

The metrics associated with a host

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: hostmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated
`live`	`bool`	*RO/runtime*	Pool master thinks this host is live
`memory_free`	`int`	*RO/runtime*	**Removed.** Free host memory (bytes)
`memory_total`	`int`	*RO/runtime*	Total host memory (bytes)
`other_config`	`(string -> string) map`	*RW*	additional configuration
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: hostmetrics

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given hostmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, host_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the hostmetrics instances known to the system.

Signature:

```
host_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host_metrics ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of hostmetrics references to hostmetrics records for all hostmetrics instances known to the system.

Signature:

```
(host_metrics ref -> host_metrics record) map get_all_records (session ref
session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(host_metrics ref -> host_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the hostmetrics instance with the specified UUID.

Signature:

```
host_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host_metrics ref`

reference to the object

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given hostmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
datetime get_last_updated (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlive

Overview:

Get the live field of the given hostmetrics.

Signature:

```
bool get_live (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmemoryfree

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Get the memory/free field of the given hostmetrics.

Signature:

```
int get_memory_free (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorytotal

Overview:

Get the memory/total field of the given hostmetrics.

Signature:

```
int get_memory_total (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given hostmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given hostmetrics.

Signature:

```
host_metrics record get_record (session ref session_ref, host_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_metrics record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given hostmetrics.

Signature:

```
string get_uuid (session ref session_ref, host_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given hostmetrics. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_other_config (session ref session_ref, host_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given hostmetrics.

Signature:

```
void set_other_config (session ref session_ref, host_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: hostpatch

This class is deprecated.

Represents a patch stored on a server

Fields for class: hostpatch

FIELD	TYPE	QUALIFIER	DESCRIPTION
`applied`	`bool`	*RO/runtime*	**Deprecated.** True if the patch has been applied
`host`	`host ref`	*RO/constructor*	**Deprecated.** Host the patch relates to
`name_description`	`string`	*RO/constructor*	**Deprecated.** a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	**Deprecated.** a human-readable name
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`pool_patch`	`pool_patch ref`	*RO/constructor*	**Deprecated.** The patch applied
`size`	`int`	*RO/runtime*	**Deprecated.** Size of the patch
`timestamp_applied`	`datetime`	*RO/runtime*	**Deprecated.** Time the patch was applied
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`version`	`string`	*RO/constructor*	**Deprecated.** Patch version number

RPCs associated with class: hostpatch

RPC name: addtootherconfig

This message is deprecated.

Overview:

Add the given keyvalue pair to the otherconfig field of the given hostpatch.

Signature:

```
void add_to_other_config (session ref session_ref, host_patch ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: apply

This message is deprecated.

Overview:

Apply the selected patch and return its output

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string apply (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	The patch to apply

Minimum Role: pool-operator

Return Type: `string`

the output of the patch application process

RPC name: destroy

This message is deprecated.

Overview:

Destroy the specified host patch, removing it from the disk. This does NOT reverse the patch

Signature:

```
void destroy (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	The patch to destroy

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the hostpatchs known to the system.

Signature:

```
host_patch ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `host_patch ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

Return a map of hostpatch references to hostpatch records for all hostpatchs known to the system.

Signature:

```
(host_patch ref -> host_patch record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(host_patch ref -> host_patch record) map`

records of all objects

RPC name: getapplied

This message is deprecated.

Overview:

Get the applied field of the given hostpatch.

Signature:

```
bool get_applied (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getbynamelabel

This message is deprecated.

Overview:

Get all the hostpatch instances with the given label.

Signature:

```
host_patch ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `host_patch ref set`

references to objects with matching names

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the hostpatch instance with the specified UUID.

Signature:

```
host_patch ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `host_patch ref`

reference to the object

RPC name: gethost

This message is deprecated.

Overview:

Get the host field of the given hostpatch.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
host ref get_host (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getnamedescription

This message is deprecated.

Overview:

Get the name/description field of the given hostpatch.

Signature:

```
string get_name_description (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnamelabel

This message is deprecated.

Overview:

Get the name/label field of the given hostpatch.

Signature:

```
string get_name_label (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given hostpatch.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpoolpatch

This message is deprecated.

Overview:

Get the poolpatch field of the given hostpatch.

Signature:

```
pool_patch ref get_pool_patch (session ref session_ref, host_patch ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_patch ref`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given hostpatch.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
host_patch record get_record (session ref session_ref, host_patch ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_patch record`

all fields from the object

RPC name: getsize

This message is deprecated.

Overview:

Get the size field of the given hostpatch.

Signature:

```
int get_size (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: gettimestampapplied

This message is deprecated.

Overview:

Get the timestampapplied field of the given hostpatch.

Signature:

```
datetime get_timestamp_applied (session ref session_ref, host_patch ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given hostpatch.

Signature:

```
string get_uuid (session ref session_ref, host_patch ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

This message is deprecated.

Overview:

Get the version field of the given hostpatch.

Signature:

```
string get_version (session ref session_ref, host_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given hostpatch. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, host_patch ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given hostpatch.

Signature:

```
void set_other_config (session ref session_ref, host_patch ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host_patch ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: LVHD

LVHD SR specific operations

Fields for class: LVHD

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: LVHD

RPC name: enablethinprovisioning

Overview:

Upgrades an LVHD SR to enable thinprovisioning. Future VDIs created in this SR will be thinlyprovisioned, although existing VDIs will be left alone. Note that the SR must be attached to the SRmaster for upgrade to work.

Signature:

```
string enable_thin_provisioning (session ref session_ref, host ref host,
SR ref SR, int initial_allocation, int allocation_quantum)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The LVHD Host to upgrade to being thin-provisioned.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`SR`	The LVHD SR to upgrade to being thin-provisioned.
`int`	`initial_allocation`	The initial amount of space to allocate to a newly-created VDI in bytes
`int`	`allocation_quantum`	The amount of space to allocate to a VDI when it needs to be enlarged in bytes

Minimum Role: pool-admin

Return Type: `string`

Message from LVHD.enablethinprovisioning extension

RPC name: getbyuuid

Overview:

Get a reference to the LVHD instance with the specified UUID.

Signature:

```
LVHD ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `LVHD ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

Overview:

Get a record containing the current state of the given LVHD.

Signature:

```
LVHD record get_record (session ref session_ref, LVHD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`LVHD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `LVHD record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given LVHD.

Signature:

```
string get_uuid (session ref session_ref, LVHD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`LVHD ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

Class: message

An message for the attention of the administrator

Fields for class: message

FIELD	TYPE	QUALIFIER	DESCRIPTION
`body`	`string`	*RO/runtime*	The body of the message
`cls`	`cls`	*RO/runtime*	The class of the object this message is associated with
`name`	`string`	*RO/runtime*	The name of the message
`obj_uuid`	`string`	*RO/runtime*	The uuid of the object this message is associated with
`priority`	`int`	*RO/runtime*	The message priority, 0 being low priority
`timestamp`	`datetime`	*RO/runtime*	The time at which the message was created
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: message

RPC name: create

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
message ref create (session ref session_ref, string name, int priority,
cls cls, string obj_uuid, string body)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	The name of the message
`int`	`priority`	The priority of the message
`cls`	`cls`	The class of object this message is associated with
`string`	`obj_uuid`	The uuid of the object this message is associated with
`string`	`body`	The body of the message

Minimum Role: pool-operator

Return Type: `message ref`

The reference of the created message

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, message ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`message ref`	`self`	The reference of the message to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroymany

Overview:

Signature:

```
void destroy_many (session ref session_ref, message ref set messages)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`message ref set`	`messages`	Messages to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: get

Overview:

Signature:

```
(message ref -> message record) map get (session ref session_ref, cls cls,
string obj_uuid, datetime since)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`cls`	`cls`	The class of object
`string`	`obj_uuid`	The uuid of the object
`datetime`	`since`	The cutoff time. When the timezone is missing, UTC is assumed

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The relevant messages

RPC name: getall

Overview:

Signature:

```
message ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `message ref set`

The references to the messages

RPC name: getallrecords

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(message ref -> message record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The messages

RPC name: getallrecordswhere

Overview:

Signature:

```
(message ref -> message record) map get_all_records_where (session ref
session_ref, string expr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`expr`	The expression to match

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The messages

RPC name: getbyuuid

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
message ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	The uuid of the message

Minimum Role: read-only

Return Type: `message ref`

The message reference

RPC name: getrecord

Overview:

Signature:

```
message record get_record (session ref session_ref, message ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`message ref`	`self`	The reference to the message

Minimum Role: read-only

Return Type: `message record`

The message record

RPC name: getsince

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(message ref -> message record) map get_since (session ref session_ref,
datetime since)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`datetime`	`since`	The cutoff time. When the timezone is missing, UTC is assumed

Minimum Role: read-only

Return Type: `(message ref -> message record) map`

The relevant messages

Class: network

A virtual network

Fields for class: network

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`network_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`assigned_ips`	`(VIF ref -> string) map`	*RO/runtime*	The IP addresses assigned to VIFs on networks that have

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			active xapi-managed DHCP
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this network
`bridge`	`string`	*RO/constructor*	name of the bridge corresponding to this network on the local host
`current_operations`	`(string -> network_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`default_locking_mode`	`network_default_locking_mode`	*RO/runtime*	The network will use this value to determine the behaviour of all VIFs where locking_mode = default
`managed`	`bool`	*RO/constructor*	true if the bridge is managed by xapi
`MTU`	`int`	*RW*	MTU in octets
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	additional configuration
`PIFs`	`PIF ref set`	*RO/runtime*	list of connected pifs
`purpose`	`network_purpose set`	*RO/runtime*	Set of purposes for which the server will use this network
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VIFs`	`VIF ref set`	*RO/runtime*	list of connected vifs

RPCs associated with class: network

RPC name: addpurpose

Overview:

Give a network a new purpose if not present already

Signature:

```
void add_purpose (session ref session_ref, network ref self,
network_purpose value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	The network
`network_purpose`	`value`	The purpose to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

Possible Error Codes: `NETWORK_INCOMPATIBLE_PURPOSES`

RPC name: addtags

Overview:

Add the given value to the tags field of the given network. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, network ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given network.

Signature:

```
void add_to_other_config (session ref session_ref, network ref self,
string key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new network instance, and return its handle.

Signature:

```
network ref create (session ref session_ref, network record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `network ref`

reference to the newly created object

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
blob ref create_new_blob (session ref session_ref, network ref network,
string name, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	The network
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: destroy

Overview:

Destroy the specified network instance.

Signature:

```
void destroy (session ref session_ref, network ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the networks known to the system.

Signature:

```
network ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `network ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of network references to network records for all networks known to the system.

Signature:

```
(network ref -> network record) map get_all_records (session ref
session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(network ref -> network record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given network.

Signature:

```
network_operations set get_allowed_operations (session ref session_ref,
network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_operations set`

value of the field

RPC name: getassignedips

Overview:

Get the assignedips field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(VIF ref -> string) map get_assigned_ips (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(VIF ref -> string) map`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given network.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbridge

Overview:

Get the bridge field of the given network.

Signature:

```
string get_bridge (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbynamelabel

Overview:

Get all the network instances with the given label.

Signature:

```
network ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `network ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the network instance with the specified UUID.

Signature:

```
network ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `network ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given network.

Signature:

```
(string -> network_operations) map get_current_operations (session ref
session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> network_operations) map`

value of the field

RPC name: getdefaultlockingmode

Overview:

Get the defaultlockingmode field of the given network.

Signature:

```
network_default_locking_mode get_default_locking_mode (session ref
session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_default_locking_mode`

value of the field

RPC name: getmanaged

Overview:

Get the managed field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_managed (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getMTU

Overview:

Get the MTU field of the given network.

Signature:

```
int get_MTU (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnamedescription

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the name/description field of the given network.

Signature:

```
string get_name_description (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given network.

Signature:

```
string get_name_label (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given network.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPIFs

Overview:

Get the PIFs field of the given network.

Signature:

```
PIF ref set get_PIFs (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `PIF ref set`

value of the field

RPC name: getpurpose

Overview:

Get the purpose field of the given network.

Signature:

```
network_purpose set get_purpose (session ref session_ref, network ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_purpose set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given network.

Signature:

```
network record get_record (session ref session_ref, network ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network record`

all fields from the object

RPC name: gettags

Overview:

Get the tags field of the given network.

Signature:

```
string set get_tags (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVIFs

Overview:

Get the VIFs field of the given network.

Signature:

```
VIF ref set get_VIFs (session ref session_ref, network ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF ref set`

value of the field

RPC name: removefromotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given network. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, network ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removepurpose

Overview:

Remove a purpose from a network if present

Signature:

```
void remove_purpose (session ref session_ref, network ref self,
network_purpose value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	The network
`network_purpose`	`value`	The purpose to remove

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given network. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, network ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: setdefaultlockingmode

Overview:

Set the default locking mode for VIFs attached to this network

Signature:

```
void set_default_locking_mode (session ref session_ref, network ref
network, network_default_locking_mode value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	The network
`network_default_locking_mode`	`value`	The default locking mode for VIFs attached to this network.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setMTU

Overview:

Set the MTU field of the given network.

Signature:

```
void set_MTU (session ref session_ref, network ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given network.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_name_description (session ref session_ref, network ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given network.

Signature:

```
void set_name_label (session ref session_ref, network ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given network.

Signature:

```
void set_other_config (session ref session_ref, network ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given network.

Signature:

```
void set_tags (session ref session_ref, network ref self, string set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

Class: networksriov

networksriov which connects logical pif and physical pif

Fields for class: networksriov

FIELD	TYPE	QUALIFIER	DESCRIPTION
`configuration_mode`	`sriov_configuration_mode`	*RO/runtime*	The mode for configure network sriov
`logical_PIF`	`PIF ref`	*RO/constructor*	The logical PIF to connect to the SR-IOV network after enable SR-IOV on the physical PIF
`physical_PIF`	`PIF ref`	*RO/constructor*	The PIF that has SR-IOV enabled
`requires_reboot`	`bool`	*RO/runtime*	Indicates whether the host need to be rebooted before SR-IOV is enabled on the physical PIF
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: networksriov

RPC name: create

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable SRIOV on the specific PIF. It will create a networksriov based on the specific PIF and automatically create a logical PIF to connect the specific network.

Signature:

```
network_sriov ref create (session ref session_ref, PIF ref pif, network ref network)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`pif`	PIF on which to enable SR-IOV
`network ref`	`network`	Network to connect SR-IOV virtual functions with VM VIFs

Minimum Role: pool-operator

Return Type: `network_sriov ref`

The reference of the created networksriov object

RPC name: destroy

Overview:

Disable SRIOV on the specific PIF. It will destroy the networksriov and the logical PIF accordingly.

Signature:

```
void destroy (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`network_sriov ref`	`self`	SRIOV to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the networksriovs known to the system.

Signature:

```
network_sriov ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `network_sriov ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of networksriov references to networksriov records for all networksriovs known to the system.

Signature:

```
(network_sriov ref -> network_sriov record) map get_all_records (session
ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(network_sriov ref -> network_sriov record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the networksriov instance with the specified UUID.

Signature:

```
network_sriov ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `network_sriov ref`

reference to the object

RPC name: getconfigurationmode

Overview:

Get the configurationmode field of the given networksriov.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
sriov_configuration_mode get_configuration_mode (session ref session_ref,
network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `sriov_configuration_mode`

value of the field

RPC name: getlogicalPIF

Overview:

Get the logicalPIF field of the given networksriov.

Signature:

```
PIF ref get_logical_PIF (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getphysicalPIF

Overview:

Get the physicalPIF field of the given networksriov.

Signature:

```
PIF ref get_physical_PIF (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given networksriov.

Signature:

```
network_sriov record get_record (session ref session_ref, network_sriov
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `network_sriov record`

all fields from the object

RPC name: getremainingcapacity

Overview:

Get the number of free SRIOV VFs on the associated PIF

Signature:

```
int get_remaining_capacity (session ref session_ref, network_sriov ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	the NETWORK_SRIOV object

Minimum Role: read-only

Return Type: `int`

The number of free SRIOV VFs on the associated PIF

RPC name: getrequiresreboot

Overview:

Get the requiresreboot field of the given networksriov.

Signature:

```
bool get_requires_reboot (session ref session_ref, network_sriov ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given networksriov.

Signature:

```
string get_uuid (session ref session_ref, network_sriov ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network_sriov ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: Observer

Describes an observer which will control observability activity in the Toolstack

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: Observer

FIELD	TYPE	QUALIFIER	DESCRIPTION
`attributes`	`(string -> string) map`	*RO/constructor*	Attributes that observer will add to the data they produce
`components`	`string set`	*RO/constructor*	The list of xenserver components the observer will broadcast. An empty list means all components
`enabled`	`bool`	*RO/constructor*	This denotes if the observer is enabled. true if it is enabled and false if it is disabled
`endpoints`	`string set`	*RO/constructor*	The list of endpoints where data is exported to. Each endpoint is a URL or the string 'bugtool' refering to the internal logs
`hosts`	`host ref set`	*RO/constructor*	The list of hosts the observer is active on. An empty list means all hosts
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: Observer

RPC name: create

Overview:

Create a new Observer instance, and return its handle.

Signature:

```
Observer ref create (session ref session_ref, Observer record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer record`	`args`	All constructor arguments

Minimum Role: pool-admin

Return Type: `Observer ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified Observer instance.

Signature:

```
void destroy (session ref session_ref, Observer ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the Observers known to the system.

Signature:

```
Observer ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `Observer ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Observer references to Observer records for all Observers known to the system.

Signature:

```
(Observer ref -> Observer record) map get_all_records (session ref
session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Observer ref -> Observer record) map`

records of all objects

RPC name: getattributes

Overview:

Get the attributes field of the given Observer.

Signature:

```
(string -> string) map get_attributes (session ref session_ref, Observer
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getbynamelabel

Overview:

Get all the Observer instances with the given label.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Observer ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `Observer ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the Observer instance with the specified UUID.

Signature:

```
Observer ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Observer ref`

reference to the object

RPC name: getcomponents

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the components field of the given Observer.

Signature:

```
string set get_components (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getenabled

Overview:

Get the enabled field of the given Observer.

Signature:

```
bool get_enabled (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getendpoints

Overview:

Get the endpoints field of the given Observer.

Signature:

```
string set get_endpoints (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gethosts

Overview:

Get the hosts field of the given Observer.

Signature:

```
host ref set get_hosts (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `host ref set`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given Observer.

Signature:

```
string get_name_description (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given Observer.

Signature:

```
string get_name_label (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Observer.

Signature:

```
Observer record get_record (session ref session_ref, Observer ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Observer record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given Observer.

Signature:

```
string get_uuid (session ref session_ref, Observer ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: setattributes

Overview:

Set the attributes of an observer. These are used to emit metadata by the observer

Signature:

```
void set_attributes (session ref session_ref, Observer ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`(string -> string) map`	`value`	The attributes that the observer emits as part of the data

Minimum Role: pool-admin

Return Type: `void`

RPC name: setcomponents

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the components on which the observer will broadcast to. i.e. xapi, xenopsd, networkd, etc.

Signature:

```
void set_components (session ref session_ref, Observer ref self, string
set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`string set`	`value`	The components the observer will broadcast to

Minimum Role: pool-admin

Return Type: `void`

RPC name: setenabled

Overview:

Enable / disable this observer which will stop the observer from producing observability information

Signature:

```
void set_enabled (session ref session_ref, Observer ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	If the observer is to be enabled (true) or disabled (false)

Minimum Role: pool-admin

Return Type: `void`

RPC name: setendpoints

Overview:

Set the file/HTTP endpoints the observer sends data to

Signature:

```
void set_endpoints (session ref session_ref, Observer ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`string set`	`value`	The endpoints that the observer will export data to. A URL or the string 'bugtool'. This can refer to an enpoint to the local file system

Minimum Role: pool-admin

Return Type: `void`

RPC name: sethosts

Overview:

Sets the hosts that the observer is to be registered on

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_hosts (session ref session_ref, Observer ref self, host ref set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	The observer
`host ref set`	`value`	Hosts the observer is registered on

Minimum Role: pool-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given Observer.

Signature:

```
void set_name_description (session ref session_ref, Observer ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given Observer.

Signature:

```
void set_name_label (session ref session_ref, Observer ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Observer ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

Class: PBD

The physical block devices through which hosts access SRs

Fields for class: PBD

FIELD	TYPE	QUALIFIER	DESCRIPTION
`currently_attached`	`bool`	*RO/runtime*	is the SR currently attached on this host?
`device_config`	`(string -> string) map`	*RO/constructor*	a config string to string map that is provided to the

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			host's SR-backend-driver
`host`	`host ref`	*RO/constructor*	physical machine on which the pbd is available
`other_config`	`(string -> string) map`	*RW*	additional configuration
`SR`	`SR ref`	*RO/constructor*	the storage repository that the pbd realises
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PBD

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PBD.

Signature:

```
void add_to_other_config (session ref session_ref, PBD ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new PBD instance, and return its handle.

Signature:

```
PBD ref create (session ref session_ref, PBD record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `PBD ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified PBD instance.

Signature:

```
void destroy (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PBD ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PBDs known to the system.

Signature:

```
PBD ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PBD ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PBD references to PBD records for all PBDs known to the system.

Signature:

```
(PBD ref -> PBD record) map get_all_records (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PBD ref -> PBD record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PBD instance with the specified UUID.

Signature:

```
PBD ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PBD ref`

reference to the object

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given PBD.

Signature:

```
bool get_currently_attached (session ref session_ref, PBD ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdeviceconfig

Overview:

Get the deviceconfig field of the given PBD.

Signature:

```
(string -> string) map get_device_config (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethost

Overview:

Get the host field of the given PBD.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
host ref get_host (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PBD.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PBD ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

Overview:

Get a record containing the current state of the given PBD.

Signature:

```
PBD record get_record (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PBD record`

all fields from the object

RPC name: getSR

Overview:

Get the SR field of the given PBD.

Signature:

```
SR ref get_SR (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `SR ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PBD.

Signature:

```
string get_uuid (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: plug

Overview:

Activate the specified PBD, causing the referenced SR to be attached and scanned

Signature:

```
void plug (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PBD ref`	`self`	The PBD to activate

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `SR_UNKNOWN_DRIVER`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PBD. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PBD ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setdeviceconfig

Overview:

Sets the PBD's deviceconfig field

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_device_config (session ref session_ref, PBD ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	The PBD to modify
`(string -> string) map`	`value`	The new value of the PBD's device_config

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PBD.

Signature:

```
void set_other_config (session ref session_ref, PBD ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: unplug

Overview:

Deactivate the specified PBD, causing the referenced SR to be detached and nolonger scanned

Signature:

```
void unplug (session ref session_ref, PBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PBD ref`	`self`	The PBD to deactivate

Minimum Role: pool-operator

Return Type: `void`

Class: PCI

A PCI device

Fields for class: PCI

FIELD	TYPE	QUALIFIER	DESCRIPTION
`class_id`	`string`	*RO/constructor*	PCI class ID
`class_name`	`string`	*RO/constructor*	PCI class name
`dependencies`	`PCI ref set`	*RO/runtime*	List of dependent PCI devices
`device_id`	`string`	*RO/constructor*	Device ID
`device_name`	`string`	*RO/constructor*	Device name
`driver_name`	`string`	*RO/constructor*	Driver name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	Physical machine that owns the PCI device
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`pci_id`	`string`	*RO/constructor*	PCI ID of the physical device
`subsystem_device_id`	`string`	*RO/constructor*	Subsystem device ID
`subsystem_device_name`	`string`	*RO/constructor*	Subsystem device name
`subsystem_vendor_id`	`string`	*RO/constructor*	Subsystem vendor ID
`subsystem_vendor_name`	`string`	*RO/constructor*	Subsystem vendor name
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_id`	`string`	*RO/constructor*	Vendor ID
`vendor_name`	`string`	*RO/constructor*	Vendor name

RPCs associated with class: PCI

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PCI.

Signature:

```
void add_to_other_config (session ref session_ref, PCI ref self, string key, string value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: disabledom0access

Overview:

Hide a PCI device from the dom0 kernel. Takes affect after next boot.

Signature:

```
pci_dom0_access disable_dom0_access (session ref session_ref, PCI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	The PCI to hide

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PCI from dom0

RPC name: enabledom0access

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Unhide a PCI device from the dom0 kernel. Takes affect after next boot.

Signature:

```
pci_dom0_access enable_dom0_access (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	The PCI to unhide

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PCI from dom0

RPC name: getall

Overview:

Return a list of all the PCIs known to the system.

Signature:

```
PCI ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PCI ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of PCI references to PCI records for all PCIs known to the system.

Signature:

```
(PCI ref -> PCI record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PCI ref -> PCI record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PCI instance with the specified UUID.

Signature:

```
PCI ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PCI ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the object

RPC name: getclassid

Overview:

Get the classid field of the given PCI.

Signature:

```
string get_class_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getclassname

Overview:

Get the classname field of the given PCI.

Signature:

```
string get_class_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdependencies

Overview:

Get the dependencies field of the given PCI.

Signature:

```
PCI ref set get_dependencies (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref set`

value of the field

RPC name: getdeviceid

Overview:

Get the deviceid field of the given PCI.

Signature:

```
string get_device_id (session ref session_ref, PCI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdevicename

Overview:

Get the devicename field of the given PCI.

Signature:

```
string get_device_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdom0accessstatus

Overview:

Return a PCI device dom0 access status.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
pci_dom0_access get_dom0_access_status (session ref session_ref, PCI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	The PCI

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PCI from dom0

RPC name: getdrivername

Overview:

Get the drivername field of the given PCI.

Signature:

```
string get_driver_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given PCI.

Signature:

```
host ref get_host (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PCI.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PCI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpciid

Overview:

Get the pciid field of the given PCI.

Signature:

```
string get_pci_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PCI.

Signature:

```
PCI record get_record (session ref session_ref, PCI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI record`

all fields from the object

RPC name: getsubsystemdeviceid

Overview:

Get the subsystemdeviceid field of the given PCI.

Signature:

```
string get_subsystem_device_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsubsystemdevicename

Overview:

Get the subsystemdevicename field of the given PCI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_subsystem_device_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsubsystemvendorid

Overview:

Get the subsystemvendorid field of the given PCI.

Signature:

```
string get_subsystem_vendor_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsubsystemvendorname

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the subsystemvendorname field of the given PCI.

Signature:

```
string get_subsystem_vendor_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PCI.

Signature:

```
string get_uuid (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getvendorid

Overview:

Get the vendorid field of the given PCI.

Signature:

```
string get_vendor_id (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorname

Overview:

Get the vendorname field of the given PCI.

Signature:

```
string get_vendor_name (session ref session_ref, PCI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PCI. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PCI ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PCI.

Signature:

```
void set_other_config (session ref session_ref, PCI ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PCI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: PGPU

A physical GPU pGPU

Fields for class: PGPU

FIELD	TYPE	QUALIFIER	DESCRIPTION
`compatibility_metadata`	`(string -> string) map`	*RO/runtime*	PGPU metadata to determine whether a VGPU can migrate between two PGPUs
`dom0_access`	`pci_dom0_access`	*RO/runtime*	**Deprecated.** The accessibility of this device from dom0
`enabled_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	List of VGPU types which have been enabled for this PGPU
`GPU_group`	`GPU_group ref`	*RO/constructor*	GPU group the pGPU is contained in
`host`	`host ref`	*RO/runtime*	Host that owns the GPU
`is_system_display_device`	`bool`	*RO/runtime*	Is this device the system display device

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PCI`	`PCI ref`	*RO/constructor*	Link to underlying PCI device
`resident_VGPUs`	`VGPU ref set`	*RO/runtime*	List of VGPUs running on this PGPU
`supported_VGPU_max_capacities`	`(VGPU_type ref -> int) map`	*RO/runtime*	A map relating each VGPU type supported on this GPU to the maximum number of VGPUs of that type which can run simultaneously on this GPU
`supported_VGPU_types`	`VGPU_type ref set`	*RO/runtime*	List of VGPU types supported by the underlying hardware
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PGPU

RPC name: addenabledVGPUtypes

Overview:

Signature:

```
void add_enabled_VGPU_types (session ref session_ref, PGPU ref self,
VGPU_type ref value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to which we are adding an enabled VGPU type
`VGPU_type ref`	`value`	The VGPU type to enable

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PGPU.

Signature:

```
void add_to_other_config (session ref session_ref, PGPU ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: disabledom0access

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Signature:

```
pci_dom0_access disable_dom0_access (session ref session_ref, PGPU ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to which dom0 will be denied access

Minimum Role: pool-operator

Return Type: `pci_dom0_access`

The accessibility of this PGPU from dom0

RPC name: enabledom0access

This message is deprecated.

Overview:

Signature:

```
pci_dom0_access enable_dom0_access (session ref session_ref, PGPU ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to which dom0 will be granted access

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `pci_dom0_access`

The accessibility of this PGPU from dom0

RPC name: getall

Overview:

Return a list of all the PGPUs known to the system.

Signature:

```
PGPU ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PGPU ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PGPU references to PGPU records for all PGPUs known to the system.

Signature:

```
(PGPU ref -> PGPU record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(PGPU ref -> PGPU record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PGPU instance with the specified UUID.

Signature:

```
PGPU ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PGPU ref`

reference to the object

RPC name: getcompatibilitymetadata

Overview:

Get the compatibilitymetadata field of the given PGPU.

Signature:

```
(string -> string) map get_compatibility_metadata (session ref
session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getdom0access

This message is deprecated.

Overview:

Get the dom0access field of the given PGPU.

Signature:

```
pci_dom0_access get_dom0_access (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pci_dom0_access`

value of the field

RPC name: getenabledVGPUtypes

Overview:

Get the enabledVGPUtypes field of the given PGPU.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VGPU_type ref set get_enabled_VGPU_types (session ref session_ref, PGPU
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getGPUgroup

Overview:

Get the GPUgroup field of the given PGPU.

Signature:

```
GPU_group ref get_GPU_group (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethost

Overview:

Get the host field of the given PGPU.

Signature:

```
host ref get_host (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getissystemdisplaydevice

Overview:

Get the issystemdisplaydevice field of the given PGPU.

Signature:

```
bool get_is_system_display_device (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PGPU.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPCI

Overview:

Get the PCI field of the given PGPU.

Signature:

```
PCI ref get_PCI (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PGPU.

Signature:

```
PGPU record get_record (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU record`

all fields from the object

RPC name: getremainingcapacity

Overview:

Signature:

```
int get_remaining_capacity (session ref session_ref, PGPU ref self,
VGPU_type ref vgpu_type)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to query
`VGPU_type ref`	`vgpu_type`	The VGPU type for which we want to find the number of VGPUs which can still be started on this PGPU

Minimum Role: read-only

Return Type: `int`

The number of VGPUs of the specified type which can still be started on this PGPU

RPC name: getresidentVGPUs

Overview:

Get the residentVGPUs field of the given PGPU.

Signature:

```
VGPU ref set get_resident_VGPUs (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsupportedVGPUmaxcapacities

Overview:

Get the supportedVGPUmaxcapacities field of the given PGPU.

Signature:

```
(VGPU_type ref -> int) map get_supported_VGPU_max_capacities (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(VGPU_type ref -> int) map`

value of the field

RPC name: getsupportedVGPUtypes

Overview:

Get the supportedVGPUtypes field of the given PGPU.

Signature:

```
VGPU_type ref set get_supported_VGPU_types (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PGPU.

Signature:

```
string get_uuid (session ref session_ref, PGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removeenabledVGPUtypes

Overview:

Signature:

```
void remove_enabled_VGPU_types (session ref session_ref, PGPU ref self,
VGPU_type ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PGPU ref`	`self`	The PGPU from which we are removing an enabled VGPU type
`VGPU_type ref`	`value`	The VGPU type to disable

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PGPU. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PGPU ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setenabledVGPUtypes

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_enabled_VGPU_types (session ref session_ref, PGPU ref self,
VGPU_type ref set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU on which we are enabling a set of VGPU types
`VGPU_type ref set`	`value`	The VGPU types to enable

Minimum Role: pool-operator

Return Type: `void`

RPC name: setGPUgroup

Overview:

Signature:

```
void set_GPU_group (session ref session_ref, PGPU ref self, GPU_group ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	The PGPU to move to a new group
`GPU_group ref`	`value`	The group to which the PGPU will be moved

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PGPU.

Signature:

```
void set_other_config (session ref session_ref, PGPU ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PGPU ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: PIF

A physical network interface note separate VLANs are represented as several PIFs

Fields for class: PIF

FIELD	TYPE	QUALIFIER	DESCRIPTION
`bond_master_of`	`Bond ref set`	*RO/runtime*	Indicates this PIF represents the results of a bond
`bond_slave_of`	`Bond ref`	*RO/runtime*	Indicates which bond this interface is part of

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`capabilities`	`string set`	*RO/runtime*	Additional capabilities on the interface.
`currently_attached`	`bool`	*RO/runtime*	true if this interface is online
`device`	`string`	*RO/constructor*	machine-readable name of the interface (for example, eth0)
`disallow_unplug`	`bool`	*RO/runtime*	Prevent this PIF from being unplugged; set this to notify the management tool-stack that the PIF has a special use and should not be unplugged under any circumstances (for example, because you're running storage traffic over it)
`DNS`	`string`	*RO/runtime*	Comma separated list of the IP addresses of the DNS servers to use
`gateway`	`string`	*RO/runtime*	IP gateway
`host`	`host ref`	*RO/constructor*	physical machine to which this pif is connected
`igmp_snooping_status`	`pif_igmp_status`	*RO/runtime*	The IGMP snooping status of the corresponding network bridge
`IP`	`string`	*RO/runtime*	IP address

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`ip_configuration_mode`	`ip_configuration_mode`	*RO/runtime*	Sets if and how this interface gets an IP address
`IPv6`	`string set`	*RO/runtime*	IPv6 address
`ipv6_configuration_mode`	`ipv6_configuration_mode`	*RO/runtime*	Sets if and how this interface gets an IPv6 address
`ipv6_gateway`	`string`	*RO/runtime*	IPv6 gateway
`MAC`	`string`	*RO/constructor*	ethernet MAC address of physical interface
`managed`	`bool`	*RO/constructor*	Indicates whether the interface is managed by xapi. If it is not, then xapi will not configure the interface, the commands PIF.plug/unplug/reconfigure_ip(v6) cannot be used, nor can the interface be bonded or have VLANs based on top through xapi.
`management`	`bool`	*RO/runtime*	Indicates whether the control software is listening for connections on this interface
`metrics`	`PIF_metrics ref`	*RO/runtime*	metrics associated with this PIF
`MTU`	`int`	*RO/constructor*	MTU in octets

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`netmask`	`string`	*RO/runtime*	IP netmask
`network`	`network ref`	*RO/constructor*	virtual network to which this pif is connected
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PCI`	`PCI ref`	*RO/runtime*	Link to underlying PCI device
`physical`	`bool`	*RO/runtime*	true if this represents a physical network interface
`primary_address_type`	`primary_address_type`	*RO/runtime*	Which protocol should define the primary address of this interface
`properties`	`(string -> string) map`	*RO/runtime*	Additional configuration properties for the interface.
`sriov_logical_PIF_of`	`network_sriov ref set`	*RO/runtime*	Indicates which network_sriov this interface is logical of
`sriov_physical_PIF_of`	`network_sriov ref set`	*RO/runtime*	Indicates which network_sriov this interface is physical of
`tunnel_access_PIF_of`	`tunnel ref set`	*RO/runtime*	Indicates to which tunnel this PIF gives access

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`tunnel_transport_PIF_of`	`tunnel ref set`	*RO/runtime*	Indicates to which tunnel this PIF provides transport
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VLAN`	`int`	*RO/constructor*	VLAN tag for all traffic passing through this interface
`VLAN_master_of`	`VLAN ref`	*RO/runtime*	Indicates which VLAN this interface receives untagged traffic from
`VLAN_slave_of`	`VLAN ref set`	*RO/runtime*	Indicates which VLANs this interface transmits tagged traffic to

RPCs associated with class: PIF

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PIF.

Signature:

```
void add_to_other_config (session ref session_ref, PIF ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: createVLAN

This message is deprecated.

Overview:

Create a VLAN interface from an existing physical interface. This call is deprecated: use VLAN.create instead

Signature:

```
PIF ref create_VLAN (session ref session_ref, string device, network ref
network, host ref host, int VLAN)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`device`	physical interface on which to create the VLAN interface
`network ref`	`network`	network to which this interface should be connected
`host ref`	`host`	physical machine to which this PIF is connected
`int`	`VLAN`	VLAN tag for the new interface

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `PIF ref`

The reference of the created PIF object

Possible Error Codes: `VLAN_TAG_INVALID`

RPC name: dbforget

Overview:

Destroy a PIF database record.

Signature:

```
void db_forget (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	The ref of the PIF whose database record should be destroyed

Minimum Role: pool-operator

Return Type: `void`

RPC name: dbintroduce

Overview:

Create a new PIF record in the database only

Signature:

```
PIF ref db_introduce (session ref session_ref, string device, network ref
network, host ref host, string MAC, int MTU, int VLAN, bool physical,
ip_configuration_mode ip_configuration_mode, string IP, string netmask,
string gateway, string DNS, Bond ref bond_slave_of, VLAN ref
VLAN_master_of, bool management, (string -> string) map other_config, bool
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
disallow_unplug, ipv6_configuration_mode ipv6_configuration_mode, string
set IPv6, string ipv6_gateway, primary_address_type primary_address_type,
bool managed, (string -> string) map properties)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`device`	
`network ref`	`network`	
`host ref`	`host`	
`string`	`MAC`	
`int`	`MTU`	
`int`	`VLAN`	
`bool`	`physical`	
`ip_configuration_mode`	`ip_configuration_mode`	
`string`	`IP`	
`string`	`netmask`	
`string`	`gateway`	
`string`	`DNS`	
`Bond ref`	`bond_slave_of`	
`VLAN ref`	`VLAN_master_of`	
`bool`	`management`	
`(string -> string) map`	`other_config`	
`bool`	`disallow_unplug`	

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`ipv6_configuration_mode`	`ipv6_configuration_mode`	
`string set`	`IPv6`	
`string`	`ipv6_gateway`	
`primary_address_type`	`primary_address_type`	
`bool`	`managed`	
`(string -> string) map`	`properties`	

Minimum Role: pool-operator

Return Type: `PIF ref`

The ref of the newly created PIF record.

RPC name: destroy

This message is deprecated.

Overview:

Destroy the PIF object provided it is a VLAN interface. This call is deprecated: use VLAN.destroy or Bond.destroy instead

Signature:

```
void destroy (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to destroy

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `PIF_IS_PHYSICAL`

RPC name: forget

Overview:

Destroy the PIF object matching a particular network interface

Signature:

```
void forget (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	The PIF object to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `PIF_TUNNEL_STILL_EXISTS`, `CLUSTERING_ENABLED`

RPC name: getall

Overview:

Return a list of all the PIFs known to the system.

Signature:

```
PIF ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `PIF ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PIF references to PIF records for all PIFs known to the system.

Signature:

```
(PIF ref -> PIF record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PIF ref -> PIF record) map`

records of all objects

RPC name: getbondmasterof

Overview:

Get the bondmasterof field of the given PIF.

Signature:

```
Bond ref set get_bond_master_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Bond ref set`

value of the field

RPC name: getbondslaveof

Overview:

Get the bondslaveof field of the given PIF.

Signature:

```
Bond ref get_bond_slave_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Bond ref`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the PIF instance with the specified UUID.

Signature:

```
PIF ref get_by_uuid (session ref session_ref, string uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PIF ref`

reference to the object

RPC name: getcapabilities

Overview:

Get the capabilities field of the given PIF.

Signature:

```
string set get_capabilities (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_currently_attached (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdevice

Overview:

Get the device field of the given PIF.

Signature:

```
string get_device (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdisallowunplug

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the disallowunplug field of the given PIF.

Signature:

```
bool get_disallow_unplug (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getDNS

Overview:

Get the DNS field of the given PIF.

Signature:

```
string get_DNS (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getgateway

Overview:

Get the gateway field of the given PIF.

Signature:

```
string get_gateway (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given PIF.

Signature:

```
host ref get_host (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `host ref`

value of the field

RPC name: getigmpsnoopingstatus

Overview:

Get the igmpsnoopingstatus field of the given PIF.

Signature:

```
pif_igmp_status get_igmp_snooping_status (session ref session_ref, PIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pif_igmp_status`

value of the field

RPC name: getIP

Overview:

Get the IP field of the given PIF.

Signature:

```
string get_IP (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getipconfigurationmode

Overview:

Get the ipconfigurationmode field of the given PIF.

Signature:

```
ip_configuration_mode get_ip_configuration_mode (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `ip_configuration_mode`

value of the field

RPC name: getIPv6

Overview:

Get the IPv6 field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set get_IPv6 (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv6configurationmode

Overview:

Get the ipv6configurationmode field of the given PIF.

Signature:

```
ipv6_configuration_mode get_ipv6_configuration_mode (session ref
session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `ipv6_configuration_mode`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getipv6gateway

Overview:

Get the ipv6gateway field of the given PIF.

Signature:

```
string get_ipv6_gateway (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getMAC

Overview:

Get the MAC field of the given PIF.

Signature:

```
string get_MAC (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getmanaged

Overview:

Get the managed field of the given PIF.

Signature:

```
bool get_managed (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmanagement

Overview:

Get the management field of the given PIF.

Signature:

```
bool get_management (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetrics

Overview:

Get the metrics field of the given PIF.

Signature:

```
PIF_metrics ref get_metrics (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF_metrics ref`

value of the field

RPC name: getMTU

Overview:

Get the MTU field of the given PIF.

Signature:

```
int get_MTU (session ref session_ref, PIF ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnetmask

Overview:

Get the netmask field of the given PIF.

Signature:

```
string get_netmask (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnetwork

Overview:

Get the network field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
network ref get_network (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PIF.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getPCI

Overview:

Get the PCI field of the given PIF.

Signature:

```
PCI ref get_PCI (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref`

value of the field

RPC name: getphysical

Overview:

Get the physical field of the given PIF.

Signature:

```
bool get_physical (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getprimaryaddresstype

Overview:

Get the primaryaddresstype field of the given PIF.

Signature:

```
primary_address_type get_primary_address_type (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `primary_address_type`

value of the field

RPC name: getproperties

Overview:

Get the properties field of the given PIF.

Signature:

```
(string -> string) map get_properties (session ref session_ref, PIF ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PIF.

Signature:

```
PIF record get_record (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF record`

all fields from the object

RPC name: getsriovlogicalPIFof

Overview:

Get the sriovlogicalPIFof field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
network_sriov ref set get_sriov_logical_PIF_of (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_sriov ref set`

value of the field

RPC name: getsriovphysicalPIFof

Overview:

Get the sriovphysicalPIFof field of the given PIF.

Signature:

```
network_sriov ref set get_sriov_physical_PIF_of (session ref session_ref,
PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network_sriov ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettunnelaccessPIFof

Overview:

Get the tunnelaccessPIFof field of the given PIF.

Signature:

```
tunnel ref set get_tunnel_access_PIF_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tunnel ref set`

value of the field

RPC name: gettunneltransportPIFof

Overview:

Get the tunneltransportPIFof field of the given PIF.

Signature:

```
tunnel ref set get_tunnel_transport_PIF_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `tunnel ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PIF.

Signature:

```
string get_uuid (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVLAN

Overview:

Get the VLAN field of the given PIF.

Signature:

```
int get_VLAN (session ref session_ref, PIF ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVLANmasterof

Overview:

Get the VLANmasterof field of the given PIF.

Signature:

```
VLAN ref get_VLAN_master_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VLAN ref`

value of the field

RPC name: getVLANslaveof

Overview:

Get the VLANslaveof field of the given PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VLAN ref set get_VLAN_slave_of (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VLAN ref set`

value of the field

RPC name: introduce

Overview:

Create a PIF object matching a particular network interface

Signature:

```
PIF ref introduce (session ref session_ref, host ref host, string MAC,
string device, bool managed)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host on which the interface exists
`string`	`MAC`	The MAC address of the interface
`string`	`device`	The device name to use for the interface

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`managed`	Indicates whether the interface is managed by xapi (defaults to "true")

Minimum Role: pool-operator

Return Type: `PIF ref`

The reference of the created PIF object

RPC name: plug

Overview:

Attempt to bring up a physical interface

Signature:

```
void plug (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to plug

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `TRANSPORT_PIF_NOT_CONFIGURED`

RPC name: reconfigureip

Overview:

Reconfigure the IP address settings for this interface

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void reconfigure_ip (session ref session_ref, PIF ref self,
ip_configuration_mode mode, string IP, string netmask, string gateway,
string DNS)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to reconfigure
`ip_configuration_mode`	`mode`	whether to use dynamic/ static/no-assignment
`string`	`IP`	the new IP address
`string`	`netmask`	the new netmask
`string`	`gateway`	the new gateway
`string`	`DNS`	the new DNS settings

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTERING_ENABLED`

RPC name: reconfigureipv6

Overview:

Reconfigure the IPv6 address settings for this interface

Signature:

```
void reconfigure_ipv6 (session ref session_ref, PIF ref self,
ipv6_configuration_mode mode, string IPv6, string gateway, string DNS)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to reconfigure
`ipv6_configuration_mode`	`mode`	whether to use dynamic/ static/no-assignment
`string`	`IPv6`	the new IPv6 address (in <addr>/<prefix length> format)
`string`	`gateway`	the new gateway
`string`	`DNS`	the new DNS settings

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `CLUSTERING_ENABLED`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PIF. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PIF ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: scan

Overview:

Scan for physical interfaces on a host and create PIF objects to represent them

Signature:

```
void scan (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host on which to scan

Minimum Role: pool-operator

Return Type: `void`

RPC name: setdisallowunplug

Overview:

Set whether unplugging the PIF is allowed

Signature:

```
void set_disallow_unplug (session ref session_ref, PIF ref self, bool
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	Reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OTHER_OPERATION_IN_PROGRESS`, `CLUSTERING_ENABLED`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PIF.

Signature:

```
void set_other_config (session ref session_ref, PIF ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setprimaryaddresstype

Overview:

Change the primary address type used by this PIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_primary_address_type (session ref session_ref, PIF ref self,
primary_address_type primary_address_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to reconfigure
`primary_address_type`	`primary_address_type`	Whether to prefer IPv4 or IPv6 connections

Minimum Role: pool-operator

Return Type: `void`

RPC name: setproperty

Overview:

Set the value of a property of the PIF

Signature:

```
void set_property (session ref session_ref, PIF ref self, string name,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	The PIF
`string`	`name`	The property name
`string`	`value`	The property value

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: unplug

Overview:

Attempt to bring down a physical interface

Signature:

```
void unplug (session ref session_ref, PIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`self`	the PIF object to unplug

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN`, `VIF_IN_USE`, `PIF_DOES_NOT_ALLOW_UNPLUG`, `PIF_HAS_FCOE_SR_IN_USE`

Class: PIFmetrics

The metrics associated with a physical network interface

Fields for class: PIFmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`carrier`	`bool`	*RO/runtime*	Report if the PIF got a carrier or not
`device_id`	`string`	*RO/runtime*	Report device ID
`device_name`	`string`	*RO/runtime*	Report device name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`duplex`	`bool`	*RO/runtime*	Full duplex capability of the link (if available)
`io_read_kbs`	`float`	*RO/runtime*	**Removed.** Read bandwidth (KiB/s)
`io_write_kbs`	`float`	*RO/runtime*	**Removed.** Write bandwidth (KiB/s)
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated
`other_config`	`(string -> string) map`	*RW*	additional configuration
`pci_bus_path`	`string`	*RO/runtime*	PCI bus path of the pif (if available)
`speed`	`int`	*RO/runtime*	Speed of the link in Mbit/s (if available)
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_id`	`string`	*RO/runtime*	Report vendor ID
`vendor_name`	`string`	*RO/runtime*	Report vendor name

RPCs associated with class: PIFmetrics

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PIFmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_other_config (session ref session_ref, PIF_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PIFmetrics instances known to the system.

Signature:

```
PIF_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PIF_metrics ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of PIFmetrics references to PIFmetrics records for all PIFmetrics instances known to the system.

Signature:

```
(PIF_metrics ref -> PIF_metrics record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PIF_metrics ref -> PIF_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PIFmetrics instance with the specified UUID.

Signature:

```
PIF_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `PIF_metrics ref`

reference to the object

RPC name: getcarrier

Overview:

Get the carrier field of the given PIFmetrics.

Signature:

```
bool get_carrier (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdeviceid

Overview:

Get the deviceid field of the given PIFmetrics.

Signature:

```
string get_device_id (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdevicename

Overview:

Get the devicename field of the given PIFmetrics.

Signature:

```
string get_device_name (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getduplex

Overview:

Get the duplex field of the given PIFmetrics.

Signature:

```
bool get_duplex (session ref session_ref, PIF_metrics ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getioreadkbs

This message is removed.

Overview:

Get the io/readkbs field of the given PIFmetrics.

Signature:

```
float get_io_read_kbs (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getiowritekbs

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the io/writekbs field of the given PIFmetrics.

Signature:

```
float get_io_write_kbs (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given PIFmetrics.

Signature:

```
datetime get_last_updated (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PIFmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpcibuspath

Overview:

Get the pcibuspath field of the given PIFmetrics.

Signature:

```
string get_pci_bus_path (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PIFmetrics.

Signature:

```
PIF_metrics record get_record (session ref session_ref, PIF_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF_metrics record`

all fields from the object

RPC name: getspeed

Overview:

Get the speed field of the given PIFmetrics.

Signature:

```
int get_speed (session ref session_ref, PIF_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PIFmetrics.

Signature:

```
string get_uuid (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorid

Overview:

Get the vendorid field of the given PIFmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_vendor_id (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorname

Overview:

Get the vendorname field of the given PIFmetrics.

Signature:

```
string get_vendor_name (session ref session_ref, PIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given PIFmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PIF_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PIFmetrics.

Signature:

```
void set_other_config (session ref session_ref, PIF_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

Class: pool

Poolwide information

Fields for class: pool

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`pool_allowed_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`auto_update_vm_secureboot_certificates`	`bool`	*RW*	When true, at VM.create time the pool automatically marks a newly created VM whose Secure Boot certificates are due to expire by setting VM.secureboot_certificates_state to update_on_boot (the same effect as VM.update_secureboot_certificates_on_boot), so the certificates are updated on the VM's next boot. This applies to VMs created by VM.create, including those created by operations that

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			internally call it: the HTTP(s) PUT /import and / import_metadata calls and disaster recovery (VM.recover and VM_appliance.recover); it does not apply to VM.clone or VM.copy, which inherit the source VM's state.
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this pool
`client_certificate_auth_enabled`	`bool`	*RO/runtime*	True if authentication by TLS client certificates is enabled
`client_certificate_auth_name`	`string`	*RO/runtime*	The name (CN/SAN) that an incoming client certificate must have to allow authentication
`coordinator_bias`	`bool`	*RW*	true if bias against pool master when scheduling vms is enabled, false otherwise
`cpu_info`	`(string -> string) map`	*RO/runtime*	Details about the physical CPUs on the pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`crash_dump_SR`	`SR ref`	*RW*	The SR in which VDIs for crash dumps are created
`current_operations`	`(string -> pool_allowed_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`custom_uefi_certificates`	`string`	*RO/constructor*	Custom UEFI certificates allowing Secure Boot
`default_SR`	`SR ref`	*RW*	Default SR for VDIs
`ext_auth_cache_enabled`	`bool`	*RO/runtime*	Specifies whether external authentication caching is enabled for this pool or not
`ext_auth_cache_expiry`	`int`	*RO/runtime*	Specifies how long external authentication entries should be cached for (seconds)
`ext_auth_cache_size`	`int`	*RO/runtime*	Maximum capacity of external authentication cache
`ext_auth_max_threads`	`int`	*RO/constructor*	Maximum number of threads to use for external (AD) authentication

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`guest_agent_config`	`(string -> string) map`	*RO/runtime*	Pool-wide guest agent configuration information
`gui_config`	`(string -> string) map`	*RW*	gui-specific configuration for pool
`ha_allow_overcommit`	`bool`	*RW*	If set to false then operations which would cause the Pool to become overcommitted will be blocked.
`ha_cluster_stack`	`string`	*RO/runtime*	The HA cluster stack that is currently in use. Only valid when HA is enabled.
`ha_configuration`	`(string -> string) map`	*RO/runtime*	The current HA configuration
`ha_enabled`	`bool`	*RO/runtime*	true if HA is enabled on the pool, false otherwise
`ha_host_failures_to_tolerate`	`int`	*RO/runtime*	Number of host failures to tolerate before the Pool is declared to be overcommitted
`ha_overcommitted`	`bool`	*RO/runtime*	True if the Pool is considered to be overcommitted i.e. if there exist insufficient physical resources to tolerate the configured number of host failures

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`ha_plan_exists_for`	`int`	*RO/runtime*	Number of future host failures we have managed to find a plan for. Once this reaches zero any future host failures will cause the failure of protected VMs.
`ha_reboot_vm_on_internal_shutdown`	`bool`	*RW*	Indicates whether an HA-protected VM that is shut down from inside (not through the API) should be automatically rebooted when HA is enabled
`ha_statefiles`	`string set`	*RO/runtime*	HA statefile VDIs in use
`health_check_config`	`(string -> string) map`	*RW*	Configuration for the automatic health check feature
`igmp_snooping_enabled`	`bool`	*RO/runtime*	true if IGMP snooping is enabled in the pool, false otherwise.
`is_psr_pending`	`bool`	*RW*	True if either a PSR is running or we are waiting for a PSR to be re-run
`last_update_sync`	`datetime`	*RO/runtime*	time of the last update sychronization

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`license_server`	`(string -> string) map`	*RW*	Licensing data shared within the whole pool
`limit_console_sessions`	`bool`	*RW*	When true, only one console connection per VM/host in the pool is accepted. Otherwise every connection for a VM/host's console is accepted. Note: when true, connection attempts via websocket will be rejected.
`live_patching_disabled`	`bool`	*RW*	The pool-wide flag to show if the live patching feauture is disabled or not.
`local_auth_max_threads`	`int`	*RO/constructor*	Maximum number of threads to use for PAM authentication
`master`	`host ref`	*RO/runtime*	The host that is pool master
`metadata_VDIs`	`VDI ref set`	*RO/runtime*	The set of currently known metadata VDIs for this pool
`migration_compression`	`bool`	*RW*	Default behaviour during migration, True if stream compression should be used
`name_description`	`string`	*RW*	Description

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_label`	`string`	*RW*	Short name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`policy_no_vendor_device`	`bool`	*RW*	**Deprecated.** This field was consulted when VM.create did not specify a value for 'has_vendor_device'; VM.create now uses a simple default and no longer consults this value.
`recommendations`	`(string -> string) map`	*RO/runtime*	The recommended pool properties for clients to respect for optimal performance. for example, max-vm-group=5
`redo_log_enabled`	`bool`	*RO/runtime*	true a redo-log is to be used other than when HA is enabled, false otherwise
`redo_log_vdi`	`VDI ref`	*RO/runtime*	indicates the VDI to use for the redo-log other than when HA is enabled
`repositories`	`Repository ref set`	*RO/runtime*	The set of currently enabled repositories
`repository_proxy_password`	`secret ref`	*RO/runtime*	Password for the authentication of the proxy used in syncing with the enabled repositories

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`repository_proxy_url`	`string`	*RO/runtime*	Url of the proxy used in syncing with the enabled repositories
`repository_proxy_username`	`string`	*RO/runtime*	Username for the authentication of the proxy used in syncing with the enabled repositories
`restrictions`	`(string -> string) map`	*RO/runtime*	Pool-wide restrictions currently in effect
`suspend_image_SR`	`SR ref`	*RW*	The SR in which VDIs for suspend images are created
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`telemetry_frequency`	`telemetry_frequency`	*RO/runtime*	How often the telemetry collection will be carried out
`telemetry_next_collection`	`datetime`	*RO/runtime*	The earliest timestamp (in UTC) when the next round of telemetry collection can be carried out
`telemetry_uuid`	`secret ref`	*RO/runtime*	The UUID of the pool for identification of telemetry data
`tls_verification_enabled`	`bool`	*RO/runtime*	True iff TLS certificate verification is enabled

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uefi_certificates`	`string`	*RO/constructor*	The UEFI certificates allowing Secure Boot
`update_sync_day`	`int`	*RO/runtime*	The day of the week the update synchronizations will be scheduled, based on pool's local timezone. Ignored when update_sync_frequency is daily
`update_sync_enabled`	`bool`	*RO/runtime*	Whether periodic update synchronization is enabled or not
`update_sync_frequency`	`update_sync_frequency`	*RO/runtime*	The frequency at which updates are synchronized from a remote CDN: daily or weekly.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vm_console_idle_timeout`	`int`	*RW*	The maximum time (in seconds) that a VM's console can be idle before it is automatically disconnected. The default value 0 means never timeout. This setting applies only to VM consoles; for host consoles, use the separate parameter

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			'host.console_idle_timeout'.
`vswitch_controller`	`string`	*RO/runtime*	**Deprecated.** address of the vswitch controller
`wlb_enabled`	`bool`	*RW*	true if workload balancing is enabled on the pool, false otherwise
`wlb_url`	`string`	*RO/runtime*	Url for the configured workload balancing host
`wlb_username`	`string`	*RO/runtime*	Username for accessing the workload balancing host
`wlb_verify_cert`	`bool`	*RW*	**Deprecated.** true if communication with the WLB server should enforce TLS certificate verification.

RPCs associated with class: pool

RPC name: addrepository

Overview:

Add a repository to the enabled set

Signature:

```
void add_repository (session ref session_ref, pool ref self, Repository ref value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Repository ref`	`value`	The repository to be added to the enabled set

Minimum Role: client-cert

Return Type: `void`

RPC name: addtags

Overview:

Add the given value to the tags field of the given pool. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtoguestagentconfig

Overview:

Add a keyvalue pair to the poolwide guest agent configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void add_to_guest_agent_config (session ref session_ref, pool ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`key`	The key to add
`string`	`value`	The value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: addtoguiconfig

Overview:

Add the given keyvalue pair to the guiconfig field of the given pool.

Signature:

```
void add_to_gui_config (session ref session_ref, pool ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtohealthcheckconfig

Overview:

Add the given keyvalue pair to the healthcheckconfig field of the given pool.

Signature:

```
void add_to_health_check_config (session ref session_ref, pool ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtolicenseserver

Overview:

Add the given keyvalue pair to the licenseserver field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_license_server (session ref session_ref, pool ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given pool.

Signature:

```
void add_to_other_config (session ref session_ref, pool ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: applyedition

Overview:

Apply an edition to all hosts in the pool

Signature:

```
void apply_edition (session ref session_ref, pool ref self, string
edition)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool
`string`	`edition`	The requested edition

Minimum Role: pool-operator

Return Type: `void`

RPC name: certificateinstall

This message is deprecated.

Overview:

Install a TLS CA certificate, poolwide.

Signature:

```
void certificate_install (session ref session_ref, string name, string
cert)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	A name to give the certificate
`string`	`cert`	The certificate in PEM format

Minimum Role: pool-operator

Return Type: `void`

RPC name: certificatelist

This message is deprecated.

Overview:

List the names of all installed TLS CA certificates.

Signature:

```
string set certificate_list (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `string set`

All installed certificates

RPC name: certificatesync

Overview:

Copy the TLS CA certificates and CRLs of the master to all slaves.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void certificate_sync (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: certificateuninstall

This message is deprecated.

Overview:

Remove a poolwide TLS CA certificate.

Signature:

```
void certificate_uninstall (session ref session_ref, string name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	The certificate name

Minimum Role: pool-operator

Return Type: `void`

RPC name: checkupdatereadiness

Overview:

Check if the pool is ready to be updated. If not, report the reasons.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set set check_update_readiness (session ref session_ref, pool ref
self, bool requires_reboot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`requires_reboot`	Assume that the update will require host reboots

Minimum Role: client-cert

Return Type: `string set set`

A set of error codes with arguments, if the pool is not ready to update. An empty list means the pool can be updated.

RPC name: configurerepositoryproxy

Overview:

Configure proxy for RPM package repositories.

Signature:

```
void configure_repository_proxy (session ref session_ref, pool ref self,
string url, string username, string password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`url`	The URL of the proxy server

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`username`	The username used to authenticate with the proxy server
`string`	`password`	The password used to authenticate with the proxy server

Minimum Role: client-cert

Return Type: `void`

RPC name: configureupdatesync

Overview:

Configure periodic update synchronization to synchronize updates from a remote CDN

Signature:

```
void configure_update_sync (session ref session_ref, pool ref self,
update_sync_frequency update_sync_frequency, int update_sync_day)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`update_sync_frequency`	`update_sync_frequency`	The frequency at which updates are synchronized from a remote CDN: daily or weekly.
`int`	`update_sync_day`	The day of the week the update synchronization will happen, based on pool's local timezone. Valid values are 0 to 6, 0 being Sunday. For

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
		'daily' schedule, the value is ignored.

Minimum Role: pool-operator

Return Type: `void`

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this pool

Signature:

```
blob ref create_new_blob (session ref session_ref, pool ref pool, string name, string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: createVLAN

Overview:

Create PIFs, mapping a network to the same physical interface/VLAN on each host. This call is deprecated: use Pool.createVLANfromPIF instead.

Signature:

```
PIF ref set create_VLAN (session ref session_ref, string device, network ref network, int VLAN)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`device`	physical interface on which to create the VLAN interface
`network ref`	`network`	network to which this interface should be connected
`int`	`VLAN`	VLAN tag for the new interface

Minimum Role: pool-operator

Return Type: `PIF ref set`

The references of the created PIF objects

Possible Error Codes: `VLAN_TAG_INVALID`

RPC name: createVLANfromPIF

Overview:

Create a poolwide VLAN by taking the PIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PIF ref set create_VLAN_from_PIF (session ref session_ref, PIF ref pif,
network ref network, int VLAN)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`pif`	physical interface on any particular host, that identifies the PIF on which to create the (pool-wide) VLAN interface
`network ref`	`network`	network to which this interface should be connected
`int`	`VLAN`	VLAN tag for the new interface

Minimum Role: pool-operator

Return Type: `PIF ref set`

The references of the created PIF objects

Possible Error Codes: `VLAN_TAG_INVALID`

RPC name: crlinstall

Overview:

Install a TLS CAissued Certificate Revocation List, poolwide.

Signature:

```
void crl_install (session ref session_ref, string name, string cert)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	A name to give the CRL
`string`	`cert`	The CRL

Minimum Role: pool-operator

Return Type: `void`

RPC name: crllist

Overview:

List the names of all installed TLS CAissued Certificate Revocation Lists.

Signature:

```
string set crl_list (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `string set`

The names of all installed CRLs

RPC name: crluninstall

Overview:

Remove a poolwide TLS CAissued Certificate Revocation List.

Signature:

```
void crl_uninstall (session ref session_ref, string name)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	The CRL name

Minimum Role: pool-operator

Return Type: `void`

RPC name: deconfigurewlb

Overview:

Permanently deconfigures workload balancing monitoring on this pool

Signature:

```
void deconfigure_wlb (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: designatenewmaster

Overview:

Perform an orderly handover of the role of master to the referenced host.

Signature:

```
void designate_new_master (session ref session_ref, host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host who should become the new master

Minimum Role: pool-operator

Return Type: `void`

RPC name: detectnonhomogeneousexternalauth

Overview:

This call asynchronously detects if the external authentication configuration in any slave is different from that in the master and raises appropriate alerts

Signature:

```
void detect_nonhomogeneous_external_auth (session ref session_ref, pool ref pool)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool where to detect non-homogeneous external authentication configuration

Minimum Role: pool-operator

Return Type: `void`

RPC name: disableclientcertificateauth

Overview:

Disable client certificate authentication on the pool

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void disable_client_certificate_auth (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: client-cert

Return Type: `void`

RPC name: disableexternalauth

Overview:

This call disables external authentication on all the hosts of the pool

Signature:

```
void disable_external_auth (session ref session_ref, pool ref pool,
(string -> string) map config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool whose external authentication should be disabled
`(string -> string) map`	`config`	Optional parameters as a list of key-values containing the configuration data

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: disableha

Overview:

Turn off High Availability mode

Signature:

```
void disable_ha (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: client-cert

Return Type: `void`

RPC name: disablelocalstoragecaching

Overview:

This call disables poolwide local storage caching

Signature:

```
void disable_local_storage_caching (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: disableredolog

Overview:

Disable the redo log if in use, unless HA is enabled.

Signature:

```
void disable_redo_log (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: disablerepositoryproxy

Overview:

Disable the proxy for RPM package repositories.

Signature:

```
void disable_repository_proxy (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: client-cert

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: disablessh

Overview:

Disable SSH access on all hosts in the pool. It's a helper which calls host.disablessh for all the hosts in the pool.

Signature:

```
void disable_ssh (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: pool-admin

Return Type: `void`

RPC name: disablessllegacy

This message is deprecated.

Overview:

Sets ssllegacy false on each host, poolmaster last. See Host.ssllegacy and Host.setssllegacy.

Signature:

```
void disable_ssl_legacy (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	(ignored)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: eject

Overview:

Instruct a pool master to eject a host from the pool

Signature:

```
void eject (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to eject

Minimum Role: pool-operator

Return Type: `void`

RPC name: emergencyresetmaster

Overview:

Instruct a slave already in a pool that the master has changed

Signature:

```
void emergency_reset_master (session ref session_ref, string
master_address)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`master_address`	The hostname of the master

Minimum Role: pool-operator

Return Type: `void`

RPC name: emergencytransitiontomaster

Overview:

Instruct host that's currently a slave to transition to being master

Signature:

```
void emergency_transition_to_master (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

RPC name: enableclientcertificateauth

Overview:

Enable client certificate authentication on the pool

Signature:

```
void enable_client_certificate_auth (session ref session_ref, pool ref
self, string name)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`name`	The name (CN/SAN) that an incoming client certificate must have to allow authentication

Minimum Role: client-cert

Return Type: `void`

RPC name: enableexternalauth

Overview:

This call enables external authentication on all the hosts of the pool

Signature:

```
void enable_external_auth (session ref session_ref, pool ref pool, (string -> string) map config, string service_name, string auth_type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`pool`	The pool whose external authentication should be enabled
`(string -> string) map`	`config`	A list of key-values containing the configuration data
`string`	`service_name`	The name of the service

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`auth_type`	The type of authentication (for example, AD for Active Directory)

Minimum Role: pool-admin

Return Type: `void`

RPC name: enableha

Overview:

Turn on High Availability mode

Signature:

```
void enable_ha (session ref session_ref, SR ref set heartbeat_srs, (string
-> string) map configuration)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref set`	`heartbeat_srs`	Set of SRs to use for storage heartbeating
`(string -> string) map`	`configuration`	Detailed HA configuration to apply

Minimum Role: client-cert

Return Type: `void`

RPC name: enablelocalstoragecaching

Overview:

This call attempts to enable poolwide local storage caching

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void enable_local_storage_caching (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool

Minimum Role: pool-operator

Return Type: `void`

RPC name: enableredolog

Overview:

Enable the redo log on the given SR and start using it, unless HA is enabled.

Signature:

```
void enable_redo_log (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	SR to hold the redo log.

Minimum Role: pool-operator

Return Type: `void`

RPC name: enablessh

Overview:

Enable SSH access on all hosts in the pool. It's a helper which calls host.enablessh for all the hosts in the pool.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void enable_ssh (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: pool-admin

Return Type: `void`

RPC name: enablessllegacy

This message is removed.

Overview:

Sets ssllegacy true on each host, poolmaster last. See Host.ssllegacy and Host.setssllegacy.

Signature:

```
void enable_ssl_legacy (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	(ignored)

Minimum Role: pool-operator

Return Type: `void`

RPC name: enabletlsverification

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable TLS server certificate verification

Signature:

```
void enable_tls_verification (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the pools known to the system.

Signature:

```
pool ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `pool ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of pool references to pool records for all pools known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(pool ref -> pool record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(pool ref -> pool record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given pool.

Signature:

```
pool_allowed_operations set get_allowed_operations (session ref
session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_allowed_operations set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getautoupdatevmsecurebootcertificates

Overview:

Get the autoupdatevmsecurebootcertificates field of the given pool.

Signature:

```
bool get_auto_update_vm_secureboot_certificates (session ref session_ref,
pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given pool.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the pool instance with the specified UUID.

Signature:

```
pool ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `pool ref`

reference to the object

RPC name: getclientcertificateauthenabled

Overview:

Get the clientcertificateauthenabled field of the given pool.

Signature:

```
bool get_client_certificate_auth_enabled (session ref session_ref, pool
ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getclientcertificateauthname

Overview:

Get the clientcertificateauthname field of the given pool.

Signature:

```
string get_client_certificate_auth_name (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getcoordinatorbias

Overview:

Get the coordinatorbias field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_coordinator_bias (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getcpuinfo

Overview:

Get the cpuinfo field of the given pool.

Signature:

```
(string -> string) map get_cpu_info (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getcrashdumpSR

Overview:

Get the crashdumpSR field of the given pool.

Signature:

```
SR ref get_crash_dump_SR (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given pool.

Signature:

```
(string -> pool_allowed_operations) map get_current_operations (session
ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> pool_allowed_operations) map`

value of the field

RPC name: getcustomueficertificates

Overview:

Get the customueficertificates field of the given pool.

Signature:

```
string get_custom_uefi_certificates (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdefaultSR

Overview:

Get the defaultSR field of the given pool.

Signature:

```
SR ref get_default_SR (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getextauthcacheenabled

Overview:

Get the extauthcacheenabled field of the given pool.

Signature:

```
bool get_ext_auth_cache_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getextauthcacheexpiry

Overview:

Get the extauthcacheexpiry field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int get_ext_auth_cache_expiry (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getextauthcachesize

Overview:

Get the extauthcachesize field of the given pool.

Signature:

```
int get_ext_auth_cache_size (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getextauthmaxthreads

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the extauthmaxthreads field of the given pool.

Signature:

```
int get_ext_auth_max_threads (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getguestagentconfig

Overview:

Get the guestagentconfig field of the given pool.

Signature:

```
(string -> string) map get_guest_agent_config (session ref session_ref,
pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getguestsecurebootreadiness

Overview:

Signature:

```
pool_guest_secureboot_readiness get_guest_secureboot_readiness (session
ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

Minimum Role: pool-operator

Return Type: `pool_guest_secureboot_readiness`

The readiness of the pool

RPC name: getguiconfig

Overview:

Get the guiconfig field of the given pool.

Signature:

```
(string -> string) map get_gui_config (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethaallowovercommit

Overview:

Get the haallowovercommit field of the given pool.

Signature:

```
bool get_ha_allow_overcommit (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethaclusterstack

Overview:

Get the haclusterstack field of the given pool.

Signature:

```
string get_ha_cluster_stack (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethaconfiguration

Overview:

Get the haconfiguration field of the given pool.

Signature:

```
(string -> string) map get_ha_configuration (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gethaenabled

Overview:

Get the haenabled field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_ha_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethahostfailurestotolerate

Overview:

Get the hahostfailurestotolerate field of the given pool.

Signature:

```
int get_ha_host_failures_to_tolerate (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethaovercommitted

Overview:

Get the haovercommitted field of the given pool.

Signature:

```
bool get_ha_overcommitted (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethaplanexistsfor

Overview:

Get the haplanexistsfor field of the given pool.

Signature:

```
int get_ha_plan_exists_for (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getharebootvmoninternalshutdown

Overview:

Get the harebootvmoninternalshutdown field of the given pool.

Signature:

```
bool get_ha_reboot_vm_on_internal_shutdown (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethastatefiles

Overview:

Get the hastatefiles field of the given pool.

Signature:

```
string set get_ha_statefiles (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gethealthcheckconfig

Overview:

Get the healthcheckconfig field of the given pool.

Signature:

```
(string -> string) map get_health_check_config (session ref session_ref,
pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getigmpsnoopingenabled

Overview:

Get the igmpsnoopingenabled field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_igmp_snooping_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getispsrpending

Overview:

Get the ispsrpending field of the given pool.

Signature:

```
bool get_is_psr_pending (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlastupdatesync

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the lastupdatesync field of the given pool.

Signature:

```
datetime get_last_update_sync (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlicenseserver

Overview:

Get the licenseserver field of the given pool.

Signature:

```
(string -> string) map get_license_server (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getlicensestate

Overview:

This call returns the license state for the pool

Signature:

```
(string -> string) map get_license_state (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool

Minimum Role: read-only

Return Type: `(string -> string) map`

The pool's license state

RPC name: getlimitconsolesessions

Overview:

Get the limitconsolesessions field of the given pool.

Signature:

```
bool get_limit_console_sessions (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlivepatchingdisabled

Overview:

Get the livepatchingdisabled field of the given pool.

Signature:

```
bool get_live_patching_disabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlocalauthmaxthreads

Overview:

Get the localauthmaxthreads field of the given pool.

Signature:

```
int get_local_auth_max_threads (session ref session_ref, pool ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmaster

Overview:

Get the master field of the given pool.

Signature:

```
host ref get_master (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getmetadataVDIs

Overview:

Get the metadataVDIs field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VDI ref set get_metadata_VDIs (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref set`

value of the field

RPC name: getmigrationcompression

Overview:

Get the migrationcompression field of the given pool.

Signature:

```
bool get_migration_compression (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the namedescription field of the given pool.

Signature:

```
string get_name_description (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the namelabel field of the given pool.

Signature:

```
string get_name_label (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given pool.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpolicynovendordevice

This message is deprecated.

Overview:

Get the policynovendordevice field of the given pool.

Signature:

```
bool get_policy_no_vendor_device (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrecommendations

Overview:

Get the recommendations field of the given pool.

Signature:

```
(string -> string) map get_recommendations (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given pool.

Signature:

```
pool record get_record (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool record`

all fields from the object

RPC name: getredologenabled

Overview:

Get the redologenabled field of the given pool.

Signature:

```
bool get_redo_log_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getredologvdi

Overview:

Get the redologvdi field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VDI ref get_redo_log_vdi (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getrepositories

Overview:

Get the repositories field of the given pool.

Signature:

```
Repository ref set get_repositories (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Repository ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrepositoryproxypassword

Overview:

Get the repositoryproxypassword field of the given pool.

Signature:

```
secret ref get_repository_proxy_password (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `secret ref`

value of the field

RPC name: getrepositoryproxyurl

Overview:

Get the repositoryproxyurl field of the given pool.

Signature:

```
string get_repository_proxy_url (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrepositoryproxyusername

Overview:

Get the repositoryproxyusername field of the given pool.

Signature:

```
string get_repository_proxy_username (session ref session_ref, pool ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrestrictions

Overview:

Get the restrictions field of the given pool.

Signature:

```
(string -> string) map get_restrictions (session ref session_ref, pool ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsuspendimageSR

Overview:

Get the suspendimageSR field of the given pool.

Signature:

```
SR ref get_suspend_image_SR (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: gettags

Overview:

Get the tags field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set get_tags (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettelemetryfrequency

Overview:

Get the telemetryfrequency field of the given pool.

Signature:

```
telemetry_frequency get_telemetry_frequency (session ref session_ref, pool
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `telemetry_frequency`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettelemetrynextcollection

Overview:

Get the telemetrynextcollection field of the given pool.

Signature:

```
datetime get_telemetry_next_collection (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: gettelemetryuuid

Overview:

Get the telemetryuuid field of the given pool.

Signature:

```
secret ref get_telemetry_uuid (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `secret ref`

value of the field

RPC name: gettlsverificationenabled

Overview:

Get the tlsverificationenabled field of the given pool.

Signature:

```
bool get_tls_verification_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getueficertificates

Overview:

Get the ueficertificates field of the given pool.

Signature:

```
string get_uefi_certificates (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getupdatesyncday

Overview:

Get the updatesyncday field of the given pool.

Signature:

```
int get_update_sync_day (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getupdatesyncenabled

Overview:

Get the updatesyncenabled field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_update_sync_enabled (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getupdatesyncfrequency

Overview:

Get the updatesyncfrequency field of the given pool.

Signature:

```
update_sync_frequency get_update_sync_frequency (session ref session_ref,
pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_sync_frequency`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given pool.

Signature:

```
string get_uuid (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvmconsoleidletimeout

Overview:

Get the vmconsoleidletimeout field of the given pool.

Signature:

```
int get_vm_console_idle_timeout (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getvswitchcontroller

This message is deprecated.

Overview:

Get the vswitchcontroller field of the given pool.

Signature:

```
string get_vswitch_controller (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getwlbenabled

Overview:

Get the wlbenabled field of the given pool.

Signature:

```
bool get_wlb_enabled (session ref session_ref, pool ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getwlburl

Overview:

Get the wlburl field of the given pool.

Signature:

```
string get_wlb_url (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getwlbusername

Overview:

Get the wlbusername field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_wlb_username (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getwlbverifycert

This message is deprecated.

Overview:

Get the wlbverifycert field of the given pool.

Signature:

```
bool get_wlb_verify_cert (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: hacomputehypotheticalmaxhostfailurestotolerate

Overview:

Returns the maximum number of host failures we could tolerate before we would be unable to restart the provided VMs

Signature:

```
int ha_compute_hypothetical_max_host_failures_to_tolerate (session ref session_ref, (VM ref -> string) map configuration)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`(VM ref -> string) map`	`configuration`	Map of protected VM reference to restart priority

Minimum Role: read-only

Return Type: `int`

maximum value for hahostfailurestotolerate given provided configuration

RPC name: hacomputemaxhostfailurestotolerate

Overview:

Returns the maximum number of host failures we could tolerate before we would be unable to restart configured VMs

Signature:

```
int ha_compute_max_host_failures_to_tolerate (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `int`

maximum value for hahostfailurestotolerate given current configuration

RPC name: hacomputevmfailoverplan

Overview:

Return a VM failover plan assuming a given subset of hosts fail

Signature:

```
(VM ref -> (string -> string) map) map ha_compute_vm_failover_plan
(session ref session_ref, host ref set failed_hosts, VM ref set
failed_vms)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref set`	`failed_hosts`	The set of hosts to assume have failed
`VM ref set`	`failed_vms`	The set of VMs to restart

Minimum Role: pool-operator

Return Type: `(VM ref -> (string -> string) map) map`

VM failover plan: a map of VM to host to restart the host on

RPC name: hafailoverplanexists

Overview:

Returns true if a VM failover plan exists for up to 'n' host failures

Signature:

```
bool ha_failover_plan_exists (session ref session_ref, int n)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`int`	`n`	The number of host failures to plan for

Minimum Role: pool-operator

Return Type: `bool`

true if a failover plan exists for the supplied number of host failures

RPC name: hapreventrestartsfor

Overview:

When this call returns the VM restart logic will not run for the requested number of seconds. If the argument is zero then the restart thread is immediately unblocked

Signature:

```
void ha_prevent_restarts_for (session ref session_ref, int seconds)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`int`	`seconds`	The number of seconds to block the restart thread for

Minimum Role: pool-operator

Return Type: `void`

RPC name: hasextension

Overview:

Return true if the extension is available on the pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
bool has_extension (session ref session_ref, pool ref self, string name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`name`	The name of the API call

Minimum Role: pool-admin

Return Type: `bool`

True if the extension exists, false otherwise

RPC name: initializewlb

Overview:

Initializes workload balancing monitoring on this pool with the specified wlb server

Signature:

```
void initialize_wlb (session ref session_ref, string wlb_url, string
wlb_username, string wlb_password, string xenserver_username, string
xenserver_password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`wlb_url`	The ip address and port to use when accessing the wlb server

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`wlb_username`	The username used to authenticate with the wlb server
`string`	`wlb_password`	The password used to authenticate with the wlb server
`string`	`xenserver_username`	The username used by the wlb server to authenticate with the xenserver
`string`	`xenserver_password`	The password used by the wlb server to authenticate with the xenserver

Minimum Role: pool-operator

Return Type: `void`

RPC name: installcacertificate

This message is deprecated.

Overview:

Install a TLS CA certificate, poolwide.

Signature:

```
void install_ca_certificate (session ref session_ref, string name, string
cert)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	A name to give the certificate

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`cert`	The certificate in PEM format

Minimum Role: client-cert

Return Type: `void`

RPC name: installtrustedcertificate

Overview:

Install a trusted TLS certificate, poolwide.

Signature:

```
void install_trusted_certificate (session ref session_ref, pool ref self,
bool ca, string cert, certificate_purpose set purpose)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`ca`	The trusted certificate is a root CA certificate used to verify a chain (true), or a leaf certificate used for certificate pinning (false)
`string`	`cert`	The certificate in PEM format
`certificate_purpose set`	`purpose`	The purpose of the certificate

Minimum Role: client-cert

Return Type: `void`

RPC name: join

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Instruct host to join a new pool

Signature:

```
void join (session ref session_ref, string master_address, string
master_username, string master_password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`master_address`	The hostname of the master of the pool to join
`string`	`master_username`	The username of the master (for initial authentication)
`string`	`master_password`	The password for the master (for initial authentication)

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `JOINING_HOST_CANNOT_CONTAIN_SHARED_SRS`

RPC name: joinforce

Overview:

Instruct host to join a new pool

Signature:

```
void join_force (session ref session_ref, string master_address, string
master_username, string master_password)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`master_address`	The hostname of the master of the pool to join
`string`	`master_username`	The username of the master (for initial authentication)
`string`	`master_password`	The password for the master (for initial authentication)

Minimum Role: pool-operator

Return Type: `void`

RPC name: managementreconfigure

Overview:

Reconfigure the management network interface for all Hosts in the Pool

Signature:

```
void management_reconfigure (session ref session_ref, network ref network)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`network ref`	`network`	The network

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `HA_IS_ENABLED`, `PIF_NOT_PRESENT`, `CANNOT_PLUG_BOND_SLAVE`, `PIF_INCOMPATIBLE_PRIMARY_ADDRESS_TYPE`, `PIF_HAS_NO_NETWORK_CONFIGURATION`, `PIF_HAS_NO_V6_NETWORK_CONFIGURATION`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: recoverslaves

Overview:

Instruct a pool master, M, to try and contact its slaves and, if slaves are in emergency mode, reset their master address to M.

Signature:

```
host ref set recover_slaves (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `host ref set`

list of hosts whose master address were successfully reset

RPC name: removefromguestagentconfig

Overview:

Remove a keyvalue pair from the poolwide guest agent configuration

Signature:

```
void remove_from_guest_agent_config (session ref session_ref, pool ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`key`	The key to remove

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: removefromguiconfig

Overview:

Remove the given key and its corresponding value from the guiconfig field of the given pool. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_gui_config (session ref session_ref, pool ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: removefromhealthcheckconfig

Overview:

Remove the given key and its corresponding value from the healthcheckconfig field of the given pool. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_health_check_config (session ref session_ref, pool ref
self, string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromlicenseserver

Overview:

Remove the given key and its corresponding value from the licenseserver field of the given pool. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_license_server (session ref session_ref, pool ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given pool. If the key is not in that Map, then do nothing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void remove_from_other_config (session ref session_ref, pool ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removerepository

Overview:

Remove a repository from the enabled set

Signature:

```
void remove_repository (session ref session_ref, pool ref self, Repository
ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Repository ref`	`value`	The repository to be removed

Minimum Role: client-cert

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given pool. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: resettelemetryuuid

Overview:

Assign a new UUID to telemetry data.

Signature:

```
void reset_telemetry_uuid (session ref session_ref, pool ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	The pool

Minimum Role: pool-admin

Return Type: `void`

RPC name: retrievewlbconfiguration

Overview:

Retrieves the pool optimization criteria from the workload balancing server

Signature:

```
(string -> string) map retrieve_wlb_configuration (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(string -> string) map`

The configuration used in optimizing this pool

RPC name: retrievewlbrecommendations

Overview:

Retrieves vm migrate recommendations for the pool from the workload balancing server

Signature:

```
(VM ref -> string set) map retrieve_wlb_recommendations (session ref
session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM ref -> string set) map`

The list of vm migration recommendations

RPC name: rotatesecret

Overview:

Signature:

```
void rotate_secret (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-admin

Return Type: `void`

Possible Error Codes: `INTERNAL_ERROR`, `HOST_IS_SLAVE`, `CANNOT_CONTACT_HOST`, `HA_IS_ENABLED`, `NOT_SUPPORTED_DURING_UPGRADE`

RPC name: sendtestpost

Overview:

Send the given body to the given host and port, using HTTPS, and print the response. This is used for debugging the SSL layer.

Signature:

```
string send_test_post (session ref session_ref, string host, int port,
string body)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`host`	
`int`	`port`	
`string`	`body`	

Minimum Role: pool-admin

Return Type: `string`

The response

RPC name: sendwlbconfiguration

Overview:

Sets the pool optimization criteria for the workload balancing server

Signature:

```
void send_wlb_configuration (session ref session_ref, (string -> string)
map config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`(string -> string) map`	`config`	The configuration to use in optimizing this pool

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setautoupdatevmsecurebootcertificates

Overview:

Set the autoupdatevmsecurebootcertificates field of the given pool.

Signature:

```
void set_auto_update_vm_secureboot_certificates (session ref session_ref,
pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setconsoleidletimeout

Overview:

Set the console idle timeout for all hosts in the pool

Signature:

```
void set_console_idle_timeout (session ref session_ref, pool ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`int`	`value`	The idle SSH/VNC session timeout in seconds. A value of 0 means no timeout.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setcoordinatorbias

Overview:

Set the coordinatorbias field of the given pool.

Signature:

```
void set_coordinator_bias (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setcrashdumpSR

Overview:

Set the crashdumpSR field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_crash_dump_SR (session ref session_ref, pool ref self, SR ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setcustomueficertificates

Overview:

Set custom UEFI certificates for a pool and all its hosts. Need `allow-custom-uefi-certs` set to true in conf. If empty: default back to Pool.ueficertificates

Signature:

```
void set_custom_uefi_certificates (session ref session_ref, pool ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`value`	The certificates to apply to the pool and its hosts

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setdefaultSR

Overview:

Set the defaultSR field of the given pool.

Signature:

```
void set_default_SR (session ref session_ref, pool ref self, SR ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthcacheenabled

Overview:

Signature:

```
void set_ext_auth_cache_enabled (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	Specifies whether caching is enabled for external authentication

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthcacheexpiry

Overview:

Signature:

```
void set_ext_auth_cache_expiry (session ref session_ref, pool ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The expiry time of entries in the external authentication cache (in seconds - 300 seconds, i.e. 5 minutes, is the default value)

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthcachesize

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_ext_auth_cache_size (session ref session_ref, pool ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The capacity of the external authentication cache

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextauthmaxthreads

Overview:

Signature:

```
void set_ext_auth_max_threads (session ref session_ref, pool ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The new maximum

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setguiconfig

Overview:

Set the guiconfig field of the given pool.

Signature:

```
void set_gui_config (session ref session_ref, pool ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: sethaallowovercommit

Overview:

Set the haallowovercommit field of the given pool.

Signature:

```
void set_ha_allow_overcommit (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethahostfailurestotolerate

Overview:

Set the maximum number of host failures to consider in the HA VM restart planner

Signature:

```
void set_ha_host_failures_to_tolerate (session ref session_ref, pool ref
self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	New number of host failures to consider

Minimum Role: pool-operator

Return Type: `void`

RPC name: setharebootvmoninternalshutdown

Overview:

Set the harebootvmoninternalshutdown field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_ha_reboot_vm_on_internal_shutdown (session ref session_ref, pool
ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethealthcheckconfig

Overview:

Set the healthcheckconfig field of the given pool.

Signature:

```
void set_health_check_config (session ref session_ref, pool ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: sethttpsonly

Overview:

updates all the host firewalls in the pool to open or close port 80 depending on the value

Signature:

```
void set_https_only (session ref session_ref, pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`value`	true - http port 80 will be blocked, false - http port 80 will be open for the hosts in the pool

Minimum Role: pool-operator

Return Type: `void`

RPC name: setigmpsnoopingenabled

Overview:

Enable or disable IGMP Snooping on the pool.

Signature:

```
void set_igmp_snooping_enabled (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	The pool
`bool`	`value`	Enable or disable IGMP Snooping on the pool

Minimum Role: pool-operator

Return Type: `void`

RPC name: setispsrpending

Overview:

Set the ispsrpending field of the given pool.

Signature:

```
void set_is_psr_pending (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlicenseserver

Overview:

Set the licenseserver field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_license_server (session ref session_ref, pool ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlimitconsolesessions

Overview:

Set the limitconsolesessions field of the given pool.

Signature:

```
void set_limit_console_sessions (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setlivepatchingdisabled

Overview:

Set the livepatchingdisabled field of the given pool.

Signature:

```
void set_live_patching_disabled (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlocalauthmaxthreads

Overview:

Signature:

```
void set_local_auth_max_threads (session ref session_ref, pool ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The new maximum

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setmigrationcompression

Overview:

Set the migrationcompression field of the given pool.

Signature:

```
void set_migration_compression (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the namedescription field of the given pool.

Signature:

```
void set_name_description (session ref session_ref, pool ref self, string
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the namelabel field of the given pool.

Signature:

```
void set_name_label (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given pool.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_other_config (session ref session_ref, pool ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setpolicynovendordevice

This message is deprecated.

Overview:

Set the policynovendordevice field of the given pool.

Signature:

```
void set_policy_no_vendor_device (session ref session_ref, pool ref self,
bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setrepositories

Overview:

Set enabled set of repositories

Signature:

```
void set_repositories (session ref session_ref, pool ref self, Repository ref set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Repository ref set`	`value`	The set of repositories to be enabled

Minimum Role: client-cert

Return Type: `void`

RPC name: setsshautomode

Overview:

Set the SSH auto mode for all hosts in the pool

Signature:

```
void set_ssh_auto_mode (session ref session_ref, pool ref self, bool value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`value`	The SSH auto mode for all hosts in the pool，when set to true, SSH to normally be disabled and SSH to be enabled only in case of emergency for example, xapi is down

Minimum Role: pool-admin

Return Type: `void`

RPC name: setsshenabledtimeout

Overview:

Set the SSH enabled timeout for all hosts in the pool

Signature:

```
void set_ssh_enabled_timeout (session ref session_ref, pool ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`int`	`value`	The SSH enabled timeout in seconds. (0 means no timeout, max 2 days)

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setsuspendimageSR

Overview:

Set the suspendimageSR field of the given pool.

Signature:

```
void set_suspend_image_SR (session ref session_ref, pool ref self, SR ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given pool.

Signature:

```
void set_tags (session ref session_ref, pool ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: settelemetrynextcollection

Overview:

Set the timestamp for the next telemetry data collection.

Signature:

```
void set_telemetry_next_collection (session ref session_ref, pool ref
self, datetime value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`datetime`	`value`	The earliest timestamp (in UTC) when the next round of telemetry collection can be carried out.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setueficertificates

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the UEFI certificates for a pool and all its hosts. Deprecated: use setcustomueficertificates instead

Signature:

```
void set_uefi_certificates (session ref session_ref, pool ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`string`	`value`	The certificates to apply to the pool and its hosts

Minimum Role: pool-admin

Return Type: `void`

RPC name: setupdatesyncenabled

Overview:

enable or disable periodic update synchronization depending on the value

Signature:

```
void set_update_sync_enabled (session ref session_ref, pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`value`	true - enable periodic update synchronization, false - disable it

Minimum Role: pool-operator

Return Type: `void`

RPC name: setvmconsoleidletimeout

Overview:

Set the vmconsoleidletimeout field of the given pool.

Signature:

```
void set_vm_console_idle_timeout (session ref session_ref, pool ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setvswitchcontroller

This message is deprecated.

Overview:

Set the IP address of the vswitch controller.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_vswitch_controller (session ref session_ref, string address)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`address`	IP address of the vswitch controller.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setwlbenabled

Overview:

Set the wlbenabled field of the given pool.

Signature:

```
void set_wlb_enabled (session ref session_ref, pool ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: client-cert

Return Type: `void`

RPC name: setwlbverifycert

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the wlbverifycert field of the given pool.

Signature:

```
void set_wlb_verify_cert (session ref session_ref, pool ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: syncdatabase

Overview:

Forcibly synchronise the database now

Signature:

```
void sync_database (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: syncupdates

Overview:

Sync with the enabled repository

Signature:

```
string sync_updates (session ref session_ref, pool ref self, bool force,
string token, string token_id, string username, string password)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`bool`	`force`	If true local mirroring repo will be removed before syncing
`string`	`token`	The token for repository client authentication
`string`	`token_id`	The ID of the token
`string`	`username`	The username of the remote pool
`string`	`password`	The password of the remote pool

Minimum Role: client-cert

Return Type: `string`

The SHA256 hash of updateinfo.xml.gz

RPC name: testarchivetarget

Overview:

This call tests if a location is valid

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string test_archive_target (session ref session_ref, pool ref self,
(string -> string) map config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	Reference to the pool
`(string -> string) map`	`config`	Location config settings to test

Minimum Role: pool-operator

Return Type: `string`

An XMLRPC result

RPC name: uninstallcacertificate

This message is deprecated.

Overview:

Remove a poolwide TLS CA certificate.

Signature:

```
void uninstall_ca_certificate (session ref session_ref, string name, bool
force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name`	The certificate name

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`force`	Remove the DB entry even if the file is non-existent

Minimum Role: client-cert

Return Type: `void`

RPC name: uninstalltrustedcertificate

Overview:

Uninstall a trusted TLS certificate, poolwide.

Signature:

```
void uninstall_trusted_certificate (session ref session_ref, pool ref
self, Certificate ref certificate)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool ref`	`self`	The pool
`Certificate ref`	`certificate`	The reference of the trusted certificate to be uninstalled

Minimum Role: client-cert

Return Type: `void`

Class: poolpatch

This class is deprecated.

Poolwide patches

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: poolpatch

FIELD	TYPE	QUALIFIER	DESCRIPTION
`after_apply_guidance`	`after_apply_guidance set`	*RO/runtime*	**Deprecated.** What the client should do after this patch has been applied.
`host_patches`	`host_patch ref set`	*RO/runtime*	**Deprecated.** This hosts this patch is applied to.
`name_description`	`string`	*RO/constructor*	**Deprecated.** a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	**Deprecated.** a human-readable name
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`pool_applied`	`bool`	*RO/runtime*	**Deprecated.** This patch should be applied across the entire pool
`pool_update`	`pool_update ref`	*RO/constructor*	**Deprecated.** A reference to the associated pool_update object
`size`	`int`	*RO/runtime*	**Deprecated.** Size of the patch
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`version`	`string`	*RO/constructor*	**Deprecated.** Patch version number

RPCs associated with class: poolpatch

RPC name: addtootherconfig

This message is deprecated.

Overview:

Add the given keyvalue pair to the otherconfig field of the given poolpatch.

Signature:

```
void add_to_other_config (session ref session_ref, pool_patch ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: apply

This message is deprecated.

Overview:

Apply the selected patch to a host and return its output

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string apply (session ref session_ref, pool_patch ref self, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to apply
`host ref`	`host`	The host to apply the patch too

Minimum Role: pool-operator

Return Type: `string`

the output of the patch application process

RPC name: clean

This message is deprecated.

Overview:

Removes the patch's files from the server

Signature:

```
void clean (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to clean up

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: cleanonhost

This message is deprecated.

Overview:

Removes the patch's files from the specified host

Signature:

```
void clean_on_host (session ref session_ref, pool_patch ref self, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to clean up
`host ref`	`host`	The host on which to clean the patch

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

This message is deprecated.

Overview:

Removes the patch's files from all hosts in the pool, and removes the database entries. Only works on unapplied patches.

Signature:

```
void destroy (session ref session_ref, pool_patch ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getafterapplyguidance

This message is deprecated.

Overview:

Get the afterapplyguidance field of the given poolpatch.

Signature:

```
after_apply_guidance set get_after_apply_guidance (session ref
session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `after_apply_guidance set`

value of the field

RPC name: getall

This message is deprecated.

Overview:

Return a list of all the poolpatchs known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
pool_patch ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `pool_patch ref set`

references to all objects

RPC name: getallrecords

This message is deprecated.

Overview:

Return a map of poolpatch references to poolpatch records for all poolpatchs known to the system.

Signature:

```
(pool_patch ref -> pool_patch record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(pool_patch ref -> pool_patch record) map`

records of all objects

RPC name: getbynamelabel

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Get all the poolpatch instances with the given label.

Signature:

```
pool_patch ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `pool_patch ref set`

references to objects with matching names

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the poolpatch instance with the specified UUID.

Signature:

```
pool_patch ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `pool_patch ref`

reference to the object

RPC name: gethostpatches

This message is deprecated.

Overview:

Get the hostpatches field of the given poolpatch.

Signature:

```
host_patch ref set get_host_patches (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host_patch ref set`

value of the field

RPC name: getnamedescription

This message is deprecated.

Overview:

Get the name/description field of the given poolpatch.

Signature:

```
string get_name_description (session ref session_ref, pool_patch ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

This message is deprecated.

Overview:

Get the name/label field of the given poolpatch.

Signature:

```
string get_name_label (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the otherconfig field of the given poolpatch.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpoolapplied

This message is deprecated.

Overview:

Get the poolapplied field of the given poolpatch.

Signature:

```
bool get_pool_applied (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getpoolupdate

This message is deprecated.

Overview:

Get the poolupdate field of the given poolpatch.

Signature:

```
pool_update ref get_pool_update (session ref session_ref, pool_patch ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update ref`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given poolpatch.

Signature:

```
pool_patch record get_record (session ref session_ref, pool_patch ref
self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_patch record`

all fields from the object

RPC name: getsize

This message is deprecated.

Overview:

Get the size field of the given poolpatch.

Signature:

```
int get_size (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the uuid field of the given poolpatch.

Signature:

```
string get_uuid (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getversion

This message is deprecated.

Overview:

Get the version field of the given poolpatch.

Signature:

```
string get_version (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: poolapply

This message is deprecated.

Overview:

Apply the selected patch to all hosts in the pool and return a map of hostref > patch output

Signature:

```
void pool_apply (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch to apply

Minimum Role: pool-operator

Return Type: `void`

RPC name: poolclean

This message is deprecated.

Overview:

Removes the patch's files from all hosts in the pool, but does not remove the database entries

Signature:

```
void pool_clean (session ref session_ref, pool_patch ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool_patch ref`	`self`	The patch to clean up

Minimum Role: pool-operator

Return Type: `void`

RPC name: precheck

This message is deprecated.

Overview:

Execute the precheck stage of the selected patch on a host and return its output

Signature:

```
string precheck (session ref session_ref, pool_patch ref self, host ref
host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	The patch whose prechecks will be run
`host ref`	`host`	The host to run the prechecks on

Minimum Role: pool-operator

Return Type: `string`

the output of the patch prechecks

RPC name: removefromotherconfig

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given poolpatch. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, pool_patch ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given poolpatch.

Signature:

```
void set_other_config (session ref session_ref, pool_patch ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_patch ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: poolupdate

Poolwide updates to the host software

Fields for class: poolupdate

FIELD	TYPE	QUALIFIER	DESCRIPTION
`after_apply_guidance`	`update_after_apply_guidance set`	*RO/constructor*	What the client should do after this update has been applied.
`enforce_homogeneity`	`bool`	*RO/constructor*	Flag - if true, all hosts in a pool must apply this update
`hosts`	`host ref set`	*RO/runtime*	The hosts that have applied this update.
`installation_size`	`int`	*RO/constructor*	Size of the update in bytes
`key`	`string`	*RO/constructor*	GPG key of the update
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vdi`	`VDI ref`	*RO/constructor*	VDI the update was uploaded to
`version`	`string`	*RO/constructor*	Update version number

RPCs associated with class: poolupdate

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given poolupdate.

Signature:

```
void add_to_other_config (session ref session_ref, pool_update ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: apply

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Apply the selected update to a host

Signature:

```
void apply (session ref session_ref, pool_update ref self, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update to apply
`host ref`	`host`	The host to apply the update to.

Minimum Role: pool-operator

Return Type: `void`

RPC name: destroy

Overview:

Removes the database entry. Only works on unapplied update.

Signature:

```
void destroy (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update to destroy

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getafterapplyguidance

Overview:

Get the afterapplyguidance field of the given poolupdate.

Signature:

```
update_after_apply_guidance set get_after_apply_guidance (session ref
session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_after_apply_guidance set`

value of the field

RPC name: getall

Overview:

Return a list of all the poolupdates known to the system.

Signature:

```
pool_update ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `pool_update ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of poolupdate references to poolupdate records for all poolupdates known to the system.

Signature:

```
(pool_update ref -> pool_update record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(pool_update ref -> pool_update record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the poolupdate instances with the given label.

Signature:

```
pool_update ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `pool_update ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the poolupdate instance with the specified UUID.

Signature:

```
pool_update ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `pool_update ref`

reference to the object

RPC name: getenforcehomogeneity

Overview:

Get the enforcehomogeneity field of the given poolupdate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_enforce_homogeneity (session ref session_ref, pool_update ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gethosts

Overview:

Get the hosts field of the given poolupdate.

Signature:

```
host ref set get_hosts (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getinstallationsize

Overview:

Get the installationsize field of the given poolupdate.

Signature:

```
int get_installation_size (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getkey

Overview:

Get the key field of the given poolupdate.

Signature:

```
string get_key (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given poolupdate.

Signature:

```
string get_name_description (session ref session_ref, pool_update ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given poolupdate.

Signature:

```
string get_name_label (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given poolupdate.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given poolupdate.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
pool_update record get_record (session ref session_ref, pool_update ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool_update record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given poolupdate.

Signature:

```
string get_uuid (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getvdi

Overview:

Get the vdi field of the given poolupdate.

Signature:

```
VDI ref get_vdi (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getversion

Overview:

Get the version field of the given poolupdate.

Signature:

```
string get_version (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: introduce

Overview:

Introduce update VDI

Signature:

```
pool_update ref introduce (session ref session_ref, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI which contains a software update.

Minimum Role: pool-operator

Return Type: `pool_update ref`

the introduced pool update

RPC name: poolapply

Overview:

Apply the selected update to all hosts in the pool

Signature:

```
void pool_apply (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`pool_update ref`	`self`	The update to apply

Minimum Role: pool-operator

Return Type: `void`

RPC name: poolclean

Overview:

Removes the update's files from all hosts in the pool, but does not revert the update

Signature:

```
void pool_clean (session ref session_ref, pool_update ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update to clean up

Minimum Role: pool-operator

Return Type: `void`

RPC name: precheck

Overview:

Execute the precheck stage of the selected update on a host

Signature:

```
livepatch_status precheck (session ref session_ref, pool_update ref self,
host ref host)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	The update whose prechecks will be run
`host ref`	`host`	The host to run the prechecks on.

Minimum Role: pool-operator

Return Type: `livepatch_status`

The precheck pool update

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given poolupdate. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, pool_update ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given poolupdate.

Signature:

```
void set_other_config (session ref session_ref, pool_update ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`pool_update ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: proberesult

A set of properties that describe one result element of SR.probe. Result elements and properties can change dynamically based on changes to the the SR.probe inputparameters or the target.

Fields for class: proberesult

FIELD	TYPE	QUALIFIER	DESCRIPTION
`complete`	`bool`	*RO/runtime*	True if this configuration is complete and can be used to call SR.create. False if it requires further iterative calls to SR.probe, to potentially narrow down on a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			configuration that can be used.
`configuration`	`(string -> string) map`	*RO/runtime*	Plugin-specific configuration which describes where and how to locate the storage repository. This may include the physical block device name, a remote NFS server and path or an RBD storage pool.
`extra_info`	`(string -> string) map`	*RO/runtime*	Additional plugin-specific information about this configuration, that might be of use for an API user. This can for example include the LUN or the WWPN.
`sr`	`sr_stat record option`	*RO/runtime*	Existing SR found for this configuration

RPCs associated with class: proberesult

Class proberesult has no RPCs associated with it.

Class: PUSB

A physical USB device

Fields for class: PUSB

FIELD	TYPE	QUALIFIER	DESCRIPTION
`description`	`string`	*RO/constructor*	USB device description

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	Physical machine that owns the USB device
`other_config`	`(string -> string) map`	*RW*	additional configuration
`passthrough_enabled`	`bool`	*RO/runtime*	enabled for passthrough
`path`	`string`	*RO/constructor*	port path of USB device
`product_desc`	`string`	*RO/constructor*	product description of the USB device
`product_id`	`string`	*RO/constructor*	product id of the USB device
`serial`	`string`	*RO/constructor*	serial of the USB device
`speed`	`float`	*RO/constructor*	USB device speed
`USB_group`	`USB_group ref`	*RO/constructor*	USB group the PUSB is contained in
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_desc`	`string`	*RO/constructor*	vendor description of the USB device
`vendor_id`	`string`	*RO/constructor*	vendor id of the USB device
`version`	`string`	*RO/constructor*	USB device version

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: PUSB

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given PUSB.

Signature:

```
void add_to_other_config (session ref session_ref, PUSB ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PUSBs known to the system.

Signature:

```
PUSB ref set get_all (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PUSB ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PUSB references to PUSB records for all PUSBs known to the system.

Signature:

```
(PUSB ref -> PUSB record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PUSB ref -> PUSB record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PUSB instance with the specified UUID.

Signature:

```
PUSB ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PUSB ref`

reference to the object

RPC name: getdescription

Overview:

Get the description field of the given PUSB.

Signature:

```
string get_description (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethost

Overview:

Get the host field of the given PUSB.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref get_host (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given PUSB.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, PUSB ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpassthroughenabled

Overview:

Get the passthroughenabled field of the given PUSB.

Signature:

```
bool get_passthrough_enabled (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getpath

Overview:

Get the path field of the given PUSB.

Signature:

```
string get_path (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getproductdesc

Overview:

Get the productdesc field of the given PUSB.

Signature:

```
string get_product_desc (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getproductid

Overview:

Get the productid field of the given PUSB.

Signature:

```
string get_product_id (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PUSB.

Signature:

```
PUSB record get_record (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PUSB record`

all fields from the object

RPC name: getserial

Overview:

Get the serial field of the given PUSB.

Signature:

```
string get_serial (session ref session_ref, PUSB ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getspeed

Overview:

Get the speed field of the given PUSB.

Signature:

```
float get_speed (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getUSBgroup

Overview:

Get the USBgroup field of the given PUSB.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
USB_group ref get_USB_group (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `USB_group ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PUSB.

Signature:

```
string get_uuid (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendordesc

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the vendordesc field of the given PUSB.

Signature:

```
string get_vendor_desc (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorid

Overview:

Get the vendorid field of the given PUSB.

Signature:

```
string get_vendor_id (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getversion

Overview:

Get the version field of the given PUSB.

Signature:

```
string get_version (session ref session_ref, PUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given PUSB. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, PUSB ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: scan

Overview:

Signature:

```
void scan (session ref session_ref, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given PUSB.

Signature:

```
void set_other_config (session ref session_ref, PUSB ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: setpassthroughenabled

Overview:

Signature:

```
void set_passthrough_enabled (session ref session_ref, PUSB ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PUSB ref`	`self`	this PUSB
`bool`	`value`	passthrough is enabled when true and disabled with false

Minimum Role: pool-admin

Return Type: `void`

Class: PVScachestorage

Describes the storage that is available to a PVS site for caching purposes

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: PVScachestorage

FIELD	TYPE	QUALIFIER	DESCRIPTION
`host`	`host ref`	*RO/constructor*	The host on which this object defines PVS cache storage
`site`	`PVS_site ref`	*RO/constructor*	The PVS_site for which this object defines the storage
`size`	`int`	*RO/constructor*	The size of the cache VDI (in bytes)
`SR`	`SR ref`	*RO/constructor*	SR providing storage for the PVS cache
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VDI`	`VDI ref`	*RO/runtime*	The VDI used for caching

RPCs associated with class: PVScachestorage

RPC name: create

Overview:

Create a new PVScachestorage instance, and return its handle.

Signature:

```
PVS_cache_storage ref create (session ref session_ref, PVS_cache_storage record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`PVS_cache_storage record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `PVS_cache_storage ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified PVScachestorage instance.

Signature:

```
void destroy (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the PVScachestorages known to the system.

Signature:

```
PVS_cache_storage ref set get_all (session ref session_ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PVS_cache_storage ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVScachestorage references to PVScachestorage records for all PVScachestorages known to the system.

Signature:

```
(PVS_cache_storage ref -> PVS_cache_storage record) map get_all_records
(session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_cache_storage ref -> PVS_cache_storage record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PVScachestorage instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_cache_storage ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_cache_storage ref`

reference to the object

RPC name: gethost

Overview:

Get the host field of the given PVScachestorage.

Signature:

```
host ref get_host (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getrecord

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a record containing the current state of the given PVScachestorage.

Signature:

```
PVS_cache_storage record get_record (session ref session_ref,
PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_cache_storage record`

all fields from the object

RPC name: getsite

Overview:

Get the site field of the given PVScachestorage.

Signature:

```
PVS_site ref get_site (session ref session_ref, PVS_cache_storage ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getsize

Overview:

Get the size field of the given PVScachestorage.

Signature:

```
int get_size (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getSR

Overview:

Get the SR field of the given PVScachestorage.

Signature:

```
SR ref get_SR (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVScachestorage.

Signature:

```
string get_uuid (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDI

Overview:

Get the VDI field of the given PVScachestorage.

Signature:

```
VDI ref get_VDI (session ref session_ref, PVS_cache_storage ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_cache_storage ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

Class: PVSproxy

a proxy connects a VM/VIF with a PVS site

Fields for class: PVSproxy

FIELD	TYPE	QUALIFIER	DESCRIPTION
`currently_attached`	`bool`	*RO/runtime*	true = VM is currently proxied
`site`	`PVS_site ref`	*RO/constructor*	PVS site this proxy is part of
`status`	`pvs_proxy_status`	*RO/runtime*	The run-time status of the proxy
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VIF`	`VIF ref`	*RO/constructor*	VIF of the VM using the proxy

RPCs associated with class: PVSproxy

RPC name: create

Overview:

Configure a VM/VIF to use a PVS proxy

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_proxy ref create (session ref session_ref, PVS_site ref site, VIF ref
VIF)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`site`	PVS site that we proxy for
`VIF ref`	`VIF`	VIF for the VM that needs to be proxied

Minimum Role: pool-operator

Return Type: `PVS_proxy ref`

The reference of the created PVS proxy

RPC name: destroy

Overview:

remove or switch off a PVS proxy for this VM

Signature:

```
void destroy (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	this PVS proxy

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getall

Overview:

Return a list of all the PVSproxys known to the system.

Signature:

```
PVS_proxy ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PVS_proxy ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVSproxy references to PVSproxy records for all PVSproxys known to the system.

Signature:

```
(PVS_proxy ref -> PVS_proxy record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_proxy ref -> PVS_proxy record) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PVSproxy instance with the specified UUID.

Signature:

```
PVS_proxy ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_proxy ref`

reference to the object

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given PVSproxy.

Signature:

```
bool get_currently_attached (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PVSproxy.

Signature:

```
PVS_proxy record get_record (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_proxy record`

all fields from the object

RPC name: getsite

Overview:

Get the site field of the given PVSproxy.

Signature:

```
PVS_site ref get_site (session ref session_ref, PVS_proxy ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site ref`

value of the field

RPC name: getstatus

Overview:

Get the status field of the given PVSproxy.

Signature:

```
pvs_proxy_status get_status (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pvs_proxy_status`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVSproxy.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVIF

Overview:

Get the VIF field of the given PVSproxy.

Signature:

```
VIF ref get_VIF (session ref session_ref, PVS_proxy ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_proxy ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF ref`

value of the field

Class: PVSserver

individual machine serving provisioning block data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: PVSserver

FIELD	TYPE	QUALIFIER	DESCRIPTION
`addresses`	`string set`	*RO/constructor*	IPv4 addresses of this server
`first_port`	`int`	*RO/constructor*	First UDP port accepted by this server
`last_port`	`int`	*RO/constructor*	Last UDP port accepted by this server
`site`	`PVS_site ref`	*RO/constructor*	PVS site this server is part of
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PVSserver

RPC name: forget

Overview:

forget a PVS server

Signature:

```
void forget (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	this PVS server

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: getaddresses

Overview:

Get the addresses field of the given PVSserver.

Signature:

```
string set get_addresses (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getall

Overview:

Return a list of all the PVSservers known to the system.

Signature:

```
PVS_server ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `PVS_server ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVSserver references to PVSserver records for all PVSservers known to the system.

Signature:

```
(PVS_server ref -> PVS_server record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_server ref -> PVS_server record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the PVSserver instance with the specified UUID.

Signature:

```
PVS_server ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_server ref`

reference to the object

RPC name: getfirstport

Overview:

Get the firstport field of the given PVSserver.

Signature:

```
int get_first_port (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getlastport

Overview:

Get the lastport field of the given PVSserver.

Signature:

```
int get_last_port (session ref session_ref, PVS_server ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PVSserver.

Signature:

```
PVS_server record get_record (session ref session_ref, PVS_server ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_server record`

all fields from the object

RPC name: getsite

Overview:

Get the site field of the given PVSserver.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
PVS_site ref get_site (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVSserver.

Signature:

```
string get_uuid (session ref session_ref, PVS_server ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_server ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: introduce

Overview:

introduce new PVS server

Signature:

```
PVS_server ref introduce (session ref session_ref, string set addresses,
int first_port, int last_port, PVS_site ref site)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string set`	`addresses`	IPv4/IPv6 addresses of the server
`int`	`first_port`	first UDP port accepted by this server
`int`	`last_port`	last UDP port accepted by this server
`PVS_site ref`	`site`	PVS site this server is a part of

Minimum Role: pool-operator

Return Type: `PVS_server ref`

the new PVS server

Class: PVSsite

machines serving blocks of data for provisioning VMs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: PVSsite

FIELD	TYPE	QUALIFIER	DESCRIPTION
`cache_storage`	`PVS_cache_storage ref set`	*RO/runtime*	The SR used by PVS proxy for the cache
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`proxies`	`PVS_proxy ref set`	*RO/runtime*	The set of proxies associated with the site
`PVS_uuid`	`string`	*RO/constructor*	Unique identifier of the PVS site, as configured in PVS
`servers`	`PVS_server ref set`	*RO/runtime*	The set of PVS servers in the site
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: PVSsite

RPC name: forget

Overview:

Remove a site's meta data

Signature:

```
void forget (session ref session_ref, PVS_site ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	this PVS site

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `PVS_SITE_CONTAINS_RUNNING_PROXIES`, `PVS_SITE_CONTAINS_SERVERS`

RPC name: getall

Overview:

Return a list of all the PVSsites known to the system.

Signature:

```
PVS_site ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `PVS_site ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of PVSsite references to PVSsite records for all PVSsites known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(PVS_site ref -> PVS_site record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(PVS_site ref -> PVS_site record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the PVSsite instances with the given label.

Signature:

```
PVS_site ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `PVS_site ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a reference to the PVSsite instance with the specified UUID.

Signature:

```
PVS_site ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `PVS_site ref`

reference to the object

RPC name: getcachestorage

Overview:

Get the cachestorage field of the given PVSsite.

Signature:

```
PVS_cache_storage ref set get_cache_storage (session ref session_ref,
PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_cache_storage ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given PVSsite.

Signature:

```
string get_name_description (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given PVSsite.

Signature:

```
string get_name_label (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getproxies

Overview:

Get the proxies field of the given PVSsite.

Signature:

```
PVS_proxy ref set get_proxies (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_proxy ref set`

value of the field

RPC name: getPVSuuid

Overview:

Get the PVSuuid field of the given PVSsite.

Signature:

```
string get_PVS_uuid (session ref session_ref, PVS_site ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given PVSsite.

Signature:

```
PVS_site record get_record (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_site record`

all fields from the object

RPC name: getservers

Overview:

Get the servers field of the given PVSsite.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_server ref set get_servers (session ref session_ref, PVS_site ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PVS_server ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given PVSsite.

Signature:

```
string get_uuid (session ref session_ref, PVS_site ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: introduce

Overview:

Introduce new PVS site

Signature:

```
PVS_site ref introduce (session ref session_ref, string name_label, string name_description, string PVS_uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	name of the PVS site
`string`	`name_description`	description of the PVS site
`string`	`PVS_uuid`	unique identifier of the PVS site

Minimum Role: pool-operator

Return Type: `PVS_site ref`

the new PVS site

RPC name: setnamedescription

Overview:

Set the name/description field of the given PVSsite.

Signature:

```
void set_name_description (session ref session_ref, PVS_site ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given PVSsite.

Signature:

```
void set_name_label (session ref session_ref, PVS_site ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setPVSuuid

Overview:

Update the PVS UUID of the PVS site

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_PVS_uuid (session ref session_ref, PVS_site ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PVS_site ref`	`self`	this PVS site
`string`	`value`	PVS UUID to be used

Minimum Role: pool-operator

Return Type: `void`

Class: Repository

Repository for updates

Fields for class: Repository

FIELD	TYPE	QUALIFIER	DESCRIPTION
`binary_url`	`string`	*RO/constructor*	Base URL of binary packages in this repository
`certificate`	`string`	*RO/constructor*	The certificate of the host which hosts this repository
`gpgkey_path`	`string`	*RO/constructor*	The file name of the GPG public key of this repository
`hash`	`string`	*RO/runtime*	SHA256 checksum of latest updateinfo.xml.gz in this repository if its 'update' is true

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`origin`	`origin`	*RO/constructor*	The origin of this repository. 'remote' if the origin of the repository is a remote one, 'bundle' if the origin of the repository is a local bundle file, 'remote_pool' if the origin of the repository is a remote pool
`source_url`	`string`	*RO/constructor*	Base URL of source packages in this repository
`up_to_date`	`bool`	*RO/runtime*	**Removed.** True if all hosts in pool is up to date with this repository
`update`	`bool`	*RO/constructor*	True if updateinfo.xml in this repository needs to be parsed
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: Repository

RPC name: forget

Overview:

Remove the repository record from the database

Signature:

```
void forget (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	The repository to be removed from the database

Minimum Role: client-cert

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the Repositorys known to the system.

Signature:

```
Repository ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `Repository ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of Repository references to Repository records for all Repositorys known to the system.

Signature:

```
(Repository ref -> Repository record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(Repository ref -> Repository record) map`

records of all objects

RPC name: getbinaryurl

Overview:

Get the binaryurl field of the given Repository.

Signature:

```
string get_binary_url (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbynamelabel

Overview:

Get all the Repository instances with the given label.

Signature:

```
Repository ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `Repository ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the Repository instance with the specified UUID.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
Repository ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `Repository ref`

reference to the object

RPC name: getcertificate

Overview:

Get the certificate field of the given Repository.

Signature:

```
string get_certificate (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getgpgkeypath

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the gpgkeypath field of the given Repository.

Signature:

```
string get_gpgkey_path (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethash

Overview:

Get the hash field of the given Repository.

Signature:

```
string get_hash (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnamedescription

Overview:

Get the name/description field of the given Repository.

Signature:

```
string get_name_description (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given Repository.

Signature:

```
string get_name_label (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getorigin

Overview:

Get the origin field of the given Repository.

Signature:

```
origin get_origin (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `origin`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given Repository.

Signature:

```
Repository record get_record (session ref session_ref, Repository ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `Repository record`

all fields from the object

RPC name: getsourceurl

Overview:

Get the sourceurl field of the given Repository.

Signature:

```
string get_source_url (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuptodate

This message is removed.

Overview:

Get the uptodate field of the given Repository.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_up_to_date (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getupdate

Overview:

Get the update field of the given Repository.

Signature:

```
bool get_update (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the uuid field of the given Repository.

Signature:

```
string get_uuid (session ref session_ref, Repository ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: introduce

Overview:

Add the configuration for a new remote repository

Signature:

```
Repository ref introduce (session ref session_ref, string name_label,
string name_description, string binary_url, string source_url, bool
update, string gpgkey_path)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	The name of the repository
`string`	`name_description`	The description of the repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`binary_url`	Base URL of binary packages in this repository
`string`	`source_url`	Base URL of source packages in this repository
`bool`	`update`	True if the repository is an update repository. This means that updateinfo.xml will be parsed
`string`	`gpgkey_path`	The GPG public key file name

Minimum Role: client-cert

Return Type: `Repository ref`

The ref of the created repository record.

RPC name: introducebundle

Overview:

Add the configuration for a new bundle repository

Signature:

```
Repository ref introduce_bundle (session ref session_ref, string
name_label, string name_description)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	The name of the repository
`string`	`name_description`	The description of the repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: client-cert

Return Type: `Repository ref`

The ref of the created repository record.

RPC name: introduceremotepool

Overview:

Add the configuration for a new remote pool repository

Signature:

```
Repository ref introduce_remote_pool (session ref session_ref, string name_label, string name_description, string binary_url, string certificate)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	The name of the repository
`string`	`name_description`	The description of the repository
`string`	`binary_url`	Base URL of binary packages in the local repository of this remote pool in https://<coordinator-ip>/repository/ enabled format
`string`	`certificate`	The host certificate of the coordinator of the remote pool

Minimum Role: client-cert

Return Type: `Repository ref`

The ref of the created repository record.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setgpgkeypath

Overview:

Set the file name of the GPG public key of the repository

Signature:

```
void set_gpgkey_path (session ref session_ref, Repository ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	The repository
`string`	`value`	The file name of the GPG public key of the repository

Minimum Role: client-cert

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given Repository.

Signature:

```
void set_name_description (session ref session_ref, Repository ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`Repository ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: client-cert

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given Repository.

Signature:

```
void set_name_label (session ref session_ref, Repository ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`Repository ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: client-cert

Return Type: `void`

Class: role

A set of permissions associated with a subject

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: role

FIELD	TYPE	QUALIFIER	DESCRIPTION
`is_internal`	`bool`	*RO/runtime*	Indicates whether the role is only to be assigned internally by xapi, or can be used by clients
`name_description`	`string`	*RO/constructor*	what this role is for
`name_label`	`string`	*RO/constructor*	a short user-friendly name for the role
`subroles`	`role ref set`	*RO/constructor*	a list of pointers to other roles or permissions
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: role

RPC name: getall

Overview:

Return a list of all the roles known to the system.

Signature:

```
role ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `role ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

references to all objects

RPC name: getallrecords

Overview:

Return a map of role references to role records for all roles known to the system.

Signature:

```
(role ref -> role record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(role ref -> role record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the role instances with the given label.

Signature:

```
role ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `role ref set`

references to objects with matching names

RPC name: getbypermission

Overview:

This call returns a list of roles given a permission

Signature:

```
role ref set get_by_permission (session ref session_ref, role ref
permission)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`permission`	a reference to a permission

Minimum Role: read-only

Return Type: `role ref set`

a list of references to roles

RPC name: getbypermissionnamelabel

Overview:

This call returns a list of roles given a permission name

Signature:

```
role ref set get_by_permission_name_label (session ref session_ref, string
label)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	The short friendly name of the role

Minimum Role: read-only

Return Type: `role ref set`

a list of references to roles

RPC name: getbyuuid

Overview:

Get a reference to the role instance with the specified UUID.

Signature:

```
role ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `role ref`

reference to the object

RPC name: getisinternal

Overview:

Get the isinternal field of the given role.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_is_internal (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given role.

Signature:

```
string get_name_description (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the name/label field of the given role.

Signature:

```
string get_name_label (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpermissions

Overview:

This call returns a list of permissions given a role

Signature:

```
role ref set get_permissions (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	a reference to a role

Minimum Role: read-only

Return Type: `role ref set`

a list of permissions

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpermissionsnamelabel

Overview:

This call returns a list of permission names given a role

Signature:

```
string set get_permissions_name_label (session ref session_ref, role ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	a reference to a role

Minimum Role: read-only

Return Type: `string set`

a list of permission names

RPC name: getrecord

Overview:

Get a record containing the current state of the given role.

Signature:

```
role record get_record (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `role record`

all fields from the object

RPC name: getsubroles

Overview:

Get the subroles field of the given role.

Signature:

```
role ref set get_subroles (session ref session_ref, role ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `role ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given role.

Signature:

```
string get_uuid (session ref session_ref, role ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`role ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

Class: SDNcontroller

Describes the SDN controller that is to connect with the pool

Fields for class: SDNcontroller

FIELD	TYPE	QUALIFIER	DESCRIPTION
`address`	`string`	*RO/constructor*	IP address of the controller
`port`	`int`	*RO/constructor*	TCP port of the controller
`protocol`	`sdn_controller_protocol`	*RO/constructor*	Protocol to connect with SDN controller
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: SDNcontroller

RPC name: forget

Overview:

Remove the OVS manager of the pool and destroy the db record.

Signature:

```
void forget (session ref session_ref, SDN_controller ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	this SDN controller

Minimum Role: pool-operator

Return Type: `void`

RPC name: getaddress

Overview:

Get the address field of the given SDNcontroller.

Signature:

```
string get_address (session ref session_ref, SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getall

Overview:

Return a list of all the SDNcontrollers known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SDN_controller ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `SDN_controller ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of SDNcontroller references to SDNcontroller records for all SDNcontrollers known to the system.

Signature:

```
(SDN_controller ref -> SDN_controller record) map get_all_records (session
ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(SDN_controller ref -> SDN_controller record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the SDNcontroller instance with the specified UUID.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
SDN_controller ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `SDN_controller ref`

reference to the object

RPC name: getport

Overview:

Get the port field of the given SDNcontroller.

Signature:

```
int get_port (session ref session_ref, SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getprotocol

Overview:

Get the protocol field of the given SDNcontroller.

Signature:

```
sdn_controller_protocol get_protocol (session ref session_ref,
SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `sdn_controller_protocol`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given SDNcontroller.

Signature:

```
SDN_controller record get_record (session ref session_ref, SDN_controller
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `SDN_controller record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given SDNcontroller.

Signature:

```
string get_uuid (session ref session_ref, SDN_controller ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SDN_controller ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: introduce

Overview:

Introduce an SDN controller to the pool.

Signature:

```
SDN_controller ref introduce (session ref session_ref,
sdn_controller_protocol protocol, string address, int port)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`sdn_controller_protocol`	`protocol`	Protocol to connect with the controller.
`string`	`address`	IP address of the controller.
`int`	`port`	TCP port of the controller.

Minimum Role: pool-operator

Return Type: `SDN_controller ref`

the introduced SDN controller

Class: secret

A secret

Fields for class: secret

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	other_config
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`value`	`string`	*RW*	the secret

RPCs associated with class: secret

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given secret.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_other_config (session ref session_ref, secret ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new secret instance, and return its handle.

Signature:

```
secret ref create (session ref session_ref, secret record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `secret ref`

reference to the newly created object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Destroy the specified secret instance.

Signature:

```
void destroy (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the secrets known to the system.

Signature:

```
secret ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `secret ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of secret references to secret records for all secrets known to the system.

Signature:

```
(secret ref -> secret record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-operator

Return Type: `(secret ref -> secret record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the secret instance with the specified UUID.

Signature:

```
secret ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `secret ref`

reference to the object

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given secret.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given secret.

Signature:

```
secret record get_record (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `secret record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given secret.

Signature:

```
string get_uuid (session ref session_ref, secret ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `string`

value of the field

RPC name: getvalue

Overview:

Get the value field of the given secret.

Signature:

```
string get_value (session ref session_ref, secret ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given secret. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, secret ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the otherconfig field of the given secret.

Signature:

```
void set_other_config (session ref session_ref, secret ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setvalue

Overview:

Set the value field of the given secret.

Signature:

```
void set_value (session ref session_ref, secret ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`secret ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Class: session

A session

Fields for class: session

FIELD	TYPE	QUALIFIER	DESCRIPTION
`auth_user_name`	`string`	*RO/runtime*	the subject name of the user that was externally authenticated. If a session instance has is_local_superuser set, then the value of this field is undefined.
`auth_user_sid`	`string`	*RO/runtime*	the subject identifier of the user that was externally authenticated. If a session instance has is_local_superuser set, then the value of this field is undefined.
`client_certificate`	`bool`	*RO/runtime*	indicates whether this session was authenticated using a client certificate
`is_local_superuser`	`bool`	*RO/runtime*	true iff this session was created using local superuser credentials
`last_active`	`datetime`	*RO/runtime*	Timestamp for last time session was active

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`originator`	`string`	*RO/runtime*	a key string provided by a API user to distinguish itself from other users sharing the same login name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`parent`	`session ref`	*RO/constructor*	references the parent session that created this session
`pool`	`bool`	*RO/runtime*	True if this session relates to a intra-pool login, false otherwise
`rbac_permissions`	`string set`	*RO/constructor*	list with all RBAC permissions for this session
`subject`	`subject ref`	*RO/runtime*	references the subject instance that created the session. If a session instance has is_local_superuser set, then the value of this field is undefined.
`tasks`	`task ref set`	*RO/runtime*	list of tasks created using the current session
`this_host`	`host ref`	*RO/runtime*	Currently connected host
`this_user`	`user ref`	*RO/runtime*	Currently connected user

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`validation_time`	`datetime`	*RO/runtime*	time when session was last validated

RPCs associated with class: session

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given session.

Signature:

```
void add_to_other_config (session ref session_ref, session ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: changepassword

Overview:

Change the account password; if your session is authenticated with root privileges then the oldpwd is validated and the newpwd is set regardless

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void change_password (session ref session_ref, string old_pwd, string
new_pwd)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`old_pwd`	Old password for account
`string`	`new_pwd`	New password for account

Return Type: `void`

RPC name: createfromdbfile

Overview:

Signature:

```
session ref create_from_db_file (session ref session_ref, string filename)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`filename`	Database dump filename.

Return Type: `session ref`

ID of newly created session

RPC name: getallsubjectidentifiers

Overview:

Return a list of all the user subjectidentifiers of all existing sessions

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
string set get_all_subject_identifiers (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `string set`

The list of user subjectidentifiers of all existing sessions

RPC name: getauthusername

Overview:

Get the authusername field of the given session.

Signature:

```
string get_auth_user_name (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getauthusersid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the authusersid field of the given session.

Signature:

```
string get_auth_user_sid (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the session instance with the specified UUID.

Signature:

```
session ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `session ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getclientcertificate

Overview:

Get the clientcertificate field of the given session.

Signature:

```
bool get_client_certificate (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getislocalsuperuser

Overview:

Get the islocalsuperuser field of the given session.

Signature:

```
bool get_is_local_superuser (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getlastactive

Overview:

Get the lastactive field of the given session.

Signature:

```
datetime get_last_active (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getoriginator

Overview:

Get the originator field of the given session.

Signature:

```
string get_originator (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given session.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, session
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getparent

Overview:

Get the parent field of the given session.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
session ref get_parent (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `session ref`

value of the field

RPC name: getpool

Overview:

Get the pool field of the given session.

Signature:

```
bool get_pool (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrbacpermissions

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the rbacpermissions field of the given session.

Signature:

```
string set get_rbac_permissions (session ref session_ref, session ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given session.

Signature:

```
session record get_record (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `session record`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

all fields from the object

RPC name: getsubject

Overview:

Get the subject field of the given session.

Signature:

```
subject ref get_subject (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `subject ref`

value of the field

RPC name: gettasks

Overview:

Get the tasks field of the given session.

Signature:

```
task ref set get_tasks (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `task ref set`

value of the field

RPC name: getthishost

Overview:

Get the thishost field of the given session.

Signature:

```
host ref get_this_host (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getthisuser

Overview:

Get the thisuser field of the given session.

Signature:

```
user ref get_this_user (session ref session_ref, session ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `user ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given session.

Signature:

```
string get_uuid (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvalidationtime

Overview:

Get the validationtime field of the given session.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
datetime get_validation_time (session ref session_ref, session ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: locallogout

Overview:

Log out of local session.

Signature:

```
void local_logout (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: pool-admin

Return Type: `void`

RPC name: loginwithpassword

Overview:

Attempt to authenticate the user, returning a session reference if successful

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
session ref login_with_password (string uname, string pwd, string version,
string originator)
```

Arguments:

TYPE	NAME	DESCRIPTION
`string`	`uname`	Username for login.
`string`	`pwd`	Password for login.
`string`	`version`	Client API version.
`string`	`originator`	Key string for distinguishing different API users sharing the same login name.

Minimum Role: read-only

Return Type: `session ref`

reference of newly created session

Possible Error Codes: `SESSION_AUTHENTICATION_FAILED`, `HOST_IS_SLAVE`

RPC name: logout

Overview:

Log out of a session

Signature:

```
void logout (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: logoutsubjectidentifier

Overview:

Log out all sessions associated to a user subjectidentifier, except the session associated with the context calling this function

Signature:

```
void logout_subject_identifier (session ref session_ref, string
subject_identifier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`subject_identifier`	User subject-identifier of the sessions to be destroyed

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given session. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, session ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given session.

Signature:

```
void set_other_config (session ref session_ref, session ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`session ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: slavelocalloginwithpassword

Overview:

Authenticate locally against a slave in emergency mode. Note the resulting sessions are only good for use on this host.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
session ref slave_local_login_with_password (string uname, string pwd)
```

Arguments:

TYPE	NAME	DESCRIPTION
`string`	`uname`	Username for login.
`string`	`pwd`	Password for login.

Minimum Role: pool-admin

Return Type: `session ref`

ID of newly created session

Class: SM

A storage manager plugin

Fields for class: SM

FIELD	TYPE	QUALIFIER	DESCRIPTION
`capabilities`	`string set`	*RO/runtime*	**Deprecated.** capabilities of the SM plugin
`configuration`	`(string -> string) map`	*RO/runtime*	names and descriptions of device config keys
`copyright`	`string`	*RO/runtime*	Entity which owns the copyright of this plugin
`driver_filename`	`string`	*RO/runtime*	filename of the storage driver
`features`	`(string -> int) map`	*RO/runtime*	capabilities of the SM plugin, with

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			capability version numbers
`name_description`	`string`	*RO/runtime*	a notes field containing human-readable description
`name_label`	`string`	*RO/runtime*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`required_api_version`	`string`	*RO/runtime*	Minimum SM API version required on the server
`required_cluster_stack`	`string set`	*RO/runtime*	The storage plugin requires that one of these cluster stacks is configured and running.
`supported_image_formats`	`image_format_type set`	*RO/runtime*	Image formats supported by the SR: raw, vhd, qcow2
`type`	`string`	*RO/runtime*	SR.type
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor`	`string`	*RO/runtime*	Vendor who created this plugin
`version`	`string`	*RO/runtime*	Version of the plugin

RPCs associated with class: SM

RPC name: addtootherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Add the given keyvalue pair to the otherconfig field of the given SM.

Signature:

```
void add_to_other_config (session ref session_ref, SM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the SMs known to the system.

Signature:

```
SM ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `SM ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

references to all objects

RPC name: getallrecords

Overview:

Return a map of SM references to SM records for all SMs known to the system.

Signature:

```
(SM ref -> SM record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(SM ref -> SM record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the SM instances with the given label.

Signature:

```
SM ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `SM ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the SM instance with the specified UUID.

Signature:

```
SM ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `SM ref`

reference to the object

RPC name: getcapabilities

This message is deprecated.

Overview:

Get the capabilities field of the given SM.

Signature:

```
string set get_capabilities (session ref session_ref, SM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getconfiguration

Overview:

Get the configuration field of the given SM.

Signature:

```
(string -> string) map get_configuration (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcopyright

Overview:

Get the copyright field of the given SM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_copyright (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdriverfilename

Overview:

Get the driverfilename field of the given SM.

Signature:

```
string get_driver_filename (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getfeatures

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the features field of the given SM.

Signature:

```
(string -> int) map get_features (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> int) map`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given SM.

Signature:

```
string get_name_description (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnamelabel

Overview:

Get the name/label field of the given SM.

Signature:

```
string get_name_label (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given SM.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, SM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given SM.

Signature:

```
SM record get_record (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SM record`

all fields from the object

RPC name: getrequiredapiversion

Overview:

Get the requiredapiversion field of the given SM.

Signature:

```
string get_required_api_version (session ref session_ref, SM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrequiredclusterstack

Overview:

Get the requiredclusterstack field of the given SM.

Signature:

```
string set get_required_cluster_stack (session ref session_ref, SM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getsupportedimageformats

Overview:

Get the supportedimageformats field of the given SM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
image_format_type set get_supported_image_formats (session ref
session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `image_format_type set`

value of the field

RPC name: gettype

Overview:

Get the type field of the given SM.

Signature:

```
string get_type (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given SM.

Signature:

```
string get_uuid (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendor

Overview:

Get the vendor field of the given SM.

Signature:

```
string get_vendor (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getversion

Overview:

Get the version field of the given SM.

Signature:

```
string get_version (session ref session_ref, SM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given SM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, SM ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given SM.

Signature:

```
void set_other_config (session ref session_ref, SM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: SR

A storage repository

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: SR

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`storage_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this SR
`clustered`	`bool`	*RO/runtime*	True if the SR is using aggregated local storage
`content_type`	`string`	*RO/constructor*	the type of the SR's content, if required (for example, ISOs)
`current_operations`	`(string -> storage_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`introduced_by`	`DR_task ref`	*RO/runtime*	The disaster recovery task which introduced this SR
`is_tools_sr`	`bool`	*RO/runtime*	True if this is the SR that contains the Tools ISO VDIs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`local_cache_enabled`	`bool`	*RO/runtime*	True if this SR is assigned to be the local cache for its host
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	additional configuration
`PBDs`	`PBD ref set`	*RO/runtime*	describes how particular hosts can see this storage repository
`physical_size`	`int`	*RO/constructor*	total physical size of the repository (in bytes)
`physical_utilisation`	`int`	*RO/runtime*	physical space currently utilised on this storage repository (in bytes). Note that for sparse disk formats, physical_utilisation may be less than virtual_allocation
`shared`	`bool`	*RO/runtime*	true if this SR is (capable of being) shared between multiple hosts
`sm_config`	`(string -> string) map`	*RW*	SM dependent data

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`type`	`string`	*RO/constructor*	type of the storage repository
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VDIs`	`VDI ref set`	*RO/runtime*	all virtual disks known to this storage repository
`virtual_allocation`	`int`	*RO/runtime*	sum of virtual_sizes of all VDIs in this storage repository (in bytes)

RPCs associated with class: SR

RPC name: addtags

Overview:

Add the given value to the tags field of the given SR. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, SR ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`value`	New value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given SR.

Signature:

```
void add_to_other_config (session ref session_ref, SR ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtosmconfig

Overview:

Add the given keyvalue pair to the smconfig field of the given SR.

Signature:

```
void add_to_sm_config (session ref session_ref, SR ref self, string key,
string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: assertcanhosthastatefile

Overview:

Returns successfully if the given SR can host an HA statefile. Otherwise returns an error to explain why not

Signature:

```
void assert_can_host_ha_statefile (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to query

Minimum Role: pool-operator

Return Type: `void`

RPC name: assertsupportsdatabasereplication

Overview:

Returns successfully if the given SR supports database replication. Otherwise returns an error to explain why not.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void assert_supports_database_replication (session ref session_ref, SR ref
sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to query

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new Storage Repository and introduce it into the managed system, creating both SR record and PBD record to attach it to current host with specified deviceconfig parameters

Signature:

```
SR ref create (session ref session_ref, host ref host, (string -> string)
map device_config, int physical_size, string name_label, string
name_description, string type, string content_type, bool shared, (string
-> string) map sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`int`	`physical_size`	The physical size of the new storage repository
`string`	`name_label`	The name of the new storage repository
`string`	`name_description`	The description of the new storage repository
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`string`	`content_type`	The type of the new SRs content, if required (for example, ISOs)
`bool`	`shared`	True if the SR (is capable of) being shared by multiple hosts
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `SR ref`

The reference of the newly created Storage Repository.

Possible Error Codes: `SR_UNKNOWN_DRIVER`

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this SR

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
blob ref create_new_blob (session ref session_ref, SR ref sr, string name,
string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`name`	The name associated with the blob
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: pool-operator

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: destroy

Overview:

Destroy specified SR, removing SRrecord from database and remove SR from disk. In order to affect this operation the appropriate deviceconfig is read from the specified SR's PBD on current host

Signature:

```
void destroy (session ref session_ref, SR ref sr)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `SR_HAS_PBD`

RPC name: disabledatabasereplication

Overview:

Signature:

```
void disable_database_replication (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to which metadata should be no longer replicated

Minimum Role: pool-operator

Return Type: `void`

RPC name: enabledatabasereplication

Overview:

Signature:

```
void enable_database_replication (session ref session_ref, SR ref sr)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to which metadata should be replicated

Minimum Role: pool-operator

Return Type: `void`

RPC name: forget

Overview:

Removing specified SRrecord from database, without attempting to remove SR from disk

Signature:

```
void forget (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to destroy

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `SR_HAS_PBD`

RPC name: forgetdatasourcearchives

Overview:

Forget the recorded statistics related to the specified data source

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void forget_data_source_archives (session ref session_ref, SR ref sr,
string data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`data_source`	The data source whose archives are to be forgotten

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the SRs known to the system.

Signature:

```
SR ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `SR ref set`

references to all objects

RPC name: getallrecords

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return a map of SR references to SR records for all SRs known to the system.

Signature:

```
(SR ref -> SR record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(SR ref -> SR record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given SR.

Signature:

```
storage_operations set get_allowed_operations (session ref session_ref, SR
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `storage_operations set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getblobs

Overview:

Get the blobs field of the given SR.

Signature:

```
(string -> blob ref) map get_blobs (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getbynamelabel

Overview:

Get all the SR instances with the given label.

Signature:

```
SR ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `SR ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the SR instance with the specified UUID.

Signature:

```
SR ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `SR ref`

reference to the object

RPC name: getclustered

Overview:

Get the clustered field of the given SR.

Signature:

```
bool get_clustered (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getcontenttype

Overview:

Get the contenttype field of the given SR.

Signature:

```
string get_content_type (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given SR.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> storage_operations) map get_current_operations (session ref
session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> storage_operations) map`

value of the field

RPC name: getdatasources

Overview:

Signature:

```
data_source record set get_data_sources (session ref session_ref, SR ref
sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to interrogate

Minimum Role: read-only

Return Type: `data_source record set`

A set of data sources

RPC name: getintroducedby

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the introducedby field of the given SR.

Signature:

```
DR_task ref get_introduced_by (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `DR_task ref`

value of the field

RPC name: getistoolssr

Overview:

Get the istoolssr field of the given SR.

Signature:

```
bool get_is_tools_sr (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getlocalcacheenabled

Overview:

Get the localcacheenabled field of the given SR.

Signature:

```
bool get_local_cache_enabled (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given SR.

Signature:

```
string get_name_description (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given SR.

Signature:

```
string get_name_label (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given SR.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, SR ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPBDs

Overview:

Get the PBDs field of the given SR.

Signature:

```
PBD ref set get_PBDs (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PBD ref set`

value of the field

RPC name: getphysicalsize

Overview:

Get the physicalsize field of the given SR.

Signature:

```
int get_physical_size (session ref session_ref, SR ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getphysicalutilisation

Overview:

Get the physicalutilisation field of the given SR.

Signature:

```
int get_physical_utilisation (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given SR.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR record get_record (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR record`

all fields from the object

RPC name: getshared

Overview:

Get the shared field of the given SR.

Signature:

```
bool get_shared (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsmconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the smconfig field of the given SR.

Signature:

```
(string -> string) map get_sm_config (session ref session_ref, SR ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsupportedtypes

Overview:

Return a set of all the SR types supported by the system

Signature:

```
string set get_supported_types (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `string set`

the supported SR types

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettags

Overview:

Get the tags field of the given SR.

Signature:

```
string set get_tags (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettype

Overview:

Get the type field of the given SR.

Signature:

```
string get_type (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given SR.

Signature:

```
string get_uuid (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDIs

Overview:

Get the VDIs field of the given SR.

Signature:

```
VDI ref set get_VDIs (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref set`

value of the field

RPC name: getvirtualallocation

Overview:

Get the virtualallocation field of the given SR.

Signature:

```
int get_virtual_allocation (session ref session_ref, SR ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: introduce

Overview:

Introduce a new Storage Repository into the managed system

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR ref introduce (session ref session_ref, string uuid, string name_label,
string name_description, string type, string content_type, bool shared,
(string -> string) map sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	The uuid assigned to the introduced SR
`string`	`name_label`	The name of the new storage repository
`string`	`name_description`	The description of the new storage repository
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`string`	`content_type`	The type of the new SRs content, if required (for example, ISOs)
`bool`	`shared`	True if the SR (is capable of) being shared by multiple hosts
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `SR ref`

The reference of the newly introduced Storage Repository.

RPC name: make

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Create a new Storage Repository on disk. This call is deprecated: use SR.create instead.

Signature:

```
string make (session ref session_ref, host ref host, (string -> string)
map device_config, int physical_size, string name_label, string
name_description, string type, string content_type, (string -> string) map
sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`int`	`physical_size`	The physical size of the new storage repository
`string`	`name_label`	The name of the new storage repository
`string`	`name_description`	The description of the new storage repository
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`string`	`content_type`	The type of the new SRs content, if required (for example, ISOs)
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `string`

The uuid of the newly created Storage Repository.

RPC name: probe

Overview:

Perform a backendspecific scan, using the given deviceconfig. If the deviceconfig is complete, then this will return a list of the SRs present of this type on the device, if any. If the deviceconfig is partial, then a backendspecific scan will be performed, returning results that will guide the user in improving the deviceconfig.

Signature:

```
string probe (session ref session_ref, host ref host, (string -> string)
map device_config, string type, (string -> string) map sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `string`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

An XML fragment containing the scan results. These are specific to the scan being performed, and the backend.

RPC name: probeext

Overview:

Perform a backendspecific scan, using the given deviceconfig. If the deviceconfig is complete, then this will return a list of the SRs present of this type on the device, if any. If the deviceconfig is partial, then a backendspecific scan will be performed, returning results that will guide the user in improving the deviceconfig.

Signature:

```
probe_result record set probe_ext (session ref session_ref, host ref host,
(string -> string) map device_config, string type, (string -> string) map
sm_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`host ref`	`host`	The host to create/make the SR on
`(string -> string) map`	`device_config`	The device config string that will be passed to backend SR driver
`string`	`type`	The type of the SR; used to specify the SR backend driver to use
`(string -> string) map`	`sm_config`	Storage backend specific configuration options

Minimum Role: pool-operator

Return Type: `probe_result record set`

A set of records containing the scan results.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: querydatasource

Overview:

Query the latest value of the specified data source

Signature:

```
float query_data_source (session ref session_ref, SR ref sr, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`data_source`	The data source to query

Minimum Role: read-only

Return Type: `float`

The latest value, averaged over the last 5 seconds

RPC name: recorddatasource

Overview:

Start recording the specified data source

Signature:

```
void record_data_source (session ref session_ref, SR ref sr, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`SR ref`	`sr`	The SR
`string`	`data_source`	The data source to record

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given SR. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, SR ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromsmconfig

Overview:

Remove the given key and its corresponding value from the smconfig field of the given SR. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_sm_config (session ref session_ref, SR ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given SR. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, SR ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: scan

Overview:

Refreshes the list of VDIs associated with an SR

Signature:

```
void scan (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR to scan

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name description of the SR

Signature:

```
void set_name_description (session ref session_ref, SR ref sr, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`value`	The name description for the SR

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name label of the SR

Signature:

```
void set_name_label (session ref session_ref, SR ref sr, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`string`	`value`	The name label for the SR

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given SR.

Signature:

```
void set_other_config (session ref session_ref, SR ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setphysicalsize

Overview:

Sets the SR's physicalsize field

Signature:

```
void set_physical_size (session ref session_ref, SR ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	The SR to modify
`int`	`value`	The new value of the SR's physical_size

Minimum Role: pool-operator

Return Type: `void`

RPC name: setshared

Overview:

Sets the shared flag on the SR

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_shared (session ref session_ref, SR ref sr, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR
`bool`	`value`	True if the SR is shared

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsmconfig

Overview:

Set the smconfig field of the given SR.

Signature:

```
void set_sm_config (session ref session_ref, SR ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: settags

Overview:

Set the tags field of the given SR.

Signature:

```
void set_tags (session ref session_ref, SR ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: update

Overview:

Refresh the fields on the SR object

Signature:

```
void update (session ref session_ref, SR ref sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`SR ref`	`sr`	The SR whose fields should be refreshed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

Class: srstat

A set of highlevel properties associated with an SR.

Fields for class: srstat

FIELD	TYPE	QUALIFIER	DESCRIPTION
`clustered`	`bool`	*RO/runtime*	Indicates whether the SR uses clustered local storage.
`free_space`	`int`	*RO/runtime*	Number of bytes free on the backing storage (in bytes)
`health`	`sr_health`	*RO/runtime*	The health status of the SR.
`name_description`	`string`	*RO/runtime*	Longer, human-readable description of the SR. Descriptions are generally only displayed by clients when the user is examining SRs in detail.
`name_label`	`string`	*RO/runtime*	Short, human-readable label for the SR.
`total_space`	`int`	*RO/runtime*	Total physical size of the backing storage (in bytes)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string option`	*RO/runtime*	Uuid that uniquely identifies this SR, if one is available.

RPCs associated with class: srstat

Class srstat has no RPCs associated with it.

Class: subject

A user or group that can log in xapi

Fields for class: subject

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RO/constructor*	additional configuration
`roles`	`role ref set`	*RO/runtime*	the roles associated with this subject
`subject_identifier`	`string`	*RO/constructor*	the subject identifier, unique in the external directory service
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: subject

RPC name: addtoroles

Overview:

This call adds a new role to a subject

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_roles (session ref session_ref, subject ref self, role ref
role)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	The subject who we want to add the role to
`role ref`	`role`	The unique role reference

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

Overview:

Create a new subject instance, and return its handle.

Signature:

```
subject ref create (session ref session_ref, subject record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject record`	`args`	All constructor arguments

Minimum Role: pool-admin

Return Type: `subject ref`

reference to the newly created object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Destroy the specified subject instance.

Signature:

```
void destroy (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the subjects known to the system.

Signature:

```
subject ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `subject ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of subject references to subject records for all subjects known to the system.

Signature:

```
(subject ref -> subject record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(subject ref -> subject record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the subject instance with the specified UUID.

Signature:

```
subject ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `subject ref`

reference to the object

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given subject.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpermissionsnamelabel

Overview:

This call returns a list of permission names given a subject

Signature:

```
string set get_permissions_name_label (session ref session_ref, subject ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	The subject whose permissions will be retrieved

Minimum Role: read-only

Return Type: `string set`

a list of permission names

RPC name: getrecord

Overview:

Get a record containing the current state of the given subject.

Signature:

```
subject record get_record (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `subject record`

all fields from the object

RPC name: getroles

Overview:

Get the roles field of the given subject.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
role ref set get_roles (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `role ref set`

value of the field

RPC name: getsubjectidentifier

Overview:

Get the subjectidentifier field of the given subject.

Signature:

```
string get_subject_identifier (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the uuid field of the given subject.

Signature:

```
string get_uuid (session ref session_ref, subject ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromroles

Overview:

This call removes a role from a subject

Signature:

```
void remove_from_roles (session ref session_ref, subject ref self, role
ref role)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`subject ref`	`self`	The subject from whom we want to remove the role
`role ref`	`role`	The unique role reference in the subject's roles field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

Class: task

A longrunning asynchronous task

Fields for class: task

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`task_allowed_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`backtrace`	`string`	*RO/runtime*	Function call trace for debugging.
`created`	`datetime`	*RO/runtime*	Time task was created
`current_operations`	`(string -> task_allowed_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`error_info`	`string set`	*RO/runtime*	if the task has failed, this field contains the set of associated error strings. Undefined otherwise.
`finished`	`datetime`	*RO/runtime*	Time task finished (i.e. succeeded or

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			failed). If task-status is pending, then the value of this field has no meaning
`name_description`	`string`	*RO/runtime*	a notes field containing human-readable description
`name_label`	`string`	*RO/runtime*	a human-readable name
`other_config`	`(string -> string) map`	*RO/runtime*	additional configuration
`progress`	`float`	*RO/runtime*	This field contains the estimated fraction of the task which is complete. This field should not be used to determine whether the task is complete - for this the status field of the task should be used.
`resident_on`	`host ref`	*RO/runtime*	the host on which the task is running
`result`	`string`	*RO/runtime*	if the task has completed successfully, this field contains the result value (either Void or an object reference). Undefined otherwise.
`status`	`task_status_type`	*RO/runtime*	current status of the task

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`subtask_of`	`task ref`	*RO/runtime*	Ref pointing to the task this is a substask of.
`subtasks`	`task ref set`	*RO/runtime*	List pointing to all the substasks.
`type`	`string`	*RO/runtime*	if the task has completed successfully, this field contains the type of the encoded result (i.e. name of the class whose reference is in the result field). Undefined otherwise.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: task

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given task.

Signature:

```
void add_to_other_config (session ref session_ref, task ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Task object to modify

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: read-only

Return Type: `void`

RPC name: cancel

Overview:

Request that a task be cancelled. Note that a task may fail to be cancelled and may complete or fail normally and note that, even when a task does cancel, it might take an arbitrary amount of time.

Signature:

```
void cancel (session ref session_ref, task ref task)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`task`	The task

Minimum Role: read-only

Return Type: `void`

Possible Error Codes: `OPERATION_NOT_ALLOWED`

RPC name: create

Overview:

Create a new task object which must be manually destroyed.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
task ref create (session ref session_ref, string label, string
description)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	short label for the new task
`string`	`description`	longer description for the new task

Minimum Role: read-only

Return Type: `task ref`

The reference of the created task object

RPC name: destroy

Overview:

Destroy the task object

Signature:

```
void destroy (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object

Minimum Role: read-only

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getall

Overview:

Return a list of all the tasks known to the system.

Signature:

```
task ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `task ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of task references to task records for all tasks known to the system.

Signature:

```
(task ref -> task record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(task ref -> task record) map`

records of all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given task.

Signature:

```
task_allowed_operations set get_allowed_operations (session ref
session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task_allowed_operations set`

value of the field

RPC name: getbacktrace

Overview:

Get the backtrace field of the given task.

Signature:

```
string get_backtrace (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getbynamelabel

Overview:

Get all the task instances with the given label.

Signature:

```
task ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `task ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the task instance with the specified UUID.

Signature:

```
task ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `task ref`

reference to the object

RPC name: getcreated

Overview:

Get the created field of the given task.

Signature:

```
datetime get_created (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given task.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> task_allowed_operations) map get_current_operations (session
ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> task_allowed_operations) map`

value of the field

RPC name: geterrorinfo

Overview:

Get the errorinfo field of the given task.

Signature:

```
string set get_error_info (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getfinished

Overview:

Get the finished field of the given task.

Signature:

```
datetime get_finished (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given task.

Signature:

```
string get_name_description (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given task.

Signature:

```
string get_name_label (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given task.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, task ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprogress

Overview:

Get the progress field of the given task.

Signature:

```
float get_progress (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given task.

Signature:

```
task record get_record (session ref session_ref, task ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task record`

all fields from the object

RPC name: getresidenton

Overview:

Get the residenton field of the given task.

Signature:

```
host ref get_resident_on (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getresult

Overview:

Get the result field of the given task.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_result (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getstatus

Overview:

Get the status field of the given task.

Signature:

```
task_status_type get_status (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task_status_type`

value of the field

RPC name: getsubtaskof

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the subtaskof field of the given task.

Signature:

```
task ref get_subtask_of (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task ref`

value of the field

RPC name: getsubtasks

Overview:

Get the subtasks field of the given task.

Signature:

```
task ref set get_subtasks (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `task ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gettype

Overview:

Get the type field of the given task.

Signature:

```
string get_type (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given task.

Signature:

```
string get_uuid (session ref session_ref, task ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given task. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, task ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Task object to modify
`string`	`key`	Key of entry to remove

Minimum Role: read-only

Return Type: `void`

RPC name: seterrorinfo

Overview:

Set the task error info

Signature:

```
void set_error_info (session ref session_ref, task ref self, string set
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`string set`	`value`	Task error info to be set

Minimum Role: read-only

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given task.

Signature:

```
void set_other_config (session ref session_ref, task ref self, (string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Task object to modify
`(string -> string) map`	`value`	New value to set

Minimum Role: read-only

Return Type: `void`

RPC name: setprogress

Overview:

Set the task progress

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_progress (session ref session_ref, task ref self, float value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`float`	`value`	Task progress value to be set

Minimum Role: read-only

Return Type: `void`

RPC name: setresult

Overview:

Set the task result

Signature:

```
void set_result (session ref session_ref, task ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`string`	`value`	Task result to be set

Minimum Role: read-only

Return Type: `void`

RPC name: setstatus

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the task status

Signature:

```
void set_status (session ref session_ref, task ref self, task_status_type
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`task ref`	`self`	Reference to the task object
`task_status_type`	`value`	task status value to be set

Minimum Role: read-only

Return Type: `void`

Class: tunnel

A tunnel for network traffic

Fields for class: tunnel

FIELD	TYPE	QUALIFIER	DESCRIPTION
`access_PIF`	`PIF ref`	*RO/constructor*	The interface through which the tunnel is accessed
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`protocol`	`tunnel_protocol`	*RW*	The protocol used for tunneling (either GRE or VxLAN)
`status`	`(string -> string) map`	*RW*	Status information about the tunnel

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`transport_PIF`	`PIF ref`	*RO/constructor*	The interface used by the tunnel
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: tunnel

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given tunnel.

Signature:

```
void add_to_other_config (session ref session_ref, tunnel ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtostatus

Overview:

Add the given keyvalue pair to the status field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_status (session ref session_ref, tunnel ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a tunnel

Signature:

```
tunnel ref create (session ref session_ref, PIF ref transport_PIF, network
ref network, tunnel_protocol protocol)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`transport_PIF`	PIF which receives the tagged traffic
`network ref`	`network`	Network to receive the tunnelled traffic

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`tunnel_protocol`	`protocol`	Protocol used for the tunnel (GRE or VxLAN)

Minimum Role: pool-operator

Return Type: `tunnel ref`

The reference of the created tunnel object

Possible Error Codes: `OPENVSWITCH_NOT_ACTIVE`, `TRANSPORT_PIF_NOT_CONFIGURED`, `IS_TUNNEL_ACCESS_PIF`

RPC name: destroy

Overview:

Destroy a tunnel

Signature:

```
void destroy (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	tunnel to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getaccessPIF

Overview:

Get the accessPIF field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PIF ref get_access_PIF (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getall

Overview:

Return a list of all the tunnels known to the system.

Signature:

```
tunnel ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `tunnel ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of tunnel references to tunnel records for all tunnels known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(tunnel ref -> tunnel record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(tunnel ref -> tunnel record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the tunnel instance with the specified UUID.

Signature:

```
tunnel ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `tunnel ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given tunnel.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getprotocol

Overview:

Get the protocol field of the given tunnel.

Signature:

```
tunnel_protocol get_protocol (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `tunnel_protocol`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given tunnel.

Signature:

```
tunnel record get_record (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tunnel record`

all fields from the object

RPC name: getstatus

Overview:

Get the status field of the given tunnel.

Signature:

```
(string -> string) map get_status (session ref session_ref, tunnel ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gettransportPIF

Overview:

Get the transportPIF field of the given tunnel.

Signature:

```
PIF ref get_transport_PIF (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, tunnel ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given tunnel. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, tunnel ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removefromstatus

Overview:

Remove the given key and its corresponding value from the status field of the given tunnel. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_status (session ref session_ref, tunnel ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given tunnel.

Signature:

```
void set_other_config (session ref session_ref, tunnel ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`tunnel ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setprotocol

Overview:

Set the protocol field of the given tunnel.

Signature:

```
void set_protocol (session ref session_ref, tunnel ref self,
tunnel_protocol value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`tunnel_protocol`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setstatus

Overview:

Set the status field of the given tunnel.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_status (session ref session_ref, tunnel ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`tunnel ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: USBgroup

A group of compatible USBs across the resource pool

Fields for class: USBgroup

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`PUSBs`	`PUSB ref set`	*RO/runtime*	List of PUSBs in the group
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VUSBs`	`VUSB ref set`	*RO/runtime*	List of VUSBs using the group

RPCs associated with class: USBgroup

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given USBgroup.

Signature:

```
void add_to_other_config (session ref session_ref, USB_group ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

Overview:

Signature:

```
USB_group ref create (session ref session_ref, string name_label, string
name_description, (string -> string) map other_config)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`name_label`	
`string`	`name_description`	
`(string -> string) map`	`other_config`	

Minimum Role: pool-admin

Return Type: `USB_group ref`

The reference of the created USBgroup

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	The USB group to destroy

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the USBgroups known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
USB_group ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `USB_group ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of USBgroup references to USBgroup records for all USBgroups known to the system.

Signature:

```
(USB_group ref -> USB_group record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(USB_group ref -> USB_group record) map`

records of all objects

RPC name: getbynamelabel

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get all the USBgroup instances with the given label.

Signature:

```
USB_group ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `USB_group ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the USBgroup instance with the specified UUID.

Signature:

```
USB_group ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `USB_group ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the object

RPC name: getnamedescription

Overview:

Get the name/description field of the given USBgroup.

Signature:

```
string get_name_description (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given USBgroup.

Signature:

```
string get_name_label (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given USBgroup.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPUSBs

Overview:

Get the PUSBs field of the given USBgroup.

Signature:

```
PUSB ref set get_PUSBs (session ref session_ref, USB_group ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PUSB ref set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given USBgroup.

Signature:

```
USB_group record get_record (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `USB_group record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given USBgroup.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_uuid (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVUSBs

Overview:

Get the VUSBs field of the given USBgroup.

Signature:

```
VUSB ref set get_VUSBs (session ref session_ref, USB_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VUSB ref set`

value of the field

RPC name: removefromotherconfig

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Remove the given key and its corresponding value from the otherconfig field of the given USBgroup. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, USB_group ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given USBgroup.

Signature:

```
void set_name_description (session ref session_ref, USB_group ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given USBgroup.

Signature:

```
void set_name_label (session ref session_ref, USB_group ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given USBgroup.

Signature:

```
void set_other_config (session ref session_ref, USB_group ref self,
(string -> string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`USB_group ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

Class: user

This class is deprecated.

A user of the system

Fields for class: user

FIELD	TYPE	QUALIFIER	DESCRIPTION
`fullname`	`string`	*RW*	**Deprecated.** full name
`other_config`	`(string -> string) map`	*RW*	**Deprecated.** additional configuration
`short_name`	`string`	*RO/constructor*	**Deprecated.** short name (for example, userid)
`uuid`	`string`	*RO/runtime*	**Deprecated.** Unique identifier/object reference

RPCs associated with class: user

RPC name: addtootherconfig

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Add the given keyvalue pair to the otherconfig field of the given user.

Signature:

```
void add_to_other_config (session ref session_ref, user ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

This message is deprecated.

Overview:

Create a new user instance, and return its handle.

Signature:

```
user ref create (session ref session_ref, user record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user record`	`args`	All constructor arguments

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-admin

Return Type: `user ref`

reference to the newly created object

RPC name: destroy

This message is deprecated.

Overview:

Destroy the specified user instance.

Signature:

```
void destroy (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: getbyuuid

This message is deprecated.

Overview:

Get a reference to the user instance with the specified UUID.

Signature:

```
user ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `user ref`

reference to the object

RPC name: getfullname

This message is deprecated.

Overview:

Get the fullname field of the given user.

Signature:

```
string get_fullname (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getotherconfig

This message is deprecated.

Overview:

Get the otherconfig field of the given user.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, user ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

This message is deprecated.

Overview:

Get a record containing the current state of the given user.

Signature:

```
user record get_record (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `user record`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

all fields from the object

RPC name: getshortname

This message is deprecated.

Overview:

Get the shortname field of the given user.

Signature:

```
string get_short_name (session ref session_ref, user ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getuuid

This message is deprecated.

Overview:

Get the uuid field of the given user.

Signature:

```
string get_uuid (session ref session_ref, user ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is deprecated.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given user. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, user ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setfullname

This message is deprecated.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the fullname field of the given user.

Signature:

```
void set_fullname (session ref session_ref, user ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

This message is deprecated.

Overview:

Set the otherconfig field of the given user.

Signature:

```
void set_other_config (session ref session_ref, user ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`user ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

Class: VBD

A virtual block device

Fields for class: VBD

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vbd_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`bootable`	`bool`	*RW*	true if this VBD is bootable
`current_operations`	`(string -> vbd_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`currently_attached`	`bool`	*RO/constructor*	is the device currently attached (erased on reboot)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`device`	`string`	*RO/constructor*	device seen by the guest for example, hda1
`empty`	`bool`	*RO/constructor*	if true this represents an empty drive
`metrics`	`VBD_metrics ref`	*RO/runtime*	**Removed.** metrics associated with this VBD
`mode`	`vbd_mode`	*RO/constructor*	the mode the VBD should be mounted with
`other_config`	`(string -> string) map`	*RW*	additional configuration
`qos_algorithm_params`	`(string -> string) map`	*RW*	parameters for chosen QoS algorithm
`qos_algorithm_type`	`string`	*RW*	QoS algorithm to use
`qos_supported_algorithms`	`string set`	*RO/runtime*	supported QoS algorithms for this VBD
`runtime_properties`	`(string -> string) map`	*RO/runtime*	Device runtime properties
`status_code`	`int`	*RO/runtime*	error/success code associated with last attach-operation (erased on reboot)
`status_detail`	`string`	*RO/runtime*	error/success information associated with last attach-operation

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			status (erased on reboot)
`storage_lock`	`bool`	*RO/runtime*	true if a storage level lock was acquired
`type`	`vbd_type`	*RW*	how the VBD will appear to the guest (for example, disk or CD)
`unpluggable`	`bool`	*RW*	true if this VBD will support hot-unplug
`userdevice`	`string`	*RW*	user-friendly device name for example, 0,1,2,etc.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VDI`	`VDI ref`	*RO/constructor*	the virtual disk
`VM`	`VM ref`	*RO/constructor*	the virtual machine

RPCs associated with class: VBD

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VBD.

Signature:

```
void add_to_other_config (session ref session_ref, VBD ref self, string
key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoqosalgorithmparams

Overview:

Add the given keyvalue pair to the qos/algorithmparams field of the given VBD.

Signature:

```
void add_to_qos_algorithm_params (session ref session_ref, VBD ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: assertattachable

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Throws an error if this VBD could not be attached to this VM if the VM were running. Intended for debugging.

Signature:

```
void assert_attachable (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to query

Minimum Role: vm-admin

Return Type: `void`

RPC name: create

Overview:

Create a new VBD instance, and return its handle.

Signature:

```
VBD ref create (session ref session_ref, VBD record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VBD ref`

reference to the newly created object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Destroy the specified VBD instance.

Signature:

```
void destroy (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: eject

Overview:

Remove the media from the device and leave it empty

Signature:

```
void eject (session ref session_ref, VBD ref vbd)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`vbd`	The vbd representing the CDROM-like device

Minimum Role: vm-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `VBD_NOT_REMOVABLE_MEDIA`, `VBD_IS_EMPTY`

RPC name: getall

Overview:

Return a list of all the VBDs known to the system.

Signature:

```
VBD ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VBD ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VBD references to VBD records for all VBDs known to the system.

Signature:

```
(VBD ref -> VBD record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VBD ref -> VBD record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VBD.

Signature:

```
vbd_operations set get_allowed_operations (session ref session_ref, VBD
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vbd_operations set`

value of the field

RPC name: getbootable

Overview:

Get the bootable field of the given VBD.

Signature:

```
bool get_bootable (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VBD instance with the specified UUID.

Signature:

```
VBD ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VBD ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> vbd_operations) map get_current_operations (session ref
session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vbd_operations) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VBD.

Signature:

```
bool get_currently_attached (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getdevice

Overview:

Get the device field of the given VBD.

Signature:

```
string get_device (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getempty

Overview:

Get the empty field of the given VBD.

Signature:

```
bool get_empty (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getmetrics

This message is removed.

Overview:

Get the metrics field of the given VBD.

Signature:

```
VBD_metrics ref get_metrics (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VBD_metrics ref`

value of the field

RPC name: getmode

Overview:

Get the mode field of the given VBD.

Signature:

```
vbd_mode get_mode (session ref session_ref, VBD ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vbd_mode`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VBD.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmparams

Overview:

Get the qos/algorithmparams field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_qos_algorithm_params (session ref session_ref,
VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmtype

Overview:

Get the qos/algorithmtype field of the given VBD.

Signature:

```
string get_qos_algorithm_type (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getqossupportedalgorithms

Overview:

Get the qos/supportedalgorithms field of the given VBD.

Signature:

```
string set get_qos_supported_algorithms (session ref session_ref, VBD ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VBD.

Signature:

```
VBD record get_record (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VBD record`

all fields from the object

RPC name: getruntimeproperties

Overview:

Get the runtimeproperties field of the given VBD.

Signature:

```
(string -> string) map get_runtime_properties (session ref session_ref,
VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getstatuscode

Overview:

Get the statuscode field of the given VBD.

Signature:

```
int get_status_code (session ref session_ref, VBD ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getstatusdetail

Overview:

Get the statusdetail field of the given VBD.

Signature:

```
string get_status_detail (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getstoragelock

Overview:

Get the storagelock field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_storage_lock (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VBD.

Signature:

```
vbd_type get_type (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vbd_type`

value of the field

RPC name: getunpluggable

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the unpluggable field of the given VBD.

Signature:

```
bool get_unpluggable (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getuserdevice

Overview:

Get the userdevice field of the given VBD.

Signature:

```
string get_userdevice (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given VBD.

Signature:

```
string get_uuid (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVDI

Overview:

Get the VDI field of the given VBD.

Signature:

```
VDI ref get_VDI (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VDI ref`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VBD.

Signature:

```
VM ref get_VM (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: insert

Overview:

Insert new media into the device

Signature:

```
void insert (session ref session_ref, VBD ref vbd, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VBD ref`	`vbd`	The vbd representing the CDROM-like device
`VDI ref`	`vdi`	The new VDI to 'insert'

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VBD_NOT_REMOVABLE_MEDIA`, `VBD_NOT_EMPTY`

RPC name: plug

Overview:

Hotplug the specified VBD, dynamically attaching it to the running VM

Signature:

```
void plug (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to hotplug

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VBD. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_other_config (session ref session_ref, VBD ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromqosalgorithmparams

Overview:

Remove the given key and its corresponding value from the qos/algorithmparams field of the given VBD. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_qos_algorithm_params (session ref session_ref, VBD ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setbootable

Overview:

Set the bootable field of the given VBD.

Signature:

```
void set_bootable (session ref session_ref, VBD ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setmode

Overview:

Sets the mode of the VBD. The powerstate of the VM must be halted.

Signature:

```
void set_mode (session ref session_ref, VBD ref self, vbd_mode value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	Reference to the object
`vbd_mode`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VBD.

Signature:

```
void set_other_config (session ref session_ref, VBD ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmparams

Overview:

Set the qos/algorithmparams field of the given VBD.

Signature:

```
void set_qos_algorithm_params (session ref session_ref, VBD ref self,
(string -> string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmtype

Overview:

Set the qos/algorithmtype field of the given VBD.

Signature:

```
void set_qos_algorithm_type (session ref session_ref, VBD ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: settype

Overview:

Set the type field of the given VBD.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_type (session ref session_ref, VBD ref self, vbd_type value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`vbd_type`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setunpluggable

Overview:

Set the unpluggable field of the given VBD.

Signature:

```
void set_unpluggable (session ref session_ref, VBD ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setuserdevice

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the userdevice field of the given VBD.

Signature:

```
void set_userdevice (session ref session_ref, VBD ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: unplug

Overview:

Hotunplug the specified VBD, dynamically unattaching it from the running VM

Signature:

```
void unplug (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to hot-unplug

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `DEVICE_DETACH_REJECTED`, `DEVICE_ALREADY_DETACHED`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: unplugforce

Overview:

Forcibly unplug the specified VBD

Signature:

```
void unplug_force (session ref session_ref, VBD ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD ref`	`self`	The VBD to forcibly unplug

Minimum Role: vm-admin

Return Type: `void`

Class: VBDmetrics

This class is removed.

The metrics associated with a virtual block device

Fields for class: VBDmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`io_read_kbs`	`float`	*RO/runtime*	**Removed.** Read bandwidth (KiB/s)
`io_write_kbs`	`float`	*RO/runtime*	**Removed.** Write bandwidth (KiB/s)
`last_updated`	`datetime`	*RO/runtime*	**Removed.** Time at which this information was last updated

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	**Removed.** additional configuration
`uuid`	`string`	*RO/runtime*	**Removed.** Unique identifier/object reference

RPCs associated with class: VBDmetrics

RPC name: addtootherconfig

This message is removed.

Overview:

Add the given keyvalue pair to the otherconfig field of the given VBDmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VBD_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Return a list of all the VBDmetrics instances known to the system.

Signature:

```
VBD_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VBD_metrics ref set`

references to all objects

RPC name: getallrecords

This message is removed.

Overview:

Return a map of VBDmetrics references to VBDmetrics records for all VBDmetrics instances known to the system.

Signature:

```
(VBD_metrics ref -> VBD_metrics record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VBD_metrics ref -> VBD_metrics record) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

records of all objects

RPC name: getbyuuid

This message is removed.

Overview:

Get a reference to the VBDmetrics instance with the specified UUID.

Signature:

```
VBD_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VBD_metrics ref`

reference to the object

RPC name: getioreadkbs

This message is removed.

Overview:

Get the io/readkbs field of the given VBDmetrics.

Signature:

```
float get_io_read_kbs (session ref session_ref, VBD_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getiowritekbs

This message is removed.

Overview:

Get the io/writekbs field of the given VBDmetrics.

Signature:

```
float get_io_write_kbs (session ref session_ref, VBD_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getlastupdated

This message is removed.

Overview:

Get the lastupdated field of the given VBDmetrics.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
datetime get_last_updated (session ref session_ref, VBD_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getotherconfig

This message is removed.

Overview:

Get the otherconfig field of the given VBDmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VBD_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getrecord

This message is removed.

Overview:

Get a record containing the current state of the given VBDmetrics.

Signature:

```
VBD_metrics record get_record (session ref session_ref, VBD_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VBD_metrics record`

all fields from the object

RPC name: getuuid

This message is removed.

Overview:

Get the uuid field of the given VBDmetrics.

Signature:

```
string get_uuid (session ref session_ref, VBD_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is removed.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VBDmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VBD_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the otherconfig field of the given VBDmetrics.

Signature:

```
void set_other_config (session ref session_ref, VBD_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VBD_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VDI

A virtual disk image

Fields for class: VDI

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allow_caching`	`bool`	*RO/runtime*	true if this VDI is to be cached in the local cache SR
`allowed_operations`	`vdi_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`cbt_enabled`	`bool`	*RO/runtime*	True if changed blocks are tracked for this VDI
`crash_dumps`	`crashdump ref set`	*RO/runtime*	list of crash dumps that refer to this disk
`current_operations`	`(string -> vdi_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`is_a_snapshot`	`bool`	*RO/runtime*	true if this is a snapshot.
`is_tools_iso`	`bool`	*RO/runtime*	Whether this VDI is a Tools ISO
`location`	`string`	*RO/runtime*	location information
`managed`	`bool`	*RO/runtime*	
`metadata_latest`	`bool`	*RO/runtime*	Whether this VDI contains the latest known accessible metadata for the pool
`metadata_of_pool`	`pool ref`	*RO/runtime*	The pool whose metadata is contained in this VDI
`missing`	`bool`	*RO/runtime*	true if SR scan operation reported this VDI as not present on disk

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RO/constructor*	a notes field containing human-readable description
`name_label`	`string`	*RO/constructor*	a human-readable name
`on_boot`	`on_boot`	*RO/runtime*	The behaviour of this VDI on a VM boot
`other_config`	`(string -> string) map`	*RW*	additional configuration
`parent`	`VDI ref`	*RO/runtime*	**Deprecated.** This field is always null. Deprecated
`physical_utilisation`	`int`	*RO/runtime*	amount of physical space that the disk image is currently taking up on the storage repository (in bytes)
`read_only`	`bool`	*RO/constructor*	true if this disk may ONLY be mounted read-only
`sharable`	`bool`	*RO/constructor*	true if this disk may be shared
`sm_config`	`(string -> string) map`	*RW*	SM dependent data
`snapshot_of`	`VDI ref`	*RO/runtime*	Ref pointing to the VDI this snapshot is of.
`snapshot_time`	`datetime`	*RO/runtime*	Date/time when this snapshot was created.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`snapshots`	`VDI ref set`	*RO/runtime*	List pointing to all the VDIs snapshots.
`SR`	`SR ref`	*RO/constructor*	storage repository in which the VDI resides
`storage_lock`	`bool`	*RO/runtime*	true if this disk is locked at the storage level
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`type`	`vdi_type`	*RO/constructor*	type of the VDI
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VBDs`	`VBD ref set`	*RO/runtime*	list of vbds that refer to this disk
`virtual_size`	`int`	*RO/constructor*	size of disk as presented to the guest (in bytes). Note that, depending on storage backend type, requested size may not be respected exactly
`xenstore_data`	`(string -> string) map`	*RW*	data to be inserted into the xenstore tree (/local/domain/0/ backend/vbd/ <domid>/<device-id>/sm-data) after the VDI is attached. This is generally set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			by the SM backends on vdi_attach.

RPCs associated with class: VDI

RPC name: addtags

Overview:

Add the given value to the tags field of the given VDI. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, VDI ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VDI.

Signature:

```
void add_to_other_config (session ref session_ref, VDI ref self, string
key, string value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtosmconfig

Overview:

Add the given keyvalue pair to the smconfig field of the given VDI.

Signature:

```
void add_to_sm_config (session ref session_ref, VDI ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: addtoxenstoredata

Overview:

Add the given keyvalue pair to the xenstoredata field of the given VDI.

Signature:

```
void add_to_xenstore_data (session ref session_ref, VDI ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: clone

Overview:

Take an exact copy of the VDI and return a reference to the new disk. If any driverparams are specified then these are passed through to the storagespecific substrate driver that implements the clone operation. NB the clone lives in the same Storage Repository as its parent.

Signature:

```
VDI ref clone (session ref session_ref, VDI ref vdi, (string -> string)
map driver_params)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to clone
`(string -> string) map`	`driver_params`	Optional parameters that are passed through to the backend driver in order to specify storage-type-specific clone options

Minimum Role: vm-admin

Return Type: `VDI ref`

The ID of the newly created VDI.

RPC name: copy

Overview:

Copy either a full VDI or the block differences between two VDIs into either a fresh VDI or an existing VDI.

Signature:

```
VDI ref copy (session ref session_ref, VDI ref vdi, SR ref sr, VDI ref
base_vdi, VDI ref into_vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to copy
`SR ref`	`sr`	The destination SR (only required if the destination VDI is not specified

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`base_vdi`	The base VDI (only required if copying only changed blocks, by default all blocks will be copied)
`VDI ref`	`into_vdi`	The destination VDI to copy blocks into (if omitted then a destination SR must be provided and a fresh VDI will be created)

Minimum Role: vm-admin

Return Type: `VDI ref`

The reference of the VDI where the blocks were written.

Possible Error Codes: `VDI_READONLY`, `VDI_TOO_SMALL`, `VDI_NOT_SPARSE`

RPC name: create

Overview:

Create a new VDI instance, and return its handle.

Signature:

```
VDI ref create (session ref session_ref, VDI record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VDI ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the newly created object

RPC name: datadestroy

Overview:

Delete the data of the snapshot VDI, but keep its changed block tracking metadata. When successful, this call changes the type of the VDI to cbtmetadata. This operation is idempotent: calling it on a VDI of type cbtmetadata results in a noop, and no error will be thrown.

Signature:

```
void data_destroy (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI whose data should be deleted.

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `OPERATION_NOT_ALLOWED`, `VDI_INCOMPATIBLE_TYPE`, `VDI_NO_CBT_METADATA`, `VDI_IN_USE`, `VDI_IS_A_PHYSICAL_DEVICE`

RPC name: destroy

Overview:

Destroy the specified VDI instance.

Signature:

```
void destroy (session ref session_ref, VDI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: disablecbt

Overview:

Disable changed block tracking for the VDI. This call is only allowed on VDIs that support enabling CBT. It is an idempotent operation disabling CBT for a VDI for which CBT is not enabled results in a noop, and no error will be thrown.

Signature:

```
void disable_cbt (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI for which CBT should be disabled

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `OPERATION_NOT_ALLOWED`, `VDI_INCOMPATIBLE_TYPE`, `VDI_ON_BOOT_MODE_INCOMPATIBLE_WITH_OPERATION`

RPC name: enablecbt

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Enable changed block tracking for the VDI. This call is idempotent enabling CBT for a VDI for which CBT is already enabled results in a noop, and no error will be thrown.

Signature:

```
void enable_cbt (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI for which CBT should be enabled

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `OPERATION_NOT_ALLOWED`, `VDI_INCOMPATIBLE_TYPE`, `VDI_ON_BOOT_MODE_INCOMPATIBLE_WITH_OPERATION`

RPC name: forget

Overview:

Removes a VDI record from the database

Signature:

```
void forget (session ref session_ref, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to forget about

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VDIs known to the system.

Signature:

```
VDI ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VDI ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VDI references to VDI records for all VDIs known to the system.

Signature:

```
(VDI ref -> VDI record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VDI ref -> VDI record) map`

records of all objects

RPC name: getallowcaching

Overview:

Get the allowcaching field of the given VDI.

Signature:

```
bool get_allow_caching (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VDI.

Signature:

```
vdi_operations set get_allowed_operations (session ref session_ref, VDI
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vdi_operations set`

value of the field

RPC name: getbynamelabel

Overview:

Get all the VDI instances with the given label.

Signature:

```
VDI ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VDI ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VDI instance with the specified UUID.

Signature:

```
VDI ref get_by_uuid (session ref session_ref, string uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VDI ref`

reference to the object

RPC name: getcbtenabled

Overview:

Get the cbtenabled field of the given VDI.

Signature:

```
bool get_cbt_enabled (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getcrashdumps

Overview:

Get the crashdumps field of the given VDI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
crashdump ref set get_crash_dumps (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `crashdump ref set`

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VDI.

Signature:

```
(string -> vdi_operations) map get_current_operations (session ref
session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vdi_operations) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getisasnapshot

Overview:

Get the isasnapshot field of the given VDI.

Signature:

```
bool get_is_a_snapshot (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getistoolsiso

Overview:

Get the istoolsiso field of the given VDI.

Signature:

```
bool get_is_tools_iso (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getlocation

Overview:

Get the location field of the given VDI.

Signature:

```
string get_location (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getmanaged

Overview:

Get the managed field of the given VDI.

Signature:

```
bool get_managed (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetadatalatest

Overview:

Get the metadatalatest field of the given VDI.

Signature:

```
bool get_metadata_latest (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetadataofpool

Overview:

Get the metadataofpool field of the given VDI.

Signature:

```
pool ref get_metadata_of_pool (session ref session_ref, VDI ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `pool ref`

value of the field

RPC name: getmissing

Overview:

Get the missing field of the given VDI.

Signature:

```
bool get_missing (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given VDI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_name_description (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VDI.

Signature:

```
string get_name_label (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnbdinfo

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get details specifying how to access this VDI via a Network Block Device server. For each of a set of NBD server addresses on which the VDI is available, the return value set contains a vdinbdserverinfo object that contains an exportname to request once the NBD connection is established, and connection details for the address. An empty list is returned if there is no network that has a PIF on a host with access to the relevant SR, or if no such network has been assigned an NBDrelated purpose in its purpose field. To access the given VDI, any of the vdinbdserverinfo objects can be used to make a connection to a server, and then the VDI will be available by requesting the exportname.

Signature:

```
vdi_nbd_server_info record set get_nbd_info (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to access via Network Block Device protocol

Minimum Role: vm-admin

Return Type: `vdi_nbd_server_info record set`

The details necessary for connecting to the VDI over NBD. This includes an authentication token, so must be treated as sensitive material and must not be sent over insecure networks.

Possible Error Codes: `VDI_INCOMPATIBLE_TYPE`

RPC name: getonboot

Overview:

Get the onboot field of the given VDI.

Signature:

```
on_boot get_on_boot (session ref session_ref, VDI ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_boot`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VDI.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getparent

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the parent field of the given VDI.

Signature:

```
VDI ref get_parent (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getphysicalutilisation

Overview:

Get the physicalutilisation field of the given VDI.

Signature:

```
int get_physical_utilisation (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getreadonly

Overview:

Get the readonly field of the given VDI.

Signature:

```
bool get_read_only (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VDI.

Signature:

```
VDI record get_record (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VDI record`

all fields from the object

RPC name: getsharable

Overview:

Get the sharable field of the given VDI.

Signature:

```
bool get_sharable (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getsmconfig

Overview:

Get the smconfig field of the given VDI.

Signature:

```
(string -> string) map get_sm_config (session ref session_ref, VDI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsnapshotof

Overview:

Get the snapshotof field of the given VDI.

Signature:

```
VDI ref get_snapshot_of (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: getsnapshottime

Overview:

Get the snapshottime field of the given VDI.

Signature:

```
datetime get_snapshot_time (session ref session_ref, VDI ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getsnapshots

Overview:

Get the snapshots field of the given VDI.

Signature:

```
VDI ref set get_snapshots (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref set`

value of the field

RPC name: getSR

Overview:

Get the SR field of the given VDI.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR ref get_SR (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getstoragelock

Overview:

Get the storagelock field of the given VDI.

Signature:

```
bool get_storage_lock (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: gettags

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the tags field of the given VDI.

Signature:

```
string set get_tags (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VDI.

Signature:

```
vdi_type get_type (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vdi_type`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuuid

Overview:

Get the uuid field of the given VDI.

Signature:

```
string get_uuid (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVBDs

Overview:

Get the VBDs field of the given VDI.

Signature:

```
VBD ref set get_VBDs (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VBD ref set`

value of the field

RPC name: getvirtualsize

Overview:

Get the virtualsize field of the given VDI.

Signature:

```
int get_virtual_size (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getxenstoredata

Overview:

Get the xenstoredata field of the given VDI.

Signature:

```
(string -> string) map get_xenstore_data (session ref session_ref, VDI ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: introduce

Overview:

Create a new VDI record in the database only

Signature:

```
VDI ref introduce (session ref session_ref, string uuid, string
name_label, string name_description, SR ref SR, vdi_type type, bool
sharable, bool read_only, (string -> string) map other_config, string
location, (string -> string) map xenstore_data, (string -> string) map
sm_config, bool managed, int virtual_size, int physical_utilisation, pool
ref metadata_of_pool, bool is_a_snapshot, datetime snapshot_time, VDI ref
snapshot_of)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	The uuid of the disk to introduce
`string`	`name_label`	The name of the disk record
`string`	`name_description`	The description of the disk record
`SR ref`	`SR`	The SR that the VDI is in
`vdi_type`	`type`	The type of the VDI

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`sharable`	true if this disk may be shared
`bool`	`read_only`	true if this disk may ONLY be mounted read-only
`(string -> string) map`	`other_config`	additional configuration
`string`	`location`	location information
`(string -> string) map`	`xenstore_data`	Data to insert into xenstore
`(string -> string) map`	`sm_config`	Storage-specific config
`bool`	`managed`	Storage-specific config
`int`	`virtual_size`	Storage-specific config
`int`	`physical_utilisation`	Storage-specific config
`pool ref`	`metadata_of_pool`	Storage-specific config
`bool`	`is_a_snapshot`	Storage-specific config
`datetime`	`snapshot_time`	Storage-specific config. When the timezone is missing, UTC is assumed
`VDI ref`	`snapshot_of`	Storage-specific config

Minimum Role: vm-admin

Return Type: `VDI ref`

The ref of the newly created VDI record.

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`

RPC name: listchangedblocks

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Compare two VDIs in 64k block increments and report which blocks differ. This operation is not allowed when vdito is attached to a VM.

Signature:

```
string list_changed_blocks (session ref session_ref, VDI ref vdi_from, VDI ref vdi_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi_from`	The first VDI.
`VDI ref`	`vdi_to`	The second VDI.

Minimum Role: vm-operator

Return Type: `string`

A base64 stringencoding of the bitmap showing which blocks differ in the two VDIs.

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`, `VDI_MISSING`, `SR_NOT_ATTACHED`, `SR_HAS_NO_PBDS`, `VDI_IN_USE`

RPC name: opendatabase

Overview:

Load the metadata found on the supplied VDI and return a session reference which can be used in API calls to query its contents.

Signature:

```
session ref open_database (session ref session_ref, VDI ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI which contains the database to open

Minimum Role: pool-operator

Return Type: `session ref`

A session which can be used to query the database

RPC name: poolmigrate

Overview:

Migrate a VDI, which may be attached to a running guest, to a different SR. The destination SR must be visible to the guest.

Signature:

```
VDI ref pool_migrate (session ref session_ref, VDI ref vdi, SR ref sr,
(string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION		
`session ref`	`session_ref`	Reference to a valid session		
`VDI ref`	`vdi`	The VDI to migrate		
`SR ref`	`sr`	The destination SR		
`(string -> string) map`	`options`	Extra parameters. Supports: "dest-img-format" (raw	vhd	qcow2) to specify the image format to use on the destination SR.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-power-admin

Return Type: `VDI ref`

The new reference of the migrated VDI.

RPC name: readdatabasepooluuid

Overview:

Check the VDI cache for the pool UUID of the database on this VDI.

Signature:

```
string read_database_pool_uuid (session ref session_ref, VDI ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The metadata VDI to look up in the cache.

Minimum Role: read-only

Return Type: `string`

The cached pool UUID of the database on the VDI.

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VDI. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VDI ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromsmconfig

Overview:

Remove the given key and its corresponding value from the smconfig field of the given VDI. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_sm_config (session ref session_ref, VDI ref self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromxenstoredata

Overview:

Remove the given key and its corresponding value from the xenstoredata field of the given VDI. If the key is not in that Map, then do nothing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void remove_from_xenstore_data (session ref session_ref, VDI ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given VDI. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, VDI ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: resize

Overview:

Resize the VDI.

Signature:

```
void resize (session ref session_ref, VDI ref vdi, int size)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to resize
`int`	`size`	The new size of the VDI

Minimum Role: vm-admin

Return Type: `void`

RPC name: resizeonline

This message is removed.

Overview:

Resize the VDI which may or may not be attached to running guests.

Signature:

```
void resize_online (session ref session_ref, VDI ref vdi, int size)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`vdi`	The VDI to resize
`int`	`size`	The new size of the VDI

Minimum Role: vm-admin

Return Type: `void`

RPC name: revert

Overview:

Copy the contents of a snapshot to the VDI it's related to. The original contents of the VDI are lost.

Signature:

```
void revert (session ref session_ref, VDI ref snapshot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`snapshot`	The snapshot to which we want to revert

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `UNIMPLEMENTED_IN_SM_BACKEND`

RPC name: setallowcaching

Overview:

Set the value of the allowcaching parameter. This value can only be changed when the VDI is not attached to a running VM. The caching behaviour is only affected by this flag for VHDbased VDIs that have one parent and no child VHDs. Moreover, caching only takes place when the host running the VM containing this VDI has a nominated SR for local caching.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_allow_caching (session ref session_ref, VDI ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`bool`	`value`	The value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name description of the VDI. This can only happen when its SR is currently attached.

Signature:

```
void set_name_description (session ref session_ref, VDI ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`string`	`value`	The name description for the VDI

Minimum Role: vm-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name label of the VDI. This can only happen when then its SR is currently attached.

Signature:

```
void set_name_label (session ref session_ref, VDI ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`string`	`value`	The name lable for the VDI

Minimum Role: vm-admin

Return Type: `void`

RPC name: setonboot

Overview:

Set the value of the onboot parameter. This value can only be changed when the VDI is not attached to a running VM.

Signature:

```
void set_on_boot (session ref session_ref, VDI ref self, on_boot value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VDI ref`	`self`	The VDI to modify
`on_boot`	`value`	The value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VDI.

Signature:

```
void set_other_config (session ref session_ref, VDI ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setreadonly

Overview:

Sets the VDI's readonly field

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_read_only (session ref session_ref, VDI ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`bool`	`value`	The new value of the VDI's read_only field

Minimum Role: vm-admin

Return Type: `void`

RPC name: setsharable

Overview:

Sets the VDI's sharable field

Signature:

```
void set_sharable (session ref session_ref, VDI ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	The VDI to modify
`bool`	`value`	The new value of the VDI's sharable field

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setsmconfig

Overview:

Set the smconfig field of the given VDI.

Signature:

```
void set_sm_config (session ref session_ref, VDI ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: settags

Overview:

Set the tags field of the given VDI.

Signature:

```
void set_tags (session ref session_ref, VDI ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: setxenstoredata

Overview:

Set the xenstoredata field of the given VDI.

Signature:

```
void set_xenstore_data (session ref session_ref, VDI ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: snapshot

Overview:

Take a readonly snapshot of the VDI, returning a reference to the snapshot. If any driverparams are specified then these are passed through to the storagespecific substrate driver that takes the snapshot. NB the snapshot lives in the same Storage Repository as its parent.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VDI ref snapshot (session ref session_ref, VDI ref vdi, (string -> string)
map driver_params)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI to snapshot
`(string -> string) map`	`driver_params`	Optional parameters that can be passed through to backend driver in order to specify storage-type-specific snapshot options

Minimum Role: vm-admin

Return Type: `VDI ref`

The ID of the newly created VDI.

RPC name: update

Overview:

Ask the storage backend to refresh the fields in the VDI object

Signature:

```
void update (session ref session_ref, VDI ref vdi)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VDI ref`	`vdi`	The VDI whose stats (eg size) should be updated

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`

Class: vdinbdserverinfo

Details for connecting to a VDI using the Network Block Device protocol

Fields for class: vdinbdserverinfo

FIELD	TYPE	QUALIFIER	DESCRIPTION
`address`	`string`	*RO/runtime*	An address on which the server can be reached; this can be IPv4, IPv6, or a DNS name.
`cert`	`string`	*RO/runtime*	The TLS certificate of the server
`exportname`	`string`	*RO/runtime*	The exportname to request over NBD. This holds details including an authentication token, so it must be protected appropriately. Clients should regard the exportname as an opaque string or token.
`port`	`int`	*RO/runtime*	The TCP port
`subject`	`string`	*RO/runtime*	For convenience, this redundant field holds a DNS (hostname) subject of the certificate. This can be a

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			wildcard, but only for a certificate that has a wildcard subject and no concrete hostname subjects.

RPCs associated with class: vdinbdserverinfo

Class vdinbdserverinfo has no RPCs associated with it.

Class: VGPU

A virtual GPU vGPU

Fields for class: VGPU

FIELD	TYPE	QUALIFIER	DESCRIPTION
`compatibility_metadata`	`(string -> string) map`	*RO/runtime*	VGPU metadata to determine whether a VGPU can migrate between two PGPUs
`currently_attached`	`bool`	*RO/runtime*	Reflects whether the virtual device is currently connected to a physical device
`device`	`string`	*RO/runtime*	Order in which the devices are plugged into the VM
`extra_args`	`string`	*RW*	Extra arguments for vGPU and passed to demu
`GPU_group`	`GPU_group ref`	*RO/runtime*	GPU group used by the vGPU
`other_config`	`(string -> string) map`	*RW*	Additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`PCI`	`PCI ref`	*RO/runtime*	Device passed trough to VM, either as full device or SR-IOV virtual function
`resident_on`	`PGPU ref`	*RO/runtime*	The PGPU on which this VGPU is running
`scheduled_to_be_resident_on`	`PGPU ref`	*RO/runtime*	The PGPU on which this VGPU is scheduled to run
`type`	`VGPU_type ref`	*RO/runtime*	Preset type for this VGPU
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VM`	`VM ref`	*RO/runtime*	VM that owns the vGPU

RPCs associated with class: VGPU

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VGPU.

Signature:

```
void add_to_other_config (session ref session_ref, VGPU ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Signature:

```
VGPU ref create (session ref session_ref, VM ref VM, GPU_group ref
GPU_group, string device, (string -> string) map other_config, VGPU_type
ref type)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`VM`	
`GPU_group ref`	`GPU_group`	
`string`	`device`	
`(string -> string) map`	`other_config`	
`VGPU_type ref`	`type`	

Minimum Role: pool-operator

Return Type: `VGPU ref`

The reference of the created VGPU object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Signature:

```
void destroy (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	The vGPU to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VGPUs known to the system.

Signature:

```
VGPU ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VGPU ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of VGPU references to VGPU records for all VGPUs known to the system.

Signature:

```
(VGPU ref -> VGPU record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VGPU ref -> VGPU record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VGPU instance with the specified UUID.

Signature:

```
VGPU ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VGPU ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

reference to the object

RPC name: getcompatibilitymetadata

Overview:

Get the compatibilitymetadata field of the given VGPU.

Signature:

```
(string -> string) map get_compatibility_metadata (session ref
session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VGPU.

Signature:

```
bool get_currently_attached (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdevice

Overview:

Get the device field of the given VGPU.

Signature:

```
string get_device (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getextraargs

Overview:

Get the extraargs field of the given VGPU.

Signature:

```
string get_extra_args (session ref session_ref, VGPU ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getGPUgroup

Overview:

Get the GPUgroup field of the given VGPU.

Signature:

```
GPU_group ref get_GPU_group (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VGPU.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_other_config (session ref session_ref, VGPU ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPCI

Overview:

Get the PCI field of the given VGPU.

Signature:

```
PCI ref get_PCI (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

Overview:

Get a record containing the current state of the given VGPU.

Signature:

```
VGPU record get_record (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU record`

all fields from the object

RPC name: getresidenton

Overview:

Get the residenton field of the given VGPU.

Signature:

```
PGPU ref get_resident_on (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `PGPU ref`

value of the field

RPC name: getscheduledtoberesidenton

Overview:

Get the scheduledtoberesidenton field of the given VGPU.

Signature:

```
PGPU ref get_scheduled_to_be_resident_on (session ref session_ref, VGPU
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VGPU.

Signature:

```
VGPU_type ref get_type (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VGPU.

Signature:

```
string get_uuid (session ref session_ref, VGPU ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VGPU.

Signature:

```
VM ref get_VM (session ref session_ref, VGPU ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VGPU. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VGPU ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setextraargs

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the extraargs field of the given VGPU.

Signature:

```
void set_extra_args (session ref session_ref, VGPU ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VGPU.

Signature:

```
void set_other_config (session ref session_ref, VGPU ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Class: VGPUtype

A type of virtual GPU

Fields for class: VGPUtype

FIELD	TYPE	QUALIFIER	DESCRIPTION
`compatible_types_in_vm`	`VGPU_type ref set`	*RO/runtime*	List of VGPU types which are compatible in one VM
`enabled_on_GPU_groups`	`GPU_group ref set`	*RO/runtime*	List of GPU groups in which at least one have this VGPU type enabled
`enabled_on_PGPUs`	`PGPU ref set`	*RO/runtime*	List of PGPUs that have this VGPU type enabled
`experimental`	`bool`	*RO/constructor*	Indicates whether VGPUs of this type should be considered experimental
`framebuffer_size`	`int`	*RO/constructor*	Framebuffer size of the VGPU type, in bytes
`identifier`	`string`	*RO/constructor*	Key used to identify VGPU types and avoid creating duplicates - this field is used internally and not intended for interpretation by API clients

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`implementation`	`vgpu_type_implementation`	*RO/constructor*	The internal implementation of this VGPU type
`max_heads`	`int`	*RO/constructor*	Maximum number of displays supported by the VGPU type
`max_resolution_x`	`int`	*RO/constructor*	Maximum resolution (width) supported by the VGPU type
`max_resolution_y`	`int`	*RO/constructor*	Maximum resolution (height) supported by the VGPU type
`model_name`	`string`	*RO/constructor*	Model name associated with the VGPU type
`supported_on_GPU_groups`	`GPU_group ref set`	*RO/runtime*	List of GPU groups in which at least one PGPU supports this VGPU type
`supported_on_PGPUs`	`PGPU ref set`	*RO/runtime*	List of PGPUs that support this VGPU type
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`vendor_name`	`string`	*RO/constructor*	Name of VGPU vendor
`VGPUs`	`VGPU ref set`	*RO/runtime*	List of VGPUs of this type

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VGPUtype

RPC name: getall

Overview:

Return a list of all the VGPUtypes known to the system.

Signature:

```
VGPU_type ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VGPU_type ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VGPUtype references to VGPUtype records for all VGPUtypes known to the system.

Signature:

```
(VGPU_type ref -> VGPU_type record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VGPU_type ref -> VGPU_type record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VGPUtype instance with the specified UUID.

Signature:

```
VGPU_type ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VGPU_type ref`

reference to the object

RPC name: getcompatibletypesinvm

Overview:

Get the compatibletypesinvm field of the given VGPUtype.

Signature:

```
VGPU_type ref set get_compatible_types_in_vm (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type ref set`

value of the field

RPC name: getenabledonGPUgroups

Overview:

Get the enabledonGPUgroups field of the given VGPUtype.

Signature:

```
GPU_group ref set get_enabled_on_GPU_groups (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref set`

value of the field

RPC name: getenabledonPGPUs

Overview:

Get the enabledonPGPUs field of the given VGPUtype.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PGPU ref set get_enabled_on_PGPUs (session ref session_ref, VGPU_type ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getexperimental

Overview:

Get the experimental field of the given VGPUtype.

Signature:

```
bool get_experimental (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getframebuffersize

Overview:

Get the framebuffersize field of the given VGPUtype.

Signature:

```
int get_framebuffer_size (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getidentifier

Overview:

Get the identifier field of the given VGPUtype.

Signature:

```
string get_identifier (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getimplementation

Overview:

Get the implementation field of the given VGPUtype.

Signature:

```
vgpu_type_implementation get_implementation (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vgpu_type_implementation`

value of the field

RPC name: getmaxheads

Overview:

Get the maxheads field of the given VGPUtype.

Signature:

```
int get_max_heads (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmaxresolutionx

Overview:

Get the maxresolutionx field of the given VGPUtype.

Signature:

```
int get_max_resolution_x (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmaxresolutiony

Overview:

Get the maxresolutiony field of the given VGPUtype.

Signature:

```
int get_max_resolution_y (session ref session_ref, VGPU_type ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmodelname

Overview:

Get the modelname field of the given VGPUtype.

Signature:

```
string get_model_name (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VGPUtype.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VGPU_type record get_record (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU_type record`

all fields from the object

RPC name: getsupportedonGPUgroups

Overview:

Get the supportedonGPUgroups field of the given VGPUtype.

Signature:

```
GPU_group ref set get_supported_on_GPU_groups (session ref session_ref,
VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `GPU_group ref set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsupportedonPGPUs

Overview:

Get the supportedonPGPUs field of the given VGPUtype.

Signature:

```
PGPU ref set get_supported_on_PGPUs (session ref session_ref, VGPU_type
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PGPU ref set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VGPUtype.

Signature:

```
string get_uuid (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getvendorname

Overview:

Get the vendorname field of the given VGPUtype.

Signature:

```
string get_vendor_name (session ref session_ref, VGPU_type ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVGPUs

Overview:

Get the VGPUs field of the given VGPUtype.

Signature:

```
VGPU ref set get_VGPUs (session ref session_ref, VGPU_type ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VGPU_type ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

Class: VIF

A virtual network interface

Fields for class: VIF

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vif_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`current_operations`	`(string -> vif_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`currently_attached`	`bool`	*RO/constructor*	is the device currently attached (erased on reboot)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`device`	`string`	*RO/constructor*	order in which VIF backends are created by xapi. Guaranteed to be an unsigned decimal integer.
`ipv4_addresses`	`string set`	*RO/runtime*	IPv4 addresses in CIDR format
`ipv4_allowed`	`string set`	*RO/constructor*	A list of IPv4 addresses which can be used to filter traffic passing through this VIF
`ipv4_configuration_mode`	`vif_ipv4_configuration_mode`	*RO/runtime*	Determines whether IPv4 addresses are configured on the VIF
`ipv4_gateway`	`string`	*RO/runtime*	IPv4 gateway (the empty string means that no gateway is set)
`ipv6_addresses`	`string set`	*RO/runtime*	IPv6 addresses in CIDR format
`ipv6_allowed`	`string set`	*RO/constructor*	A list of IPv6 addresses which can be used to filter traffic passing through this VIF
`ipv6_configuration_mode`	`vif_ipv6_configuration_mode`	*RO/runtime*	Determines whether IPv6 addresses are configured on the VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`ipv6_gateway`	`string`	*RO/runtime*	IPv6 gateway (the empty string means that no gateway is set)
`locking_mode`	`vif_locking_mode`	*RO/constructor*	current locking mode of the VIF
`MAC`	`string`	*RO/constructor*	ethernet MAC address of virtual interface, as exposed to guest
`MAC_autogenerated`	`bool`	*RO/runtime*	true if the MAC was autogenerated; false indicates it was set manually
`metrics`	`VIF_metrics ref`	*RO/runtime*	**Removed.** metrics associated with this VIF
`MTU`	`int`	*RO/constructor*	MTU in octets
`network`	`network ref`	*RO/constructor*	virtual network to which this vif is connected
`other_config`	`(string -> string) map`	*RW*	additional configuration
`qos_algorithm_params`	`(string -> string) map`	*RW*	parameters for chosen QoS algorithm
`qos_algorithm_type`	`string`	*RW*	QoS algorithm to use
`qos_supported_algorithms`	`string set`	*RO/runtime*	supported QoS algorithms for this VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`runtime_properties`	`(string -> string) map`	*RO/runtime*	Device runtime properties
`status_code`	`int`	*RO/runtime*	error/success code associated with last attach-operation (erased on reboot)
`status_detail`	`string`	*RO/runtime*	error/success information associated with last attach-operation status (erased on reboot)
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VM`	`VM ref`	*RO/constructor*	virtual machine to which this vif is connected

RPCs associated with class: VIF

RPC name: addipv4allowed

Overview:

Associates an IPv4 address with this VIF

Signature:

```
void add_ipv4_allowed (session ref session_ref, VIF ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	The VIF which the IP address will be associated with
`string`	`value`	The IP address which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: addipv6allowed

Overview:

Associates an IPv6 address with this VIF

Signature:

```
void add_ipv6_allowed (session ref session_ref, VIF ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF which the IP address will be associated with
`string`	`value`	The IP address which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VIF.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void add_to_other_config (session ref session_ref, VIF ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoqosalgorithmparams

Overview:

Add the given keyvalue pair to the qos/algorithmparams field of the given VIF.

Signature:

```
void add_to_qos_algorithm_params (session ref session_ref, VIF ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: configureipv4

Overview:

Configure IPv4 settings for this virtual interface

Signature:

```
void configure_ipv4 (session ref session_ref, VIF ref self,
vif_ipv4_configuration_mode mode, string address, string gateway)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to configure
`vif_ipv4_configuration_mode`	`mode`	Whether to use DHCP, static or no IPv4 assignment
`string`	`address`	The IPv4 address in <addr>/<prefix length> format (for static mode only)
`string`	`gateway`	The IPv4 gateway (for static mode only; leave empty to not set a gateway)

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: configureipv6

Overview:

Configure IPv6 settings for this virtual interface

Signature:

```
void configure_ipv6 (session ref session_ref, VIF ref self,
vif_ipv6_configuration_mode mode, string address, string gateway)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to configure
`vif_ipv6_configuration_mode`	`mode`	Whether to use autoconf, static or no IPv6 assignment
`string`	`address`	The IPv6 address in <addr>/<prefix length> format (for static mode only)
`string`	`gateway`	The IPv6 gateway (for static mode only; leave empty to not set a gateway)

Minimum Role: vm-operator

Return Type: `void`

RPC name: create

Overview:

Create a new VIF instance, and return its handle.

Signature:

```
VIF ref create (session ref session_ref, VIF record args)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VIF ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VIF instance.

Signature:

```
void destroy (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VIFs known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VIF ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VIF ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VIF references to VIF records for all VIFs known to the system.

Signature:

```
(VIF ref -> VIF record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VIF ref -> VIF record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
vif_operations set get_allowed_operations (session ref session_ref, VIF
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_operations set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VIF instance with the specified UUID.

Signature:

```
VIF ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VIF ref`

reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VIF.

Signature:

```
(string -> vif_operations) map get_current_operations (session ref
session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vif_operations) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VIF.

Signature:

```
bool get_currently_attached (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getdevice

Overview:

Get the device field of the given VIF.

Signature:

```
string get_device (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getipv4addresses

Overview:

Get the ipv4addresses field of the given VIF.

Signature:

```
string set get_ipv4_addresses (session ref session_ref, VIF ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv4allowed

Overview:

Get the ipv4allowed field of the given VIF.

Signature:

```
string set get_ipv4_allowed (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv4configurationmode

Overview:

Get the ipv4configurationmode field of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
vif_ipv4_configuration_mode get_ipv4_configuration_mode (session ref
session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_ipv4_configuration_mode`

value of the field

RPC name: getipv4gateway

Overview:

Get the ipv4gateway field of the given VIF.

Signature:

```
string get_ipv4_gateway (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getipv6addresses

Overview:

Get the ipv6addresses field of the given VIF.

Signature:

```
string set get_ipv6_addresses (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getipv6allowed

Overview:

Get the ipv6allowed field of the given VIF.

Signature:

```
string set get_ipv6_allowed (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string set`

value of the field

RPC name: getipv6configurationmode

Overview:

Get the ipv6configurationmode field of the given VIF.

Signature:

```
vif_ipv6_configuration_mode get_ipv6_configuration_mode (session ref
session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_ipv6_configuration_mode`

value of the field

RPC name: getipv6gateway

Overview:

Get the ipv6gateway field of the given VIF.

Signature:

```
string get_ipv6_gateway (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getlockingmode

Overview:

Get the lockingmode field of the given VIF.

Signature:

```
vif_locking_mode get_locking_mode (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vif_locking_mode`

value of the field

RPC name: getMAC

Overview:

Get the MAC field of the given VIF.

Signature:

```
string get_MAC (session ref session_ref, VIF ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getMACautogenerated

Overview:

Get the MACautogenerated field of the given VIF.

Signature:

```
bool get_MAC_autogenerated (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmetrics

This message is removed.

Overview:

Get the metrics field of the given VIF.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VIF_metrics ref get_metrics (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF_metrics ref`

value of the field

RPC name: getMTU

Overview:

Get the MTU field of the given VIF.

Signature:

```
int get_MTU (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnetwork

Overview:

Get the network field of the given VIF.

Signature:

```
network ref get_network (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `network ref`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VIF.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmparams

Overview:

Get the qos/algorithmparams field of the given VIF.

Signature:

```
(string -> string) map get_qos_algorithm_params (session ref session_ref,
VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getqosalgorithmtype

Overview:

Get the qos/algorithmtype field of the given VIF.

Signature:

```
string get_qos_algorithm_type (session ref session_ref, VIF ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getqossupportedalgorithms

Overview:

Get the qos/supportedalgorithms field of the given VIF.

Signature:

```
string set get_qos_supported_algorithms (session ref session_ref, VIF ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VIF record get_record (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF record`

all fields from the object

RPC name: getruntimeproperties

Overview:

Get the runtimeproperties field of the given VIF.

Signature:

```
(string -> string) map get_runtime_properties (session ref session_ref,
VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getstatuscode

Overview:

Get the statuscode field of the given VIF.

Signature:

```
int get_status_code (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getstatusdetail

Overview:

Get the statusdetail field of the given VIF.

Signature:

```
string get_status_detail (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VIF.

Signature:

```
string get_uuid (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VIF.

Signature:

```
VM ref get_VM (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: move

Overview:

Move the specified VIF to the specified network, even while the VM is running

Signature:

```
void move (session ref session_ref, VIF ref self, network ref network)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to move
`network ref`	`network`	The network to move it to

Minimum Role: vm-admin

Return Type: `void`

RPC name: plug

Overview:

Hotplug the specified VIF, dynamically attaching it to the running VM

Signature:

```
void plug (session ref session_ref, VIF ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to hotplug

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VIF. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VIF ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromqosalgorithmparams

Overview:

Remove the given key and its corresponding value from the qos/algorithmparams field of the given VIF. If the key is not in that Map, then do nothing.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void remove_from_qos_algorithm_params (session ref session_ref, VIF ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removeipv4allowed

Overview:

Removes an IPv4 address from this VIF

Signature:

```
void remove_ipv4_allowed (session ref session_ref, VIF ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF from which the IP address will be removed
`string`	`value`	The IP address which will be removed from the VIF

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: removeipv6allowed

Overview:

Removes an IPv6 address from this VIF

Signature:

```
void remove_ipv6_allowed (session ref session_ref, VIF ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF from which the IP address will be removed
`string`	`value`	The IP address which will be removed from the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: setipv4allowed

Overview:

Set the IPv4 addresses to which traffic on this VIF can be restricted

Signature:

```
void set_ipv4_allowed (session ref session_ref, VIF ref self, string set
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF which the IP addresses will be associated with
`string set`	`value`	The IP addresses which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: setipv6allowed

Overview:

Set the IPv6 addresses to which traffic on this VIF can be restricted

Signature:

```
void set_ipv6_allowed (session ref session_ref, VIF ref self, string set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF which the IP addresses will be associated with
`string set`	`value`	The IP addresses which will be associated with the VIF

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setlockingmode

Overview:

Set the locking mode for this VIF

Signature:

```
void set_locking_mode (session ref session_ref, VIF ref self,
vif_locking_mode value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF whose locking mode will be set
`vif_locking_mode`	`value`	The new locking mode for the VIF

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VIF.

Signature:

```
void set_other_config (session ref session_ref, VIF ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmparams

Overview:

Set the qos/algorithmparams field of the given VIF.

Signature:

```
void set_qos_algorithm_params (session ref session_ref, VIF ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setqosalgorithmtype

Overview:

Set the qos/algorithmtype field of the given VIF.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_qos_algorithm_type (session ref session_ref, VIF ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: unplug

Overview:

Hotunplug the specified VIF, dynamically unattaching it from the running VM

Signature:

```
void unplug (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to hot-unplug

Minimum Role: vm-admin

Return Type: `void`

RPC name: unplugforce

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Forcibly unplug the specified VIF

Signature:

```
void unplug_force (session ref session_ref, VIF ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF ref`	`self`	The VIF to forcibly unplug

Minimum Role: vm-admin

Return Type: `void`

Class: VIFmetrics

This class is removed.

The metrics associated with a virtual network device

Fields for class: VIFmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`io_read_kbs`	`float`	*RO/runtime*	**Removed.** Read bandwidth (KiB/s)
`io_write_kbs`	`float`	*RO/runtime*	**Removed.** Write bandwidth (KiB/s)
`last_updated`	`datetime`	*RO/runtime*	**Removed.** Time at which this information was last updated
`other_config`	`(string -> string) map`	*RW*	**Removed.** additional configuration

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`uuid`	`string`	*RO/runtime*	**Removed.** Unique identifier/object reference

RPCs associated with class: VIFmetrics

RPC name: addtootherconfig

This message is removed.

Overview:

Add the given keyvalue pair to the otherconfig field of the given VIFmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VIF_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

This message is removed.

Overview:

Return a list of all the VIFmetrics instances known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VIF_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VIF_metrics ref set`

references to all objects

RPC name: getallrecords

This message is removed.

Overview:

Return a map of VIFmetrics references to VIFmetrics records for all VIFmetrics instances known to the system.

Signature:

```
(VIF_metrics ref -> VIF_metrics record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VIF_metrics ref -> VIF_metrics record) map`

records of all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbyuuid

This message is removed.

Overview:

Get a reference to the VIFmetrics instance with the specified UUID.

Signature:

```
VIF_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VIF_metrics ref`

reference to the object

RPC name: getioreadkbs

This message is removed.

Overview:

Get the io/readkbs field of the given VIFmetrics.

Signature:

```
float get_io_read_kbs (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getiowritekbs

This message is removed.

Overview:

Get the io/writekbs field of the given VIFmetrics.

Signature:

```
float get_io_write_kbs (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getlastupdated

This message is removed.

Overview:

Get the lastupdated field of the given VIFmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
datetime get_last_updated (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getotherconfig

This message is removed.

Overview:

Get the otherconfig field of the given VIFmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

This message is removed.

Overview:

Get a record containing the current state of the given VIFmetrics.

Signature:

```
VIF_metrics record get_record (session ref session_ref, VIF_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VIF_metrics record`

all fields from the object

RPC name: getuuid

This message is removed.

Overview:

Get the uuid field of the given VIFmetrics.

Signature:

```
string get_uuid (session ref session_ref, VIF_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VIF_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

This message is removed.

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VIFmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VIF_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the otherconfig field of the given VIFmetrics.

Signature:

```
void set_other_config (session ref session_ref, VIF_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VIF_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VLAN

A VLAN mux/demux

Fields for class: VLAN

FIELD	TYPE	QUALIFIER	DESCRIPTION
`other_config`	`(string -> string) map`	*RW*	additional configuration
`tag`	`int`	*RO/constructor*	VLAN tag in use
`tagged_PIF`	`PIF ref`	*RO/constructor*	interface on which traffic is tagged
`untagged_PIF`	`PIF ref`	*RO/runtime*	interface on which traffic is untagged
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VLAN

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VLAN.

Signature:

```
void add_to_other_config (session ref session_ref, VLAN ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a VLAN mux/demuxer

Signature:

```
VLAN ref create (session ref session_ref, PIF ref tagged_PIF, int tag,
network ref network)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`PIF ref`	`tagged_PIF`	PIF which receives the tagged traffic
`int`	`tag`	VLAN tag to use
`network ref`	`network`	Network to receive the untagged traffic

Minimum Role: pool-operator

Return Type: `VLAN ref`

The reference of the created VLAN object

RPC name: destroy

Overview:

Destroy a VLAN mux/demuxer

Signature:

```
void destroy (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	VLAN mux/demuxer to destroy

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return a list of all the VLANs known to the system.

Signature:

```
VLAN ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VLAN ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VLAN references to VLAN records for all VLANs known to the system.

Signature:

```
(VLAN ref -> VLAN record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VLAN ref -> VLAN record) map`

records of all objects

RPC name: getbyuuid

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get a reference to the VLAN instance with the specified UUID.

Signature:

```
VLAN ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VLAN ref`

reference to the object

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VLAN.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VLAN ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VLAN.

Signature:

```
VLAN record get_record (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VLAN record`

all fields from the object

RPC name: gettag

Overview:

Get the tag field of the given VLAN.

Signature:

```
int get_tag (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gettaggedPIF

Overview:

Get the taggedPIF field of the given VLAN.

Signature:

```
PIF ref get_tagged_PIF (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getuntaggedPIF

Overview:

Get the untaggedPIF field of the given VLAN.

Signature:

```
PIF ref get_untagged_PIF (session ref session_ref, VLAN ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PIF ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VLAN.

Signature:

```
string get_uuid (session ref session_ref, VLAN ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VLAN. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_other_config (session ref session_ref, VLAN ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VLAN.

Signature:

```
void set_other_config (session ref session_ref, VLAN ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VLAN ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Class: VM

A virtual machine or 'guest'.

Fields for class: VM

FIELD	TYPE	QUALIFIER	DESCRIPTION
`actions_after_crash`	`on_crash_behaviour`	*RO/constructor*	action to take if the guest crashes
`actions_after_reboot`	`on_normal_exit`	*RW*	action to take after the guest has rebooted itself
`actions_after_shutdown`	`on_normal_exit`	*RW*	action to take after the guest has shutdown itself
`actions_after_softreboot`	`on_softreboot_behavior`	*RW*	action to take after soft reboot
`affinity`	`host ref`	*RW*	A host which the VM has some affinity for (or NULL). This is used as a hint to the start call when it decides where to run the VM. Resource constraints may cause the VM to be started elsewhere.
`allowed_operations`	`vm_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`appliance`	`VM_appliance ref`	*RO/constructor*	the appliance to which this VM belongs
`attached_PCIs`	`PCI ref set`	*RO/runtime*	Currently passed-through PCI devices
`bios_strings`	`(string -> string) map`	*RO/runtime*	BIOS strings
`blobs`	`(string -> blob ref) map`	*RO/runtime*	Binary blobs associated with this VM
`blocked_operations`	`(vm_operations -> string) map`	*RO/constructor*	List of operations which have been explicitly blocked and an error code
`children`	`VM ref set`	*RO/runtime*	List pointing to all the children of this VM
`consoles`	`console ref set`	*RO/runtime*	virtual console devices
`crash_dumps`	`crashdump ref set`	*RO/runtime*	crash dumps associated with this VM
`current_operations`	`(string -> vm_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`domain_type`	`domain_type`	*RO/constructor*	The type of domain that will be created

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			when the VM is started
`domarch`	`string`	*RO/runtime*	Domain architecture (if available, null string otherwise)
`domid`	`int`	*RO/runtime*	domain ID (if available, -1 otherwise)
`generation_id`	`string`	*RO/constructor*	Generation ID of the VM
`groups`	`VM_group ref set`	*RO/runtime*	VM groups associated with the VM
`guest_metrics`	`VM_guest_metrics ref`	*RO/runtime*	metrics associated with the running guest
`ha_always_run`	`bool`	*RO/constructor*	**Deprecated.** if true then the system will attempt to keep the VM running as much as possible.
`ha_restart_priority`	`string`	*RO/constructor*	has possible values: "best-effort" meaning "try to restart this VM if possible but don't consider the Pool to be overcommitted if this is not possible"; "restart" meaning "this VM should be restarted"; "" meaning "do not try to restart this VM"

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`hardware_platform_version`	`int`	*RW*	The host virtual hardware platform version the VM can run on
`has_vendor_device`	`bool`	*RO/constructor*	When an HVM guest starts, this controls the presence of the emulated C000 PCI device which triggers Windows Update to fetch or update PV drivers.
`HVM_boot_params`	`(string -> string) map`	*RW*	HVM boot params
`HVM_boot_policy`	`string`	*RO/constructor*	**Deprecated.** HVM boot policy
`HVM_shadow_multiplier`	`float`	*RO/constructor*	multiplier applied to the amount of shadow that will be made available to the guest
`is_a_snapshot`	`bool`	*RO/runtime*	true if this is a snapshot. Snapshotted VMs can never be started, they are used only for cloning other VMs
`is_a_template`	`bool`	*RW*	true if this is a template. Template VMs can never be started, they are used only for cloning other VMs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`is_control_domain`	`bool`	*RO/runtime*	true if this is a control domain (domain 0 or a driver domain)
`is_default_template`	`bool`	*RO/runtime*	true if this is a default template. Default template VMs can never be started or migrated, they are used only for cloning other VMs
`is_snapshot_from_vmpp`	`bool`	*RO/constructor*	**Removed.** true if this snapshot was created by the protection policy
`is_vmss_snapshot`	`bool`	*RO/constructor*	true if this snapshot was created by the snapshot schedule
`last_boot_CPU_flags`	`(string -> string) map`	*RO/constructor*	describes the CPU flags on which the VM was last booted
`last_booted_record`	`string`	*RO/constructor*	marshalled value containing VM record at time of last boot
`memory_dynamic_max`	`int`	*RO/constructor*	Dynamic maximum (bytes)
`memory_dynamic_min`	`int`	*RO/constructor*	Dynamic minimum (bytes)
`memory_overhead`	`int`	*RO/runtime*	Virtualization memory overhead (bytes).
`memory_static_max`	`int`	*RO/constructor*	Statically-set (i.e. absolute) maximum

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			(bytes). The value of this field at VM start time acts as a hard limit of the amount of memory a guest can use. New values only take effect on reboot.
`memory_static_min`	`int`	*RO/constructor*	Statically-set (i.e. absolute) mininum (bytes). The value of this field indicates the least amount of memory this VM can boot with without crashing.
`memory_target`	`int`	*RO/constructor*	**Deprecated.** Dynamically-set memory target (bytes). The value of this field indicates the current target for memory available to this VM.
`metrics`	`VM_metrics ref`	*RO/runtime*	metrics associated with this VM
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`NVRAM`	`(string -> string) map`	*RO/constructor*	initial value for guest NVRAM (containing UEFI variables, etc). Cannot be changed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			while the VM is running
`order`	`int`	*RO/constructor*	The point in the startup or shutdown sequence at which this VM will be started
`other_config`	`(string -> string) map`	*RO/constructor*	additional configuration
`parent`	`VM ref`	*RO/runtime*	Ref pointing to the parent of this VM
`PCI_bus`	`string`	*RW*	**Deprecated.** PCI bus path for pass-through devices
`pending_guidances`	`update_guidances set`	*RO/runtime*	The set of pending mandatory guidances after applying updates, which must be applied, as otherwise there may be, for example, VM failures
`pending_guidances_full`	`update_guidances set`	*RO/runtime*	The set of pending full guidances after applying updates, which a user should follow to make some updates, for example, specific hardware drivers or CPU features, fully effective, but the 'average user' doesn't need to

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`pending_guidances_recommended`	`update_guidances set`	*RO/runtime*	The set of pending recommended guidances after applying updates, which most users should follow to make the updates effective, but if not followed, will not cause a failure
`platform`	`(string -> string) map`	*RO/constructor*	platform-specific configuration
`power_state`	`vm_power_state`	*RO/constructor*	Current power state of the machine
`protection_policy`	`VMPP ref`	*RO/constructor*	**Deprecated.** Ref pointing to a protection policy for this VM
`PV_args`	`string`	*RW*	kernel command-line arguments
`PV_bootloader`	`string`	*RW*	name of or path to bootloader
`PV_bootloader_args`	`string`	*RW*	miscellaneous arguments for the bootloader
`PV_kernel`	`string`	*RW*	path to the kernel
`PV_legacy_args`	`string`	*RW*	to make Zurich guests boot
`PV_ramdisk`	`string`	*RW*	path to the initrd
`recommendations`	`string`	*RW*	An XML specification of recommended values and ranges

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			for properties of this VM
`reference_label`	`string`	*RO/constructor*	Textual reference to the template used to create a VM. This can be used by clients in need of an immutable reference to the template since the latter's uuid and name_label may change, for example, after a package installation or upgrade.
`requires_reboot`	`bool`	*RO/runtime*	Indicates whether a VM requires a reboot in order to update its configuration, for example, its memory allocation.
`resident_on`	`host ref`	*RO/runtime*	the host the VM is currently resident on
`scheduled_to_be_resident_on`	`host ref`	*RO/runtime*	the host on which the VM is due to be started/resumed/migrated. This acts as a memory reservation indicator
`secureboot_certificates_state`	`vm_secureboot_certificates_state`	*RO/runtime*	The state of the Secure Boot certificates, showing whether an update is available, already scheduled, or not needed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`shutdown_delay`	`int`	*RO/constructor*	The delay to wait before proceeding to the next order in the shutdown sequence (seconds)
`snapshot_info`	`(string -> string) map`	*RO/runtime*	Human-readable information concerning this snapshot
`snapshot_metadata`	`string`	*RO/runtime*	Encoded information about the VM's metadata this is a snapshot of
`snapshot_of`	`VM ref`	*RO/runtime*	Ref pointing to the VM this snapshot is of.
`snapshot_schedule`	`VMSS ref`	*RO/constructor*	Ref pointing to a snapshot schedule for this VM
`snapshot_time`	`datetime`	*RO/runtime*	Date/time when this snapshot was created.
`snapshots`	`VM ref set`	*RO/runtime*	List pointing to all the VM snapshots.
`start_delay`	`int`	*RO/constructor*	The delay to wait before proceeding to the next order in the startup sequence (seconds)
`suspend_SR`	`SR ref`	*RW*	The SR on which a suspend image is stored

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`suspend_VDI`	`VDI ref`	*RO/constructor*	The VDI that a suspend image is stored on. (Only has meaning if VM is currently suspended)
`tags`	`string set`	*RW*	user-specified tags for categorization purposes
`transportable_snapshot_id`	`string`	*RO/runtime*	Transportable ID of the snapshot VM
`user_version`	`int`	*RW*	Creators of VMs and templates may store version information here.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VBDs`	`VBD ref set`	*RO/runtime*	virtual block devices
`VCPUs_at_startup`	`int`	*RO/constructor*	Boot number of VCPUs
`VCPUs_max`	`int`	*RO/constructor*	Max number of VCPUs
`VCPUs_params`	`(string -> string) map`	*RW*	configuration parameters for the selected VCPU policy
`version`	`int`	*RO/constructor*	The number of times this VM has been recovered
`VGPUs`	`VGPU ref set`	*RO/runtime*	Virtual GPUs

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VIFs`	`VIF ref set`	*RO/runtime*	virtual network interfaces
`VTPMs`	`VTPM ref set`	*RO/runtime*	virtual TPMs
`VUSBs`	`VUSB ref set`	*RO/runtime*	virtual usb devices
`xenstore_data`	`(string -> string) map`	*RW*	data to be inserted into the xenstore tree (/local/domain/<domid>/vm-data) after the VM is created.

RPCs associated with class: VM

RPC name: addtags

Overview:

Add the given value to the tags field of the given VM. If the value is already in that Set, then do nothing.

Signature:

```
void add_tags (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to add

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: addtoblockedoperations

Overview:

Update list of operations which have been explicitly blocked and an error code

Signature:

```
void add_to_blocked_operations (session ref session_ref, VM ref self,
vm_operations key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`vm_operations`	`key`	Blocked operation
`string`	`value`	Error code

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoHVMbootparams

Overview:

Add the given keyvalue pair to the HVM/bootparams field of the given VM.

Signature:

```
void add_to_HVM_boot_params (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoNVRAM

Overview:

Signature:

```
void add_to_NVRAM (session ref session_ref, VM ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`key`	The key
`string`	`value`	The value

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void add_to_other_config (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoplatform

Overview:

Add the given keyvalue pair to the platform field of the given VM.

Signature:

```
void add_to_platform (session ref session_ref, VM ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoVCPUsparams

Overview:

Add the given keyvalue pair to the VCPUs/params field of the given VM.

Signature:

```
void add_to_VCPUs_params (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoVCPUsparamslive

Overview:

Add the given keyvalue pair to VM.VCPUsparams, and apply that value on the running VM

Signature:

```
void add_to_VCPUs_params_live (session ref session_ref, VM ref self,
string key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`key`	The key
`string`	`value`	The value

Minimum Role: vm-admin

Return Type: `void`

RPC name: addtoxenstoredata

Overview:

Add the given keyvalue pair to the xenstoredata field of the given VM.

Signature:

```
void add_to_xenstore_data (session ref session_ref, VM ref self, string
key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: assertagile

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Returns an error if the VM is not considered agile, for example, because it is tied to a resource local to a host

Signature:

```
void assert_agile (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: read-only

Return Type: `void`

RPC name: assertcanberecovered

Overview:

Assert whether all SRs required to recover this VM are available.

Signature:

```
void assert_can_be_recovered (session ref session_ref, VM ref self,
session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to recover
`session ref`	`session_to`	The session to which the VM is to be recovered.

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `VM_IS_PART_OF_AN_APPLIANCE`, `VM_REQUIRES_SR`

RPC name: assertcanboothere

Overview:

Returns an error if the VM could not boot on this host for some reason

Signature:

```
void assert_can_boot_here (session ref session_ref, VM ref self, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`host ref`	`host`	The host

Minimum Role: read-only

Return Type: `void`

Possible Error Codes: `HOST_NOT_ENOUGH_FREE_MEMORY`, `HOST_NOT_ENOUGH_PCPUS`, `NETWORK_SRIOV_INSUFFICIENT_CAPACITY`, `HOST_NOT_LIVE`, `HOST_DISABLED`, `HOST_CANNOT_ATTACH_NETWORK`, `VM_HVM_REQUIRED`, `VM_REQUIRES_GPU`, `VM_REQUIRES_IOMMU`, `VM_REQUIRES_NETWORK`, `VM_REQUIRES_SR`, `VM_REQUIRES_VGPU`, `VM_HOST_INCOMPATIBLE_VERSION`, `VM_HOST_INCOMPATIBLE_VIRTUAL_HARDWARE_PLATFORM_VERSION`, `INVALID_VALUE`, `MEMORY_CONSTRAINT_VIOLATION`, `OPERATION_NOT_ALLOWED`, `VALUE_NOT_SUPPORTED`, `VM_INCOMPATIBLE_WITH_THIS_HOST`

RPC name: assertcanmigrate

Overview:

Assert whether a VM can be migrated to the specified destination.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void assert_can_migrate (session ref session_ref, VM ref vm, (string ->
string) map dest, bool live, (VDI ref -> SR ref) map vdi_map, (VIF ref ->
network ref) map vif_map, (string -> string) map options, (VGPU ref ->
GPU_group ref) map vgpu_map)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`(string -> string) map`	`dest`	The result of a Host.migrate_receive call.
`bool`	`live`	Live migration
`(VDI ref -> SR ref) map`	`vdi_map`	Map of source VDI to destination SR
`(VIF ref -> network ref) map`	`vif_map`	Map of source VIF to destination network
`(string -> string) map`	`options`	Other parameters
`(VGPU ref -> GPU_group ref) map`	`vgpu_map`	Map of source vGPU to destination GPU group

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `LICENCE_RESTRICTION`

RPC name: assertoperationvalid

Overview:

Check to see whether this operation is acceptable in the current state of the system, raising an error if the operation is invalid for some reason

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void assert_operation_valid (session ref session_ref, VM ref self,
vm_operations op)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`vm_operations`	`op`	proposed operation

Minimum Role: read-only

Return Type: `void`

RPC name: callhostplugin

Overview:

Call an API plugin on the host where this vm resides

Signature:

```
string call_host_plugin (session ref session_ref, VM ref vm, string
plugin, string fn, (string -> string) map args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The vm
`string`	`plugin`	The name of the plugin
`string`	`fn`	The name of the function within the plugin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`args`	Arguments for the function

Minimum Role: vm-operator

Return Type: `string`

Result from the plugin

RPC name: callplugin

Overview:

Call an API plugin on this vm

Signature:

```
string call_plugin (session ref session_ref, VM ref vm, string plugin,
string fn, (string -> string) map args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The vm
`string`	`plugin`	The name of the plugin
`string`	`fn`	The name of the function within the plugin
`(string -> string) map`	`args`	Arguments for the function

Minimum Role: vm-operator

Return Type: `string`

Result from the plugin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: checkpoint

Overview:

Checkpoints the specified VM, making a new VM. Checkpoint automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write and saves the memory image as well.

Signature:

```
VM ref checkpoint (session ref session_ref, VM ref vm, string new_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be checkpointed
`string`	`new_name`	The name of the checkpointed VM

Minimum Role: vm-power-admin

Return Type: `VM ref`

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `VM_CHECKPOINT_SUSPEND_FAILED`, `VM_CHECKPOINT_RESUME_FAILED`

RPC name: cleanreboot

Overview:

Attempt to cleanly shutdown the specified VM Note: this may not be supportedfor example, if a guest agent is not installed. This can only be called when the specified VM is in the Running state.

Signature:

```
void clean_reboot (session ref session_ref, VM ref vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to shutdown

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: cleanshutdown

Overview:

Attempt to cleanly shutdown the specified VM. Note: this may not be supportedfor example, if a guest agent is not installed. This can only be called when the specified VM is in the Running state.

Signature:

```
void clean_shutdown (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to shutdown

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: clone

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Clones the specified VM, making a new VM. Clone automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write. This function can only be called when the VM is in the Halted State.

Signature:

```
VM ref clone (session ref session_ref, VM ref vm, string new_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be cloned
`string`	`new_name`	The name of the cloned VM

Minimum Role: vm-admin

Return Type: `VM ref`

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: computememoryoverhead

Overview:

Computes the virtualization memory overhead of a VM.

Signature:

```
int compute_memory_overhead (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`vm`	The VM for which to compute the memory overhead

Minimum Role: read-only

Return Type: `int`

the virtualization memory overhead of the VM.

RPC name: copy

Overview:

Copied the specified VM, making a new VM. Unlike clone, copy does not exploits the capabilities of the underlying storage repository in which the VM's disk images are stored. Instead, copy guarantees that the disk images of the newly created VM will be 'full disks' i.e. not part of a CoW chain. This function can only be called when the VM is in the Halted State.

Signature:

```
VM ref copy (session ref session_ref, VM ref vm, string new_name, SR ref
sr)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be copied
`string`	`new_name`	The name of the copied VM
`SR ref`	`sr`	An SR to copy all the VM's disks into (if an invalid reference then it uses the existing SRs)

Minimum Role: vm-admin

Return Type: `VM ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: copybiosstrings

Overview:

Copy the BIOS strings from the given host to this VM

Signature:

```
void copy_bios_strings (session ref session_ref, VM ref vm, host ref host)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to modify
`host ref`	`host`	The host to copy the BIOS strings from

Minimum Role: vm-admin

Return Type: `void`

RPC name: create

Overview:

NOT RECOMMENDED VM.clone or VM.copy or VM.import is a better choice in almost all situations. The standard way to obtain a new VM is to call VM.clone on a template VM, then call VM.provision on the new clone. Caution: if VM.create is used and then the new VM is attached to a virtual disc that has an operating system already installed, then there is no guarantee that the operating system will boot and run. Any software that calls VM.create on a future version of this API may fail or give unexpected results. For example this could happen if an additional parameter were added to VM.create. VM.create is intended only for use in the automatic creation of the system VM templates. It creates a new VM instance, and returns its handle.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM ref create (session ref session_ref, VM record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VM ref`

reference to the newly created object

RPC name: createnewblob

Overview:

Create a placeholder for a named binary blob of data that is associated with this VM

Signature:

```
blob ref create_new_blob (session ref session_ref, VM ref vm, string name,
string mime_type, bool public)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`string`	`name`	The name associated with the blob

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`mime_type`	The mime type for the data. Empty string translates to application/octet-stream
`bool`	`public`	True if the blob should be publicly available

Minimum Role: vm-power-admin

Return Type: `blob ref`

The reference of the blob, needed for populating its data

RPC name: destroy

Overview:

Destroy the specified VM. The VM is completely removed from the system. This function can only be called when the VM is in the Halted State.

Signature:

```
void destroy (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: vm-admin

Return Type: `void`

RPC name: forgetdatasourcearchives

Overview:

Forget the recorded statistics related to the specified data source

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void forget_data_source_archives (session ref session_ref, VM ref self,
string data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`data_source`	The data source whose archives are to be forgotten

Minimum Role: vm-admin

Return Type: `void`

RPC name: getactionsaftercrash

Overview:

Get the actions/aftercrash field of the given VM.

Signature:

```
on_crash_behaviour get_actions_after_crash (session ref session_ref, VM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_crash_behaviour`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getactionsafterreboot

Overview:

Get the actions/afterreboot field of the given VM.

Signature:

```
on_normal_exit get_actions_after_reboot (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_normal_exit`

value of the field

RPC name: getactionsaftershutdown

Overview:

Get the actions/aftershutdown field of the given VM.

Signature:

```
on_normal_exit get_actions_after_shutdown (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_normal_exit`

value of the field

RPC name: getactionsaftersoftreboot

Overview:

Get the actions/aftersoftreboot field of the given VM.

Signature:

```
on_softreboot_behavior get_actions_after_softreboot (session ref
session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `on_softreboot_behavior`

value of the field

RPC name: getaffinity

Overview:

Get the affinity field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref get_affinity (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getall

Overview:

Return a list of all the VMs known to the system.

Signature:

```
VM ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VM references to VM records for all VMs known to the system.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
(VM ref -> VM record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM ref -> VM record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VM.

Signature:

```
vm_operations set get_allowed_operations (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_operations set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallowedVBDdevices

Overview:

Returns a list of the allowed values that a VBD device field can take

Signature:

```
string set get_allowed_VBD_devices (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to query

Minimum Role: read-only

Return Type: `string set`

The allowed values

RPC name: getallowedVIFdevices

Overview:

Returns a list of the allowed values that a VIF device field can take

Signature:

```
string set get_allowed_VIF_devices (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to query

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string set`

The allowed values

RPC name: getappliance

Overview:

Get the appliance field of the given VM.

Signature:

```
VM_appliance ref get_appliance (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_appliance ref`

value of the field

RPC name: getattachedPCIs

Overview:

Get the attachedPCIs field of the given VM.

Signature:

```
PCI ref set get_attached_PCIs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `PCI ref set`

value of the field

RPC name: getbiosstrings

Overview:

Get the biosstrings field of the given VM.

Signature:

```
(string -> string) map get_bios_strings (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getblobs

Overview:

Get the blobs field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> blob ref) map get_blobs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> blob ref) map`

value of the field

RPC name: getblockedoperations

Overview:

Get the blockedoperations field of the given VM.

Signature:

```
(vm_operations -> string) map get_blocked_operations (session ref
session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(vm_operations -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbootrecord

This message is deprecated.

Overview:

Returns a record describing the VM's dynamic state, initialised when the VM boots and updated to reflect runtime configuration changes, for example, CPU hotplug

Signature:

```
VM record get_boot_record (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM whose boot-time state to return

Minimum Role: read-only

Return Type: `VM record`

A record describing the VM

RPC name: getbynamelabel

Overview:

Get all the VM instances with the given label.

Signature:

```
VM ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VM ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VM instance with the specified UUID.

Signature:

```
VM ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM ref`

reference to the object

RPC name: getchildren

Overview:

Get the children field of the given VM.

Signature:

```
VM ref set get_children (session ref session_ref, VM ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: getconsoles

Overview:

Get the consoles field of the given VM.

Signature:

```
console ref set get_consoles (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `console ref set`

value of the field

RPC name: getcooperative

This message is deprecated.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return true if the VM is currently 'cooperative' i.e. is expected to reach a balloon target and actually has done

Signature:

```
bool get_cooperative (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: read-only

Return Type: `bool`

true if the VM is currently 'cooperative'; false otherwise

RPC name: getcrashdumps

Overview:

Get the crashdumps field of the given VM.

Signature:

```
crashdump ref set get_crash_dumps (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `crashdump ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VM.

Signature:

```
(string -> vm_operations) map get_current_operations (session ref
session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vm_operations) map`

value of the field

RPC name: getdatasources

Overview:

Signature:

```
data_source record set get_data_sources (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to interrogate

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `data_source record set`

A set of data sources

RPC name: getdomaintype

Overview:

Get the domaintype field of the given VM.

Signature:

```
domain_type get_domain_type (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `domain_type`

value of the field

RPC name: getdomarch

Overview:

Get the domarch field of the given VM.

Signature:

```
string get_domarch (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getdomid

Overview:

Get the domid field of the given VM.

Signature:

```
int get_domid (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getgenerationid

Overview:

Get the generationid field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_generation_id (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getgroups

Overview:

Get the groups field of the given VM.

Signature:

```
VM_group ref set get_groups (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_group ref set`

value of the field

RPC name: getguestmetrics

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the guestmetrics field of the given VM.

Signature:

```
VM_guest_metrics ref get_guest_metrics (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_guest_metrics ref`

value of the field

RPC name: gethaalwaysrun

This message is deprecated.

Overview:

Get the haalwaysrun field of the given VM.

Signature:

```
bool get_ha_always_run (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getharestartpriority

Overview:

Get the harestartpriority field of the given VM.

Signature:

```
string get_ha_restart_priority (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: gethardwareplatformversion

Overview:

Get the hardwareplatformversion field of the given VM.

Signature:

```
int get_hardware_platform_version (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: gethasvendordevice

Overview:

Get the hasvendordevice field of the given VM.

Signature:

```
bool get_has_vendor_device (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getHVMbootparams

Overview:

Get the HVM/bootparams field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_HVM_boot_params (session ref session_ref, VM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getHVMbootpolicy

This message is deprecated.

Overview:

Get the HVM/bootpolicy field of the given VM.

Signature:

```
string get_HVM_boot_policy (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getHVMshadowmultiplier

Overview:

Get the HVM/shadowmultiplier field of the given VM.

Signature:

```
float get_HVM_shadow_multiplier (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `float`

value of the field

RPC name: getisasnapshot

Overview:

Get the isasnapshot field of the given VM.

Signature:

```
bool get_is_a_snapshot (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `bool`

value of the field

RPC name: getisatemplate

Overview:

Get the isatemplate field of the given VM.

Signature:

```
bool get_is_a_template (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getiscontroldomain

Overview:

Get the iscontroldomain field of the given VM.

Signature:

```
bool get_is_control_domain (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisdefaulttemplate

Overview:

Get the isdefaulttemplate field of the given VM.

Signature:

```
bool get_is_default_template (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getissnapshotfromvmpp

This message is removed.

Overview:

Get the issnapshotfromvmpp field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_is_snapshot_from_vmpp (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisvmsssnapshot

Overview:

Get the isvmsssnapshot field of the given VM.

Signature:

```
bool get_is_vmss_snapshot (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getlastbootCPUflags

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the lastbootCPUflags field of the given VM.

Signature:

```
(string -> string) map get_last_boot_CPU_flags (session ref session_ref,
VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlastbootedrecord

Overview:

Get the lastbootedrecord field of the given VM.

Signature:

```
string get_last_booted_record (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

value of the field

RPC name: getmemorydynamicmax

Overview:

Get the memory/dynamicmax field of the given VM.

Signature:

```
int get_memory_dynamic_max (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorydynamicmin

Overview:

Get the memory/dynamicmin field of the given VM.

Signature:

```
int get_memory_dynamic_min (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemoryoverhead

Overview:

Get the memory/overhead field of the given VM.

Signature:

```
int get_memory_overhead (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorystaticmax

Overview:

Get the memory/staticmax field of the given VM.

Signature:

```
int get_memory_static_max (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorystaticmin

Overview:

Get the memory/staticmin field of the given VM.

Signature:

```
int get_memory_static_min (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmemorytarget

This message is deprecated.

Overview:

Get the memory/target field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
int get_memory_target (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getmetrics

Overview:

Get the metrics field of the given VM.

Signature:

```
VM_metrics ref get_metrics (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_metrics ref`

value of the field

RPC name: getnamedescription

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the name/description field of the given VM.

Signature:

```
string get_name_description (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VM.

Signature:

```
string get_name_label (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getNVRAM

Overview:

Get the NVRAM field of the given VM.

Signature:

```
(string -> string) map get_NVRAM (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getorder

Overview:

Get the order field of the given VM.

Signature:

```
int get_order (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `int`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VM.

Signature:

```
(string -> string) map get_other_config (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getparent

Overview:

Get the parent field of the given VM.

Signature:

```
VM ref get_parent (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: getPCIbus

This message is deprecated.

Overview:

Get the PCIbus field of the given VM.

Signature:

```
string get_PCI_bus (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getpendingguidances

Overview:

Get the pendingguidances field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
update_guidances set get_pending_guidances (session ref session_ref, VM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getpendingguidancesfull

Overview:

Get the pendingguidancesfull field of the given VM.

Signature:

```
update_guidances set get_pending_guidances_full (session ref session_ref,
VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getpendingguidancesrecommended

Overview:

Get the pendingguidancesrecommended field of the given VM.

Signature:

```
update_guidances set get_pending_guidances_recommended (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `update_guidances set`

value of the field

RPC name: getplatform

Overview:

Get the platform field of the given VM.

Signature:

```
(string -> string) map get_platform (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getpossiblehosts

Overview:

Return the list of hosts on which this VM may run.

Signature:

```
host ref set get_possible_hosts (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM

Minimum Role: read-only

Return Type: `host ref set`

The possible hosts

RPC name: getpowerstate

Overview:

Get the powerstate field of the given VM.

Signature:

```
vm_power_state get_power_state (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_power_state`

value of the field

RPC name: getprotectionpolicy

This message is deprecated.

Overview:

Get the protectionpolicy field of the given VM.

Signature:

```
VMPP ref get_protection_policy (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VMPP ref`

value of the field

RPC name: getPVargs

Overview:

Get the PV/args field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string get_PV_args (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVbootloader

Overview:

Get the PV/bootloader field of the given VM.

Signature:

```
string get_PV_bootloader (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVbootloaderargs

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the PV/bootloaderargs field of the given VM.

Signature:

```
string get_PV_bootloader_args (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVkernel

Overview:

Get the PV/kernel field of the given VM.

Signature:

```
string get_PV_kernel (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getPVlegacyargs

Overview:

Get the PV/legacyargs field of the given VM.

Signature:

```
string get_PV_legacy_args (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getPVramdisk

Overview:

Get the PV/ramdisk field of the given VM.

Signature:

```
string get_PV_ramdisk (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getrecommendations

Overview:

Get the recommendations field of the given VM.

Signature:

```
string get_recommendations (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VM.

Signature:

```
VM record get_record (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM record`

all fields from the object

RPC name: getreferencelabel

Overview:

Get the referencelabel field of the given VM.

Signature:

```
string get_reference_label (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrequiresreboot

Overview:

Get the requiresreboot field of the given VM.

Signature:

```
bool get_requires_reboot (session ref session_ref, VM ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getresidenton

Overview:

Get the residenton field of the given VM.

Signature:

```
host ref get_resident_on (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getscheduledtoberesidenton

Overview:

Get the scheduledtoberesidenton field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
host ref get_scheduled_to_be_resident_on (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `host ref`

value of the field

RPC name: getsecurebootcertificatesstate

Overview:

Get the securebootcertificatesstate field of the given VM.

Signature:

```
vm_secureboot_certificates_state get_secureboot_certificates_state
(session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_secureboot_certificates_state`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getsecurebootreadiness

Overview:

Return the secureboot readiness of the VM

Signature:

```
vm_secureboot_readiness get_secureboot_readiness (session ref session_ref,
VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: pool-admin

Return Type: `vm_secureboot_readiness`

The secureboot readiness of the VM

RPC name: getshutdowndelay

Overview:

Get the shutdowndelay field of the given VM.

Signature:

```
int get_shutdown_delay (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getsnapshotinfo

Overview:

Get the snapshotinfo field of the given VM.

Signature:

```
(string -> string) map get_snapshot_info (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getsnapshotmetadata

Overview:

Get the snapshotmetadata field of the given VM.

Signature:

```
string get_snapshot_metadata (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getsnapshotof

Overview:

Get the snapshotof field of the given VM.

Signature:

```
VM ref get_snapshot_of (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: getsnapshotschedule

Overview:

Get the snapshotschedule field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VMSS ref get_snapshot_schedule (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VMSS ref`

value of the field

RPC name: getsnapshottime

Overview:

Get the snapshottime field of the given VM.

Signature:

```
datetime get_snapshot_time (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getsnapshots

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the snapshots field of the given VM.

Signature:

```
VM ref set get_snapshots (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: getSRsrequiredforrecovery

Overview:

List all the SR's that are required for the VM to be recovered

Signature:

```
SR ref set get_SRs_required_for_recovery (session ref session_ref, VM ref
self, session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM for which the SRs have to be recovered
`session ref`	`session_to`	The session to which the SRs of the VM have to be recovered.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `SR ref set`

refs for SRs required to recover the VM

RPC name: getstartdelay

Overview:

Get the startdelay field of the given VM.

Signature:

```
int get_start_delay (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getsuspendSR

Overview:

Get the suspendSR field of the given VM.

Signature:

```
SR ref get_suspend_SR (session ref session_ref, VM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `SR ref`

value of the field

RPC name: getsuspendVDI

Overview:

Get the suspendVDI field of the given VM.

Signature:

```
VDI ref get_suspend_VDI (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VDI ref`

value of the field

RPC name: gettags

Overview:

Get the tags field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
string set get_tags (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: gettransportablesnapshotid

Overview:

Get the transportablesnapshotid field of the given VM.

Signature:

```
string get_transportable_snapshot_id (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getuserversion

Overview:

Get the userversion field of the given VM.

Signature:

```
int get_user_version (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VM.

Signature:

```
string get_uuid (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getVBDs

Overview:

Get the VBDs field of the given VM.

Signature:

```
VBD ref set get_VBDs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VBD ref set`

value of the field

RPC name: getVCPUsatstartup

Overview:

Get the VCPUs/atstartup field of the given VM.

Signature:

```
int get_VCPUs_at_startup (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVCPUsmax

Overview:

Get the VCPUs/max field of the given VM.

Signature:

```
int get_VCPUs_max (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVCPUsparams

Overview:

Get the VCPUs/params field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_VCPUs_params (session ref session_ref, VM ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getversion

Overview:

Get the version field of the given VM.

Signature:

```
int get_version (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getVGPUs

Overview:

Get the VGPUs field of the given VM.

Signature:

```
VGPU ref set get_VGPUs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VGPU ref set`

value of the field

RPC name: getVIFs

Overview:

Get the VIFs field of the given VM.

Signature:

```
VIF ref set get_VIFs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VIF ref set`

value of the field

RPC name: getVTPMs

Overview:

Get the VTPMs field of the given VM.

Signature:

```
VTPM ref set get_VTPMs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VTPM ref set`

value of the field

RPC name: getVUSBs

Overview:

Get the VUSBs field of the given VM.

Signature:

```
VUSB ref set get_VUSBs (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VUSB ref set`

value of the field

RPC name: getxenstoredata

Overview:

Get the xenstoredata field of the given VM.

Signature:

```
(string -> string) map get_xenstore_data (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: hardreboot

Overview:

Stop executing the specified VM without attempting a clean shutdown and immediately restart the VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void hard_reboot (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to reboot

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: hardshutdown

Overview:

Stop executing the specified VM without attempting a clean shutdown.

Signature:

```
void hard_shutdown (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to destroy

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: import

Overview:

Import an XVA from a URI

Signature:

```
VM ref set import (session ref session_ref, string url, SR ref sr, bool full_restore, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`url`	The URL of the XVA file
`SR ref`	`sr`	The destination SR for the disks
`bool`	`full_restore`	Perform a full restore
`bool`	`force`	Force the import

Minimum Role: pool-operator

Return Type: `VM ref set`

Imported VM reference

RPC name: importconvert

Overview:

Import using a conversion service.

Signature:

```
void import_convert (session ref session_ref, string type, string username, string password, SR ref sr, (string -> string) map remote_config)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`type`	Type of the conversion
`string`	`username`	Admin username on the host
`string`	`password`	Password on the host
`SR ref`	`sr`	The destination SR
`(string -> string) map`	`remote_config`	Remote configuration options

Minimum Role: vm-admin

Return Type: `void`

RPC name: maximisememory

Overview:

Returns the maximum amount of guest memory which will fit, together with overheads, in the supplied amount of physical memory. If 'exact' is true then an exact calculation is performed using the VM's current settings. If 'exact' is false then a more conservative approximation is used

Signature:

```
int maximise_memory (session ref session_ref, VM ref self, int total, bool approximate)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`total`	Total amount of physical RAM to fit within

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`bool`	`approximate`	If false the limit is calculated with the guest's current exact configuration. Otherwise a more approximate calculation is performed

Minimum Role: read-only

Return Type: `int`

The maximum possible staticmax

RPC name: migratesend

Overview:

Migrate the VM to another host. This can only be called when the specified VM is in the Running state.

Signature:

```
VM ref migrate_send (session ref session_ref, VM ref vm, (string ->
string) map dest, bool live, (VDI ref -> SR ref) map vdi_map, (VIF ref ->
network ref) map vif_map, (string -> string) map options, (VGPU ref ->
GPU_group ref) map vgpu_map, (VDI ref -> string) map vdi_format_map)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`(string -> string) map`	`dest`	The result of a Host.migrate_receive call.
`bool`	`live`	Live migration
`(VDI ref -> SR ref) map`	`vdi_map`	Map of source VDI to destination SR

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(VIF ref -> network ref) map`	`vif_map`	Map of source VIF to destination network
`(string -> string) map`	`options`	Other parameters
`(VGPU ref -> GPU_group ref) map`	`vgpu_map`	Map of source vGPU to destination GPU group
`(VDI ref -> string) map`	`vdi_format_map`	Map of source VDI to it's expected type on destination

Minimum Role: vm-power-admin

Return Type: `VM ref`

The reference of the newly created VM in the destination pool

Possible Error Codes: `VM_BAD_POWER_STATE`, `LICENCE_RESTRICTION`

RPC name: pause

Overview:

Pause the specified VM. This can only be called when the specified VM is in the Running state.

Signature:

```
void pause (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to pause

Minimum Role: vm-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: poolmigrate

Overview:

Migrate a VM to another Host.

Signature:

```
void pool_migrate (session ref session_ref, VM ref vm, host ref host,
(string -> string) map options)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to migrate
`host ref`	`host`	The target host
`(string -> string) map`	`options`	Extra configuration operations: force, live, copy, compress. Each is a boolean option, taking 'true' or 'false' as a value. Option 'compress' controls the use of stream compression during migration.

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `VM_IS_TEMPLATE`, `OPERATION_NOT_ALLOWED`, `VM_BAD_POWER_STATE`

RPC name: powerstatereset

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Reset the powerstate of the VM to halted in the database only. Used to recover from slave failures in pooling scenarios by resetting the powerstates of VMs running on dead slaves to halted. This is a potentially dangerous operation; use with care.

Signature:

```
void power_state_reset (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to reset

Minimum Role: pool-operator

Return Type: `void`

RPC name: provision

Overview:

Inspects the disk configuration contained within the VM's otherconfig, creates VDIs and VBDs and then executes any applicable postinstall script.

Signature:

```
void provision (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be provisioned

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: querydatasource

Overview:

Query the latest value of the specified data source

Signature:

```
float query_data_source (session ref session_ref, VM ref self, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`data_source`	The data source to query

Minimum Role: read-only

Return Type: `float`

The latest value, averaged over the last 5 seconds

RPC name: queryservices

Overview:

Query the system services advertised by this VM and register them. This can only be applied to a system domain.

Signature:

```
(string -> string) map query_services (session ref session_ref, VM ref
self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: pool-admin

Return Type: `(string -> string) map`

map of service type to name

RPC name: recorddatasource

Overview:

Start recording the specified data source

Signature:

```
void record_data_source (session ref session_ref, VM ref self, string
data_source)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`data_source`	The data source to record

Minimum Role: vm-admin

Return Type: `void`

RPC name: recover

Overview:

Recover the VM

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void recover (session ref session_ref, VM ref self, session ref
session_to, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to recover
`session ref`	`session_to`	The session to which the VM is to be recovered.
`bool`	`force`	Whether the VM should replace newer versions of itself.

Minimum Role: read-only

Return Type: `void`

RPC name: removefromblockedoperations

Overview:

Update list of operations which have been explicitly blocked and an error code

Signature:

```
void remove_from_blocked_operations (session ref session_ref, VM ref self,
vm_operations key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`vm_operations`	`key`	Blocked operation

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromHVMbootparams

Overview:

Remove the given key and its corresponding value from the HVM/bootparams field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_HVM_boot_params (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromNVRAM

Overview:

Signature:

```
void remove_from_NVRAM (session ref session_ref, VM ref self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM
`string`	`key`	The key

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key of entry to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromplatform

Overview:

Remove the given key and its corresponding value from the platform field of the given VM. If the key is not in that Map, then do nothing.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_platform (session ref session_ref, VM ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key of entry to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removefromVCPUsparams

Overview:

Remove the given key and its corresponding value from the VCPUs/params field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_VCPUs_params (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removefromxenstoredata

Overview:

Remove the given key and its corresponding value from the xenstoredata field of the given VM. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_xenstore_data (session ref session_ref, VM ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: removetags

Overview:

Remove the given value from the tags field of the given VM. If the value is not in that Set, then do nothing.

Signature:

```
void remove_tags (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object
`string`	`value`	Value to remove

Minimum Role: vm-operator

Return Type: `void`

RPC name: restartdevicemodels

Overview:

Restart device models of the VM

Signature:

```
void restart_device_models (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `VM_IS_TEMPLATE`, `OPERATION_NOT_ALLOWED`, `VM_BAD_POWER_STATE`

RPC name: resume

Overview:

Awaken the specified VM and resume it. This can only be called when the specified VM is in the Suspended state.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void resume (session ref session_ref, VM ref vm, bool start_paused, bool
force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to resume
`bool`	`start_paused`	Resume VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to resume. If this flag is false then the VM may fail pre-resume safety checks (for example, if the CPU the VM was running on looks substantially different to the current one)

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: resumeon

Overview:

Awaken the specified VM and resume it on a particular Host. This can only be called when the specified VM is in the Suspended state.

Signature:

```
void resume_on (session ref session_ref, VM ref vm, host ref host, bool
start_paused, bool force)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to resume
`host ref`	`host`	The Host on which to resume the VM
`bool`	`start_paused`	Resume VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to resume. If this flag is false then the VM may fail pre-resume safety checks (for example, if the CPU the VM was running on looks substantially different to the current one)

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: retrievewlbrecommendations

Overview:

Returns mapping of hosts to ratings, indicating the suitability of starting the VM at that location according to wlb. Rating is replaced with an error if the VM cannot boot there.

Signature:

```
(host ref -> string set) map retrieve_wlb_recommendations (session ref
session_ref, VM ref vm)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM

Minimum Role: read-only

Return Type: `(host ref -> string set) map`

The potential hosts and their corresponding recommendations or errors

RPC name: revert

Overview:

Reverts the specified VM to a previous state.

Signature:

```
void revert (session ref session_ref, VM ref snapshot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`snapshot`	The snapshotted state that we revert to

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `SR_FULL`, `VM_REVERT_FAILED`

RPC name: sendsysrq

Overview:

Send the given key as a sysrq to this VM. The key is specified as a single character a String of length 1. This can only be called when the specified VM is in the Running state.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void send_sysrq (session ref session_ref, VM ref vm, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`string`	`key`	The key to send

Minimum Role: pool-admin

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`

RPC name: sendtrigger

Overview:

Send the named trigger to this VM. This can only be called when the specified VM is in the Running state.

Signature:

```
void send_trigger (session ref session_ref, VM ref vm, string trigger)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM
`string`	`trigger`	The trigger to send

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`

RPC name: setactionsaftercrash

Overview:

Sets the actionsaftercrash parameter

Signature:

```
void set_actions_after_crash (session ref session_ref, VM ref self,
on_crash_behaviour value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to set
`on_crash_behaviour`	`value`	The new value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setactionsafterreboot

Overview:

Set the actions/afterreboot field of the given VM.

Signature:

```
void set_actions_after_reboot (session ref session_ref, VM ref self,
on_normal_exit value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`on_normal_exit`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setactionsaftershutdown

Overview:

Set the actions/aftershutdown field of the given VM.

Signature:

```
void set_actions_after_shutdown (session ref session_ref, VM ref self,
on_normal_exit value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`on_normal_exit`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setactionsaftersoftreboot

Overview:

Set the actions/aftersoftreboot field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_actions_after_softreboot (session ref session_ref, VM ref self,
on_softreboot_behavior value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`on_softreboot_behavior`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setaffinity

Overview:

Set the affinity field of the given VM.

Signature:

```
void set_affinity (session ref session_ref, VM ref self, host ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`host ref`	`value`	New value to set

Minimum Role: vm-power-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setappliance

Overview:

Assign this VM to an appliance.

Signature:

```
void set_appliance (session ref session_ref, VM ref self, VM_appliance ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to assign to an appliance.
`VM_appliance ref`	`value`	The appliance to which this VM should be assigned.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbiosstrings

Overview:

Set custom BIOS strings to this VM. VM will be given a default set of BIOS strings, only some of which can be overridden by the supplied values. Allowed keys are: 'biosvendor', 'biosversion', 'systemmanufacturer', 'systemproductname', 'systemversion', 'systemserialnumber', 'enclosureassettag', 'baseboardmanufacturer', 'baseboardproductname', 'baseboardversion', 'baseboardserialnumber', 'baseboardassettag', 'baseboardlocationinchassis', 'enclosureassettag'

Signature:

```
void set_bios_strings (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`(string -> string) map`	`value`	The custom BIOS strings as a list of key-value pairs

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `VM_BIOS_STRINGS_ALREADY_SET`, `INVALID_VALUE`

RPC name: setblockedoperations

Overview:

Update list of operations which have been explicitly blocked and an error code

Signature:

```
void set_blocked_operations (session ref session_ref, VM ref self,
(vm_operations -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`(vm_operations -> string) map`	`value`	Blocked operations

Minimum Role: vm-admin

Return Type: `void`

RPC name: setdomaintype

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the VM.domaintype field of the given VM, which will take effect when it is next started

Signature:

```
void set_domain_type (session ref session_ref, VM ref self, domain_type
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`domain_type`	`value`	The new domain type

Minimum Role: vm-admin

Return Type: `void`

RPC name: setgroups

Overview:

Associate this VM with VM groups.

Signature:

```
void set_groups (session ref session_ref, VM ref self, VM_group ref set
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`VM_group ref set`	`value`	The VM groups to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `OPERATION_NOT_ALLOWED`

RPC name: sethaalwaysrun

This message is deprecated.

Overview:

Set the value of the haalwaysrun

Signature:

```
void set_ha_always_run (session ref session_ref, VM ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`bool`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: setharestartpriority

Overview:

Set the value of the harestartpriority field

Signature:

```
void set_ha_restart_priority (session ref session_ref, VM ref self, string
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: sethardwareplatformversion

Overview:

Set the hardwareplatformversion field of the given VM.

Signature:

```
void set_hardware_platform_version (session ref session_ref, VM ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: sethasvendordevice

Overview:

Controls whether, when the VM starts in HVM mode, its virtual hardware will include the emulated PCI device for which drivers may be available through Windows Update. Usually this should never

© 1997–2026 Citrix Systems, Inc. All rights reserved.

be changed on a VM on which Windows has been installed: changing it on such a VM is likely to lead to a crash on next start.

Signature:

```
void set_has_vendor_device (session ref session_ref, VM ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM on which to set this flag
`bool`	`value`	True to provide the vendor PCI device.

Minimum Role: vm-admin

Return Type: `void`

RPC name: setHVMbootparams

Overview:

Set the HVM/bootparams field of the given VM.

Signature:

```
void set_HVM_boot_params (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setHVMbootpolicy

This message is deprecated.

Overview:

Set the VM.HVMbootpolicy field of the given VM, which will take effect when it is next started

Signature:

```
void set_HVM_boot_policy (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`value`	The new HVM boot policy

Minimum Role: vm-admin

Return Type: `void`

RPC name: setHVMshadowmultiplier

Overview:

Set the shadow memory multiplier on a halted VM

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_HVM_shadow_multiplier (session ref session_ref, VM ref self,
float value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`float`	`value`	The new shadow memory multiplier to set

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setisatemplate

Overview:

Set the isatemplate field of the given VM.

Signature:

```
void set_is_a_template (session ref session_ref, VM ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setmemory

Overview:

Set the memory allocation of this VM. Sets all of memorystaticmax, memorydynamicmin, and memorydynamicmax to the given value, and leaves memorystaticmin untouched.

Signature:

```
void set_memory (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	The new memory allocation (bytes).

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorydynamicmax

Overview:

Set the value of the memorydynamicmax field

Signature:

```
void set_memory_dynamic_max (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_dynamic_max

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorydynamicmin

Overview:

Set the value of the memorydynamicmin field

Signature:

```
void set_memory_dynamic_min (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_dynamic_min

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorydynamicrange

Overview:

Set the minimum and maximum amounts of physical memory the VM is allowed to use.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_memory_dynamic_range (session ref session_ref, VM ref self, int
min, int max)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`min`	The new minimum value
`int`	`max`	The new maximum value

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorylimits

Overview:

Set the memory limits of this VM.

Signature:

```
void set_memory_limits (session ref session_ref, VM ref self, int
static_min, int static_max, int dynamic_min, int dynamic_max)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`static_min`	The new value of memory_static_min.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`int`	`static_max`	The new value of memory_static_max.
`int`	`dynamic_min`	The new value of memory_dynamic_min.
`int`	`dynamic_max`	The new value of memory_dynamic_max.

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorystaticmax

Overview:

Set the value of the memorystaticmax field

Signature:

```
void set_memory_static_max (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_static_max

Minimum Role: vm-power-admin

Return Type: `void`

Possible Error Codes: `HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setmemorystaticmin

Overview:

Set the value of the memorystaticmin field

Signature:

```
void set_memory_static_min (session ref session_ref, VM ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM to modify
`int`	`value`	The new value of memory_static_min

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorystaticrange

Overview:

Set the static ie boottime range of virtual memory that the VM is allowed to use.

Signature:

```
void set_memory_static_range (session ref session_ref, VM ref self, int
min, int max)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM
`int`	`min`	The new minimum value
`int`	`max`	The new maximum value

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setmemorytargetlive

This message is deprecated.

Overview:

Set the memory target for a running VM

Signature:

```
void set_memory_target_live (session ref session_ref, VM ref self, int
target)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`target`	The target in bytes

Minimum Role: vm-power-admin

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_name_description (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VM.

Signature:

```
void set_name_label (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setNVRAM

Overview:

Signature:

```
void set_NVRAM (session ref session_ref, VM ref self, (string -> string)
map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`(string -> string) map`	`value`	The value

Minimum Role: vm-admin

Return Type: `void`

RPC name: setorder

Overview:

Set this VM's boot order

Signature:

```
void set_order (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	This VM's boot order

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: pool-operator

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VM.

Signature:

```
void set_other_config (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPCIbus

This message is deprecated.

Overview:

Set the PCIbus field of the given VM.

Signature:

```
void set_PCI_bus (session ref session_ref, VM ref self, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setplatform

Overview:

Set the platform field of the given VM.

Signature:

```
void set_platform (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setprotectionpolicy

This message is removed.

Overview:

Set the value of the protectionpolicy field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_protection_policy (session ref session_ref, VM ref self, VMPP ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`VMPP ref`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: setPVargs

Overview:

Set the PV/args field of the given VM.

Signature:

```
void set_PV_args (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setPVbootloader

Overview:

Set the PV/bootloader field of the given VM.

Signature:

```
void set_PV_bootloader (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVbootloaderargs

Overview:

Set the PV/bootloaderargs field of the given VM.

Signature:

```
void set_PV_bootloader_args (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVkernel

Overview:

Set the PV/kernel field of the given VM.

Signature:

```
void set_PV_kernel (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVlegacyargs

Overview:

Set the PV/legacyargs field of the given VM.

Signature:

```
void set_PV_legacy_args (session ref session_ref, VM ref self, string
value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setPVramdisk

Overview:

Set the PV/ramdisk field of the given VM.

Signature:

```
void set_PV_ramdisk (session ref session_ref, VM ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setrecommendations

Overview:

Set the recommendations field of the given VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_recommendations (session ref session_ref, VM ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setshadowmultiplierlive

Overview:

Set the shadow memory multiplier on a running VM

Signature:

```
void set_shadow_multiplier_live (session ref session_ref, VM ref self,
float multiplier)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`float`	`multiplier`	The new shadow memory multiplier to set

Minimum Role: vm-power-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setshutdowndelay

Overview:

Set this VM's shutdown delay in seconds

Signature:

```
void set_shutdown_delay (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	This VM's shutdown delay in seconds

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsnapshotschedule

Overview:

Set the value of the snapshot schedule field

Signature:

```
void set_snapshot_schedule (session ref session_ref, VM ref self, VMSS ref
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	The VM
`VMSS ref`	`value`	The value

Minimum Role: pool-operator

Return Type: `void`

RPC name: setstartdelay

Overview:

Set this VM's start delay in seconds

Signature:

```
void set_start_delay (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	This VM's start delay in seconds

Minimum Role: pool-operator

Return Type: `void`

RPC name: setsuspendSR

Overview:

Set the suspendSR field of the given VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_suspend_SR (session ref session_ref, VM ref self, SR ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`SR ref`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setsuspendVDI

Overview:

Set this VM's suspend VDI, which must be indentical to its current one

Signature:

```
void set_suspend_VDI (session ref session_ref, VM ref self, VDI ref value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`VDI ref`	`value`	The suspend VDI uuid

Minimum Role: pool-operator

Return Type: `void`

RPC name: settags

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Set the tags field of the given VM.

Signature:

```
void set_tags (session ref session_ref, VM ref self, string set value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`string set`	`value`	New value to set

Minimum Role: vm-operator

Return Type: `void`

RPC name: setuefimode

Overview:

Set the UEFI mode of a VM

Signature:

```
string set_uefi_mode (session ref session_ref, VM ref self, vm_uefi_mode
mode)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`vm_uefi_mode`	`mode`	The UEFI mode to set

Minimum Role: pool-admin

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

Result from the varstoresbstate call

RPC name: setuserversion

Overview:

Set the userversion field of the given VM.

Signature:

```
void set_user_version (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`int`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setVCPUsatstartup

Overview:

Set the number of startup VCPUs for a halted VM

Signature:

```
void set_VCPUs_at_startup (session ref session_ref, VM ref self, int
value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	The new maximum number of VCPUs

Minimum Role: vm-admin

Return Type: `void`

RPC name: setVCPUsmax

Overview:

Set the maximum number of VCPUs for a halted VM

Signature:

```
void set_VCPUs_max (session ref session_ref, VM ref self, int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`value`	The new maximum number of VCPUs

Minimum Role: vm-admin

Return Type: `void`

RPC name: setVCPUsnumberlive

Overview:

Set the number of VCPUs for a running VM

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_VCPUs_number_live (session ref session_ref, VM ref self, int
nvcpu)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`int`	`nvcpu`	The number of VCPUs

Minimum Role: vm-admin

Return Type: `void`

Possible Error Codes: `OPERATION_NOT_ALLOWED`, `LICENCE_RESTRICTION`

RPC name: setVCPUsparams

Overview:

Set the VCPUs/params field of the given VM.

Signature:

```
void set_VCPUs_params (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: setxenstoredata

Overview:

Set the xenstoredata field of the given VM.

Signature:

```
void set_xenstore_data (session ref session_ref, VM ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: shutdown

Overview:

Attempts to first clean shutdown a VM and if it should fail then perform a hard shutdown on it.

Signature:

```
void shutdown (session ref session_ref, VM ref vm)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to shutdown

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: snapshot

Overview:

Snapshots the specified VM, making a new VM. Snapshot automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write.

Signature:

```
VM ref snapshot (session ref session_ref, VM ref vm, string new_name, VDI ref set ignore_vdis)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be snapshotted
`string`	`new_name`	The name of the snapshotted VM
`VDI ref set`	`ignore_vdis`	A list of VDIs to ignore for the snapshot

Minimum Role: vm-power-admin

Return Type: `VM ref`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`

RPC name: snapshotwithquiesce

This message is removed.

Overview:

Snapshots the specified VM with quiesce, making a new VM. Snapshot automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored for example, Copy on Write.

Signature:

```
VM ref snapshot_with_quiesce (session ref session_ref, VM ref vm, string new_name)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to be snapshotted
`string`	`new_name`	The name of the snapshotted VM

Minimum Role: vm-power-admin

Return Type: `VM ref`

The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `VM_SNAPSHOT_WITH_QUIESCE_FAILED`, `VM_SNAPSHOT_WITH_QUIESCE_TIMEOUT`, `VM_SNAPSHOT_WITH_QUIESCE_PLUGIN_DEOS_NOT_RESPOND`, `VM_SNAPSHOT_WITH_QUIESCE_NOT_SUPPORTED`

RPC name: start

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Start the specified VM. This function can only be called with the VM is in the Halted State.

Signature:

```
void start (session ref session_ref, VM ref vm, bool start_paused, bool
force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to start
`bool`	`start_paused`	Instantiate VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to start. If this flag is false then the VM may fail pre-boot safety checks (for example, if the CPU the VM last booted on looks substantially different to the current one)

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `VM_HVM_REQUIRED`, `VM_IS_TEMPLATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `BOOTLOADER_FAILED`, `UNKNOWN_BOOTLOADER`, `NO_HOSTS_AVAILABLE`, `LICENCE_RESTRICTION`

RPC name: starton

Overview:

Start the specified VM on a particular host. This function can only be called with the VM is in the Halted State.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void start_on (session ref session_ref, VM ref vm, host ref host, bool
start_paused, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to start
`host ref`	`host`	The Host on which to start the VM
`bool`	`start_paused`	Instantiate VM in paused state if set to true.
`bool`	`force`	Attempt to force the VM to start. If this flag is false then the VM may fail pre-boot safety checks (for example, if the CPU the VM last booted on looks substantially different to the current one)

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `VM_IS_TEMPLATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `BOOTLOADER_FAILED`, `UNKNOWN_BOOTLOADER`

RPC name: suspend

Overview:

Suspend the specified VM to disk. This can only be called when the specified VM is in the Running state.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void suspend (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to suspend

Minimum Role: client-cert

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: sysprep

Overview:

Pass unattend.xml to Windows sysprep and wait for the VM to shut down as part of a reboot.

Signature:

```
void sysprep (session ref session_ref, VM ref self, string unattend, float
timeout)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`string`	`unattend`	XML content passed to sysprep
`float`	`timeout`	timeout in seconds for expected reboot

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: unpause

Overview:

Resume the specified VM. This can only be called when the specified VM is in the Paused state.

Signature:

```
void unpause (session ref session_ref, VM ref vm)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vm`	The VM to unpause

Minimum Role: vm-operator

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: updateallowedoperations

Overview:

Recomputes the list of acceptable operations

Signature:

```
void update_allowed_operations (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM ref`	`self`	reference to the object

Minimum Role: pool-admin

Return Type: `void`

RPC name: updatesecurebootcertificatesonboot

Overview:

Mark or unmark secure boot certificate update on VM boot

Signature:

```
void update_secureboot_certificates_on_boot (session ref session_ref, VM ref self, bool mark)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM
`bool`	`mark`	If true: mark certificates for update on next boot. If false: remove the mark

Minimum Role: vm-admin

Return Type: `void`

RPC name: waitmemorytargetlive

This message is deprecated.

Overview:

Wait for a running VM to reach its current memory target

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void wait_memory_target_live (session ref session_ref, VM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`self`	The VM

Minimum Role: read-only

Return Type: `void`

Class: VMappliance

VM appliance

Fields for class: VMappliance

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vm_appliance_operation set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`current_operations`	`(string -> vm_appliance_operation) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VMs`	`VM ref set`	*RO/runtime*	all VMs in this appliance

RPCs associated with class: VMappliance

RPC name: assertcanberecovered

Overview:

Assert whether all SRs required to recover this VM appliance are available.

Signature:

```
void assert_can_be_recovered (session ref session_ref, VM_appliance ref
self, session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance to recover
`session ref`	`session_to`	The session to which the VM appliance is to be recovered.

Minimum Role: read-only

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Possible Error Codes: `VM_REQUIRES_SR`

RPC name: cleanshutdown

Overview:

Perform a clean shutdown of all the VMs in the appliance

Signature:

```
void clean_shutdown (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

RPC name: create

Overview:

Create a new VMappliance instance, and return its handle.

Signature:

```
VM_appliance ref create (session ref session_ref, VM_appliance record
args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_appliance record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `VM_appliance ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VMappliance instance.

Signature:

```
void destroy (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMappliances known to the system.

Signature:

```
VM_appliance ref set get_all (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_appliance ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMappliance references to VMappliance records for all VMappliances known to the system.

Signature:

```
(VM_appliance ref -> VM_appliance record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM_appliance ref -> VM_appliance record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VMappliance.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
vm_appliance_operation set get_allowed_operations (session ref
session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vm_appliance_operation set`

value of the field

RPC name: getbynamelabel

Overview:

Get all the VMappliance instances with the given label.

Signature:

```
VM_appliance ref set get_by_name_label (session ref session_ref, string
label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VM_appliance ref set`

references to objects with matching names

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getbyuuid

Overview:

Get a reference to the VMappliance instance with the specified UUID.

Signature:

```
VM_appliance ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_appliance ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VMappliance.

Signature:

```
(string -> vm_appliance_operation) map get_current_operations (session ref
session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> vm_appliance_operation) map`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given VMappliance.

Signature:

```
string get_name_description (session ref session_ref, VM_appliance ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VMappliance.

Signature:

```
string get_name_label (session ref session_ref, VM_appliance ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMappliance.

Signature:

```
VM_appliance record get_record (session ref session_ref, VM_appliance ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_appliance record`

all fields from the object

RPC name: getSRsrequiredforrecovery

Overview:

Get the list of SRs required by the VM appliance to recover.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR ref set get_SRs_required_for_recovery (session ref session_ref,
VM_appliance ref self, session ref session_to)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance for which the required list of SRs has to be recovered.
`session ref`	`session_to`	The session to which the list of SRs have to be recovered .

Minimum Role: read-only

Return Type: `SR ref set`

refs for SRs required to recover the VM

RPC name: getuuid

Overview:

Get the uuid field of the given VMappliance.

Signature:

```
string get_uuid (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getVMs

Overview:

Get the VMs field of the given VMappliance.

Signature:

```
VM ref set get_VMs (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: hardshutdown

Overview:

Perform a hard shutdown of all the VMs in the appliance

Signature:

```
void hard_shutdown (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_appliance ref`	`self`	The VM appliance

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

RPC name: recover

Overview:

Recover the VM appliance

Signature:

```
void recover (session ref session_ref, VM_appliance ref self, session ref session_to, bool force)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance to recover
`session ref`	`session_to`	The session to which the VM appliance is to be recovered.
`bool`	`force`	Whether the VMs should replace newer versions of themselves.

Minimum Role: read-only

Return Type: `void`

Possible Error Codes: `VM_REQUIRES_SR`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setnamedescription

Overview:

Set the name/description field of the given VMappliance.

Signature:

```
void set_name_description (session ref session_ref, VM_appliance ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VMappliance.

Signature:

```
void set_name_label (session ref session_ref, VM_appliance ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: shutdown

Overview:

For each VM in the appliance, try to shut it down cleanly. If this fails, perform a hard shutdown of the VM.

Signature:

```
void shutdown (session ref session_ref, VM_appliance ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

RPC name: start

Overview:

Start all VMs in the appliance

Signature:

```
void start (session ref session_ref, VM_appliance ref self, bool paused)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_appliance ref`	`self`	The VM appliance
`bool`	`paused`	Instantiate all VMs belonging to this appliance in paused state if set to true.

Minimum Role: pool-operator

Return Type: `void`

Possible Error Codes: `OPERATION_PARTIALLY_FAILED`

Class: VMgroup

A VM group

Fields for class: VMgroup

FIELD	TYPE	QUALIFIER	DESCRIPTION
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`placement`	`placement_policy`	*RO/constructor*	The placement policy of the VM group
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VMs`	`VM ref set`	*RO/runtime*	The list of VMs associated with the group

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VMgroup

RPC name: create

Overview:

Create a new VMgroup instance, and return its handle.

Signature:

```
VM_group ref create (session ref session_ref, VM_group record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group record`	`args`	All constructor arguments

Minimum Role: vm-admin

Return Type: `VM_group ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VMgroup instance.

Signature:

```
void destroy (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMgroups known to the system.

Signature:

```
VM_group ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_group ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMgroup references to VMgroup records for all VMgroups known to the system.

Signature:

```
(VM_group ref -> VM_group record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `(VM_group ref -> VM_group record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the VMgroup instances with the given label.

Signature:

```
VM_group ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VM_group ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VMgroup instance with the specified UUID.

Signature:

```
VM_group ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_group ref`

reference to the object

RPC name: getnamedescription

Overview:

Get the name/description field of the given VMgroup.

Signature:

```
string get_name_description (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

Overview:

Get the name/label field of the given VMgroup.

Signature:

```
string get_name_label (session ref session_ref, VM_group ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getplacement

Overview:

Get the placement field of the given VMgroup.

Signature:

```
placement_policy get_placement (session ref session_ref, VM_group ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `placement_policy`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMgroup.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM_group record get_record (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_group record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given VMgroup.

Signature:

```
string get_uuid (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getVMs

Overview:

Get the VMs field of the given VMgroup.

Signature:

```
VM ref set get_VMs (session ref session_ref, VM_group ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: setnamedescription

Overview:

Set the name/description field of the given VMgroup.

Signature:

```
void set_name_description (session ref session_ref, VM_group ref self,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VMgroup.

Signature:

```
void set_name_label (session ref session_ref, VM_group ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_group ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VMguestmetrics

The metrics reported by the guest as opposed to inferred from outside

Fields for class: VMguestmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`can_use_hotplug_vbd`	`tristate_type`	*RO/runtime*	The guest's statement of whether

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			it supports VBD hotplug, i.e. whether it is capable of responding immediately to instantiation of a new VBD by bringing online a new PV block device. If the guest states that it is not capable, then the VBD plug and unplug operations will not be allowed while the guest is running.
`can_use_hotplug_vif`	`tristate_type`	*RO/runtime*	The guest's statement of whether it supports VIF hotplug, i.e. whether it is capable of responding immediately to instantiation of a new VIF by bringing online a new PV network device. If the guest states that it is not capable, then the VIF plug and unplug operations will not be allowed while the guest is running.
`disks`	`(string -> string) map`	*RO/runtime*	**Removed.** This field exists but has no data.
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`live`	`bool`	*RO/runtime*	True if the guest is sending heartbeat messages via the guest agent
`memory`	`(string -> string) map`	*RO/runtime*	**Removed.** This field exists but has no data. Use the memory and memory_internal_free RRD data-sources instead.
`netbios_name`	`(string -> string) map`	*RO/runtime*	The NETBIOS name of the machine
`networks`	`(string -> string) map`	*RO/runtime*	network configuration
`os_version`	`(string -> string) map`	*RO/runtime*	version of the OS
`other`	`(string -> string) map`	*RO/runtime*	anything else
`other_config`	`(string -> string) map`	*RW*	additional configuration
`PV_drivers_detected`	`bool`	*RO/runtime*	At least one of the guest's devices has successfully connected to the backend.
`PV_drivers_up_to_date`	`bool`	*RO/runtime*	**Deprecated.** Logically equivalent to PV_drivers_detected
`PV_drivers_version`	`(string -> string) map`	*RO/runtime*	version of the PV drivers

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`services`	`(string -> string) map`	*RO/runtime*	The guest's services data.
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference

RPCs associated with class: VMguestmetrics

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VMguestmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VM_guest_metrics ref self, string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMguestmetrics instances known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_guest_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_guest_metrics ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMguestmetrics references to VMguestmetrics records for all VMguestmetrics instances known to the system.

Signature:

```
(VM_guest_metrics ref -> VM_guest_metrics record) map get_all_records
(session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM_guest_metrics ref -> VM_guest_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VMguestmetrics instance with the specified UUID.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM_guest_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_guest_metrics ref`

reference to the object

RPC name: getcanusehotplugvbd

Overview:

Get the canusehotplugvbd field of the given VMguestmetrics.

Signature:

```
tristate_type get_can_use_hotplug_vbd (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tristate_type`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getcanusehotplugvif

Overview:

Get the canusehotplugvif field of the given VMguestmetrics.

Signature:

```
tristate_type get_can_use_hotplug_vif (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `tristate_type`

value of the field

RPC name: getdisks

This message is removed.

Overview:

Get the disks field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_disks (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given VMguestmetrics.

Signature:

```
datetime get_last_updated (session ref session_ref, VM_guest_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getlive

Overview:

Get the live field of the given VMguestmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_live (session ref session_ref, VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getmemory

This message is removed.

Overview:

Get the memory field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_memory (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnetbiosname

Overview:

Get the netbiosname field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_netbios_name (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getnetworks

Overview:

Get the networks field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_networks (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getosversion

Overview:

Get the osversion field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_os_version (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getother

Overview:

Get the other field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_other (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getPVdriversdetected

Overview:

Get the PVdriversdetected field of the given VMguestmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_PV_drivers_detected (session ref session_ref, VM_guest_metrics
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getPVdriversuptodate

This message is deprecated.

Overview:

Get the PVdriversuptodate field of the given VMguestmetrics.

Signature:

```
bool get_PV_drivers_up_to_date (session ref session_ref, VM_guest_metrics
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getPVdriversversion

Overview:

Get the PVdriversversion field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_PV_drivers_version (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMguestmetrics.

Signature:

```
VM_guest_metrics record get_record (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VM_guest_metrics record`

all fields from the object

RPC name: getservices

Overview:

Get the services field of the given VMguestmetrics.

Signature:

```
(string -> string) map get_services (session ref session_ref,
VM_guest_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VMguestmetrics.

Signature:

```
string get_uuid (session ref session_ref, VM_guest_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VMguestmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VM_guest_metrics ref self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VMguestmetrics.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_other_config (session ref session_ref, VM_guest_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_guest_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VMmetrics

The metrics associated with a VM

Fields for class: VMmetrics

FIELD	TYPE	QUALIFIER	DESCRIPTION
`current_domain_type`	`domain_type`	*RO/runtime*	The current domain type of the VM (for running,suspended, or paused VMs). The last-known domain type for halted VMs.
`hvm`	`bool`	*RO/runtime*	hardware virtual machine
`install_time`	`datetime`	*RO/runtime*	Time at which the VM was installed

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`last_updated`	`datetime`	*RO/runtime*	Time at which this information was last updated
`memory_actual`	`int`	*RO/runtime*	Guest's actual memory (bytes)
`nested_virt`	`bool`	*RO/runtime*	VM supports nested virtualisation
`nomigrate`	`bool`	*RO/runtime*	VM is immobile and can't migrate between hosts
`other_config`	`(string -> string) map`	*RW*	additional configuration
`start_time`	`datetime`	*RO/runtime*	Time at which this VM was last booted
`state`	`string set`	*RO/runtime*	The state of the guest, eg blocked, dying etc
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VCPUs_CPU`	`(int -> int) map`	*RO/runtime*	VCPU to PCPU map
`VCPUs_flags`	`(int -> string set) map`	*RO/runtime*	CPU flags (blocked,online,running)
`VCPUs_number`	`int`	*RO/runtime*	Current number of VCPUs
`VCPUs_params`	`(string -> string) map`	*RO/runtime*	The live equivalent to VM.VCPUs_params

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`VCPUs_utilisation`	`(int -> float) map`	*RO/runtime*	**Removed.** Utilisation for all of guest's current VCPUs

RPCs associated with class: VMmetrics

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VMmetrics.

Signature:

```
void add_to_other_config (session ref session_ref, VM_metrics ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: vm-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMmetrics instances known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_metrics ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VM_metrics ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMmetrics references to VMmetrics records for all VMmetrics instances known to the system.

Signature:

```
(VM_metrics ref -> VM_metrics record) map get_all_records (session ref
session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VM_metrics ref -> VM_metrics record) map`

records of all objects

RPC name: getbyuuid

Overview:

Get a reference to the VMmetrics instance with the specified UUID.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM_metrics ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VM_metrics ref`

reference to the object

RPC name: getcurrentdomaintype

Overview:

Get the currentdomaintype field of the given VMmetrics.

Signature:

```
domain_type get_current_domain_type (session ref session_ref, VM_metrics
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `domain_type`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: gethvm

Overview:

Get the hvm field of the given VMmetrics.

Signature:

```
bool get_hvm (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getinstalltime

Overview:

Get the installtime field of the given VMmetrics.

Signature:

```
datetime get_install_time (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `datetime`

value of the field

RPC name: getlastupdated

Overview:

Get the lastupdated field of the given VMmetrics.

Signature:

```
datetime get_last_updated (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getmemoryactual

Overview:

Get the memory/actual field of the given VMmetrics.

Signature:

```
int get_memory_actual (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getnestedvirt

Overview:

Get the nestedvirt field of the given VMmetrics.

Signature:

```
bool get_nested_virt (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getnomigrate

Overview:

Get the nomigrate field of the given VMmetrics.

Signature:

```
bool get_nomigrate (session ref session_ref, VM_metrics ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VMmetrics.

Signature:

```
(string -> string) map get_other_config (session ref session_ref,
VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMmetrics.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
VM_metrics record get_record (session ref session_ref, VM_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM_metrics record`

all fields from the object

RPC name: getstarttime

Overview:

Get the starttime field of the given VMmetrics.

Signature:

```
datetime get_start_time (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getstate

Overview:

Get the state field of the given VMmetrics.

Signature:

```
string set get_state (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VMmetrics.

Signature:

```
string get_uuid (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getVCPUsCPU

Overview:

Get the VCPUs/CPU field of the given VMmetrics.

Signature:

```
(int -> int) map get_VCPUs_CPU (session ref session_ref, VM_metrics ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(int -> int) map`

value of the field

RPC name: getVCPUsflags

Overview:

Get the VCPUs/flags field of the given VMmetrics.

Signature:

```
(int -> string set) map get_VCPUs_flags (session ref session_ref,
VM_metrics ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(int -> string set) map`

value of the field

RPC name: getVCPUsnumber

Overview:

Get the VCPUs/number field of the given VMmetrics.

Signature:

```
int get_VCPUs_number (session ref session_ref, VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getVCPUsparams

Overview:

Get the VCPUs/params field of the given VMmetrics.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_VCPUs_params (session ref session_ref,
VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getVCPUsutilisation

This message is removed.

Overview:

Get the VCPUs/utilisation field of the given VMmetrics.

Signature:

```
(int -> float) map get_VCPUs_utilisation (session ref session_ref,
VM_metrics ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(int -> float) map`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VMmetrics. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VM_metrics ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM_metrics ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: vm-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VMmetrics.

Signature:

```
void set_other_config (session ref session_ref, VM_metrics ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VM_metrics ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: vm-admin

Return Type: `void`

Class: VMPP

This class is removed.

VM Protection Policy

Fields for class: VMPP

FIELD	TYPE	QUALIFIER	DESCRIPTION
`alarm_config`	`(string -> string) map`	*RO/constructor*	**Removed.** configuration for the alarm
`archive_frequency`	`vmpp_archive_frequency`	*RO/constructor*	**Removed.** frequency of the archive schedule
`archive_last_run_time`	`datetime`	*RO/runtime*	**Removed.** time of the last archive
`archive_schedule`	`(string -> string) map`	*RO/constructor*	**Removed.** schedule of the archive containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone
`archive_target_config`	`(string -> string) map`	*RO/constructor*	**Removed.** configuration for the archive, including its 'location',

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
			'username', 'password'
`archive_target_type`	`vmpp_archive_target_type`	*RO/constructor*	**Removed.** type of the archive target config
`backup_frequency`	`vmpp_backup_frequency`	*RO/constructor*	**Removed.** frequency of the backup schedule
`backup_last_run_time`	`datetime`	*RO/runtime*	**Removed.** time of the last backup
`backup_retention_value`	`int`	*RO/constructor*	**Removed.** maximum number of backups that should be stored at any time
`backup_schedule`	`(string -> string) map`	*RO/constructor*	**Removed.** schedule of the backup containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone
`backup_type`	`vmpp_backup_type`	*RW*	**Removed.** type of the backup sub-policy
`is_alarm_enabled`	`bool`	*RO/constructor*	**Removed.** true if alarm is enabled for this policy
`is_archive_running`	`bool`	*RO/runtime*	**Removed.** true if this protection policy's archive is running

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`is_backup_running`	`bool`	*RO/runtime*	**Removed.** true if this protection policy's backup is running
`is_policy_enabled`	`bool`	*RW*	**Removed.** enable or disable this policy
`name_description`	`string`	*RW*	**Removed.** a notes field containing human-readable description
`name_label`	`string`	*RW*	**Removed.** a human-readable name
`recent_alerts`	`string set`	*RO/runtime*	**Removed.** recent alerts
`uuid`	`string`	*RO/runtime*	**Removed.** Unique identifier/object reference
`VMs`	`VM ref set`	*RO/runtime*	**Removed.** all VMs attached to this protection policy

RPCs associated with class: VMPP

RPC name: addtoalarmconfig

This message is removed.

Overview:

Signature:

```
void add_to_alarm_config (session ref session_ref, VMPP ref self, string key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtoarchiveschedule

This message is removed.

Overview:

Signature:

```
void add_to_archive_schedule (session ref session_ref, VMPP ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtoarchivetargetconfig

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Signature:

```
void add_to_archive_target_config (session ref session_ref, VMPP ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: addtobackupschedule

This message is removed.

Overview:

Signature:

```
void add_to_backup_schedule (session ref session_ref, VMPP ref self,
string key, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to add

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: archivenow

This message is removed.

Overview:

This call archives the snapshot provided as a parameter

Signature:

```
string archive_now (session ref session_ref, VM ref snapshot)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`snapshot`	The snapshot to archive

Minimum Role: vm-power-admin

Return Type: `string`

An XMLRPC result

RPC name: create

This message is removed.

Overview:

Create a new VMPP instance, and return its handle.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VMPP ref create (session ref session_ref, VMPP record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `VMPP ref`

reference to the newly created object

RPC name: destroy

This message is removed.

Overview:

Destroy the specified VMPP instance.

Signature:

```
void destroy (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getalarmconfig

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Get the alarmconfig field of the given VMPP.

Signature:

```
(string -> string) map get_alarm_config (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getalerts

This message is removed.

Overview:

This call fetches a history of alerts for a given protection policy

Signature:

```
string set get_alerts (session ref session_ref, VMPP ref vmpp, int hours_from_now)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`vmpp`	The protection policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`int`	`hours_from_now`	how many hours in the past the oldest record to fetch is

Minimum Role: pool-operator

Return Type: `string set`

A list of alerts encoded in xml

RPC name: getall

This message is removed.

Overview:

Return a list of all the VMPPs known to the system.

Signature:

```
VMPP ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VMPP ref set`

references to all objects

RPC name: getallrecords

This message is removed.

Overview:

Return a map of VMPP references to VMPP records for all VMPPs known to the system.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(VMPP ref -> VMPP record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VMPP ref -> VMPP record) map`

records of all objects

RPC name: getarchivefrequency

This message is removed.

Overview:

Get the archivefrequency field of the given VMPP.

Signature:

```
vmpp_archive_frequency get_archive_frequency (session ref session_ref,
VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmpp_archive_frequency`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getarchivelastruntime

This message is removed.

Overview:

Get the archivelastruntime field of the given VMPP.

Signature:

```
datetime get_archive_last_run_time (session ref session_ref, VMPP ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getarchiveschedule

This message is removed.

Overview:

Get the archiveschedule field of the given VMPP.

Signature:

```
(string -> string) map get_archive_schedule (session ref session_ref, VMPP
ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getarchivetargetconfig

This message is removed.

Overview:

Get the archivetargetconfig field of the given VMPP.

Signature:

```
(string -> string) map get_archive_target_config (session ref session_ref,
VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getarchivetargettype

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the archivetargettype field of the given VMPP.

Signature:

```
vmpp_archive_target_type get_archive_target_type (session ref session_ref,
VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmpp_archive_target_type`

value of the field

RPC name: getbackupfrequency

This message is removed.

Overview:

Get the backupfrequency field of the given VMPP.

Signature:

```
vmpp_backup_frequency get_backup_frequency (session ref session_ref, VMPP
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `vmpp_backup_frequency`

value of the field

RPC name: getbackuplastruntime

This message is removed.

Overview:

Get the backuplastruntime field of the given VMPP.

Signature:

```
datetime get_backup_last_run_time (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getbackupretentionvalue

This message is removed.

Overview:

Get the backupretentionvalue field of the given VMPP.

Signature:

```
int get_backup_retention_value (session ref session_ref, VMPP ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getbackupschedule

This message is removed.

Overview:

Get the backupschedule field of the given VMPP.

Signature:

```
(string -> string) map get_backup_schedule (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getbackuptype

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the backuptype field of the given VMPP.

Signature:

```
vmpp_backup_type get_backup_type (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmpp_backup_type`

value of the field

RPC name: getbynamelabel

This message is removed.

Overview:

Get all the VMPP instances with the given label.

Signature:

```
VMPP ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VMPP ref set`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

references to objects with matching names

RPC name: getbyuuid

This message is removed.

Overview:

Get a reference to the VMPP instance with the specified UUID.

Signature:

```
VMPP ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VMPP ref`

reference to the object

RPC name: getisalarmenabled

This message is removed.

Overview:

Get the isalarmenabled field of the given VMPP.

Signature:

```
bool get_is_alarm_enabled (session ref session_ref, VMPP ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisarchiverunning

This message is removed.

Overview:

Get the isarchiverunning field of the given VMPP.

Signature:

```
bool get_is_archive_running (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisbackuprunning

This message is removed.

Overview:

Get the isbackuprunning field of the given VMPP.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
bool get_is_backup_running (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getispolicyenabled

This message is removed.

Overview:

Get the ispolicyenabled field of the given VMPP.

Signature:

```
bool get_is_policy_enabled (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnamedescription

This message is removed.

Overview:

Get the name/description field of the given VMPP.

Signature:

```
string get_name_description (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getnamelabel

This message is removed.

Overview:

Get the name/label field of the given VMPP.

Signature:

```
string get_name_label (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecentalerts

This message is removed.

Overview:

Get the recentalerts field of the given VMPP.

Signature:

```
string set get_recent_alerts (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string set`

value of the field

RPC name: getrecord

This message is removed.

Overview:

Get a record containing the current state of the given VMPP.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VMPP record get_record (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VMPP record`

all fields from the object

RPC name: getuuid

This message is removed.

Overview:

Get the uuid field of the given VMPP.

Signature:

```
string get_uuid (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getVMs

This message is removed.

Overview:

Get the VMs field of the given VMPP.

Signature:

```
VM ref set get_VMs (session ref session_ref, VMPP ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: protectnow

This message is removed.

Overview:

This call executes the protection policy immediately

Signature:

```
string protect_now (session ref session_ref, VMPP ref vmpp)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMPP ref`	`vmpp`	The protection policy to execute

Minimum Role: pool-operator

Return Type: `string`

An XMLRPC result

RPC name: removefromalarmconfig

This message is removed.

Overview:

Signature:

```
void remove_from_alarm_config (session ref session_ref, VMPP ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromarchiveschedule

This message is removed.

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_archive_schedule (session ref session_ref, VMPP ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: removefromarchivetargetconfig

This message is removed.

Overview:

Signature:

```
void remove_from_archive_target_config (session ref session_ref, VMPP ref
self, string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: removefrombackupschedule

This message is removed.

Overview:

Signature:

```
void remove_from_backup_schedule (session ref session_ref, VMPP ref self,
string key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setalarmconfig

This message is removed.

Overview:

Signature:

```
void set_alarm_config (session ref session_ref, VMPP ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setarchivefrequency

This message is removed.

Overview:

Set the value of the archivefrequency field

Signature:

```
void set_archive_frequency (session ref session_ref, VMPP ref self,
vmpp_archive_frequency value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`vmpp_archive_frequency`	`value`	the archive frequency

Minimum Role: pool-operator

Return Type: `void`

RPC name: setarchivelastruntime

This message is removed.

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_archive_last_run_time (session ref session_ref, VMPP ref self,
datetime value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`datetime`	`value`	The time at which the last archive was created. When the timezone is missing, UTC is assumed

Return Type: `void`

RPC name: setarchiveschedule

This message is removed.

Overview:

Signature:

```
void set_archive_schedule (session ref session_ref, VMPP ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setarchivetargetconfig

This message is removed.

Overview:

Signature:

```
void set_archive_target_config (session ref session_ref, VMPP ref self,
(string -> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setarchivetargettype

This message is removed.

Overview:

Set the value of the archivetargetconfigtype field

Signature:

```
void set_archive_target_type (session ref session_ref, VMPP ref self,
vmpp_archive_target_type value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMPP ref`	`self`	The protection policy
`vmpp_archive_target_type`	`value`	the archive target config type

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbackupfrequency

This message is removed.

Overview:

Set the value of the backupfrequency field

Signature:

```
void set_backup_frequency (session ref session_ref, VMPP ref self,
vmpp_backup_frequency value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`vmpp_backup_frequency`	`value`	the backup frequency

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbackuplastruntime

This message is removed.

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
void set_backup_last_run_time (session ref session_ref, VMPP ref self,
datetime value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`datetime`	`value`	The time at which the last backup was done. When the timezone is missing, UTC is assumed

Return Type: `void`

RPC name: setbackupretentionvalue

This message is removed.

Overview:

Signature:

```
void set_backup_retention_value (session ref session_ref, VMPP ref self,
int value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`int`	`value`	the value to set

Minimum Role: pool-operator

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `void`

RPC name: setbackupschedule

This message is removed.

Overview:

Signature:

```
void set_backup_schedule (session ref session_ref, VMPP ref self, (string
-> string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setbackuptype

This message is removed.

Overview:

Set the backuptype field of the given VMPP.

Signature:

```
void set_backup_type (session ref session_ref, VMPP ref self,
vmpp_backup_type value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object
`vmpp_backup_type`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setisalarmenabled

This message is removed.

Overview:

Set the value of the isalarmenabled field

Signature:

```
void set_is_alarm_enabled (session ref session_ref, VMPP ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	The protection policy
`bool`	`value`	true if alarm is enabled for this policy

Minimum Role: pool-operator

Return Type: `void`

RPC name: setispolicyenabled

This message is removed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Overview:

Set the ispolicyenabled field of the given VMPP.

Signature:

```
void set_is_policy_enabled (session ref session_ref, VMPP ref self, bool
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamedescription

This message is removed.

Overview:

Set the name/description field of the given VMPP.

Signature:

```
void set_name_description (session ref session_ref, VMPP ref self, string
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

This message is removed.

Overview:

Set the name/label field of the given VMPP.

Signature:

```
void set_name_label (session ref session_ref, VMPP ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMPP ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

Class: VMSS

VM Snapshot Schedule

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Fields for class: VMSS

FIELD	TYPE	QUALIFIER	DESCRIPTION
`enabled`	`bool`	*RW*	enable or disable this snapshot schedule
`frequency`	`vmss_frequency`	*RO/constructor*	frequency of taking snapshot from snapshot schedule
`last_run_time`	`datetime`	*RO/runtime*	time of the last snapshot
`name_description`	`string`	*RW*	a notes field containing human-readable description
`name_label`	`string`	*RW*	a human-readable name
`retained_snapshots`	`int`	*RO/constructor*	maximum number of snapshots that should be stored at any time
`schedule`	`(string -> string) map`	*RO/constructor*	schedule of the snapshot containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone
`type`	`vmss_type`	*RO/constructor*	type of the snapshot schedule
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VMs`	`VM ref set`	*RO/runtime*	all VMs attached to this snapshot schedule

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPCs associated with class: VMSS

RPC name: addtoschedule

Overview:

Signature:

```
void add_to_schedule (session ref session_ref, VMSS ref self, string key,
string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`string`	`key`	the key to add
`string`	`value`	the value to add

Minimum Role: pool-operator

Return Type: `void`

RPC name: create

Overview:

Create a new VMSS instance, and return its handle.

Signature:

```
VMSS ref create (session ref session_ref, VMSS record args)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMSS record`	`args`	All constructor arguments

Minimum Role: pool-operator

Return Type: `VMSS ref`

reference to the newly created object

RPC name: destroy

Overview:

Destroy the specified VMSS instance.

Signature:

```
void destroy (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: pool-operator

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VMSSs known to the system.

Signature:

```
VMSS ref set get_all (session ref session_ref)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VMSS ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VMSS references to VMSS records for all VMSSs known to the system.

Signature:

```
(VMSS ref -> VMSS record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VMSS ref -> VMSS record) map`

records of all objects

RPC name: getbynamelabel

Overview:

Get all the VMSS instances with the given label.

Signature:

```
VMSS ref set get_by_name_label (session ref session_ref, string label)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`label`	label of object to return

Minimum Role: read-only

Return Type: `VMSS ref set`

references to objects with matching names

RPC name: getbyuuid

Overview:

Get a reference to the VMSS instance with the specified UUID.

Signature:

```
VMSS ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VMSS ref`

reference to the object

RPC name: getenabled

Overview:

Get the enabled field of the given VMSS.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_enabled (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getfrequency

Overview:

Get the frequency field of the given VMSS.

Signature:

```
vmss_frequency get_frequency (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmss_frequency`

value of the field

RPC name: getlastruntime

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Get the lastruntime field of the given VMSS.

Signature:

```
datetime get_last_run_time (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `datetime`

value of the field

RPC name: getnamedescription

Overview:

Get the name/description field of the given VMSS.

Signature:

```
string get_name_description (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getnamelabel

Overview:

Get the name/label field of the given VMSS.

Signature:

```
string get_name_label (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VMSS.

Signature:

```
VMSS record get_record (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `VMSS record`

all fields from the object

RPC name: getretainedsnapshots

Overview:

Get the retainedsnapshots field of the given VMSS.

Signature:

```
int get_retained_snapshots (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `int`

value of the field

RPC name: getschedule

Overview:

Get the schedule field of the given VMSS.

Signature:

```
(string -> string) map get_schedule (session ref session_ref, VMSS ref
self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: gettype

Overview:

Get the type field of the given VMSS.

Signature:

```
vmss_type get_type (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vmss_type`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VMSS.

Signature:

```
string get_uuid (session ref session_ref, VMSS ref self)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `string`

value of the field

RPC name: getVMs

Overview:

Get the VMs field of the given VMSS.

Signature:

```
VM ref set get_VMs (session ref session_ref, VMSS ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref set`

value of the field

RPC name: removefromschedule

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void remove_from_schedule (session ref session_ref, VMSS ref self, string
key)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`string`	`key`	the key to remove

Minimum Role: pool-operator

Return Type: `void`

RPC name: setenabled

Overview:

Set the enabled field of the given VMSS.

Signature:

```
void set_enabled (session ref session_ref, VMSS ref self, bool value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object
`bool`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: setfrequency

Overview:

Set the value of the frequency field

Signature:

```
void set_frequency (session ref session_ref, VMSS ref self, vmss_frequency value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`vmss_frequency`	`value`	the snapshot schedule frequency

Minimum Role: pool-operator

Return Type: `void`

RPC name: setlastruntime

Overview:

Signature:

```
void set_last_run_time (session ref session_ref, VMSS ref self, datetime value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`datetime`	`value`	The time at which the schedule was last run. When the timezone is missing, UTC is assumed

Return Type: `void`

RPC name: setnamedescription

Overview:

Set the name/description field of the given VMSS.

Signature:

```
void set_name_description (session ref session_ref, VMSS ref self, string value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setnamelabel

Overview:

Set the name/label field of the given VMSS.

Signature:

```
void set_name_label (session ref session_ref, VMSS ref self, string value)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	reference to the object
`string`	`value`	New value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setretainedsnapshots

Overview:

Signature:

```
void set_retained_snapshots (session ref session_ref, VMSS ref self, int
value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The schedule snapshot
`int`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: setschedule

Overview:

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void set_schedule (session ref session_ref, VMSS ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`(string -> string) map`	`value`	the value to set

Minimum Role: pool-operator

Return Type: `void`

RPC name: settype

Overview:

Signature:

```
void set_type (session ref session_ref, VMSS ref self, vmss_type value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`self`	The snapshot schedule
`vmss_type`	`value`	the snapshot schedule type

Minimum Role: pool-operator

Return Type: `void`

RPC name: snapshotnow

Overview:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

This call executes the snapshot schedule immediately

Signature:

```
string snapshot_now (session ref session_ref, VMSS ref vmss)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VMSS ref`	`vmss`	Snapshot Schedule to execute

Minimum Role: pool-operator

Return Type: `string`

An XMLRPC result

Class: VTPM

A virtual TPM device

Fields for class: VTPM

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vtpm_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.
`backend`	`VM ref`	*RO/runtime*	The domain where the backend is located (unused)

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`current_operations`	`(string -> vtpm_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`is_protected`	`bool`	*RO/runtime*	Whether the contents of the VTPM are secured according to the TPM spec
`is_unique`	`bool`	*RO/constructor*	Whether the contents are never copied, satisfying the TPM spec
`persistence_backend`	`persistence_backend`	*RO/runtime*	The backend where the vTPM is persisted
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VM`	`VM ref`	*RO/constructor*	The virtual machine the TPM is attached to

RPCs associated with class: VTPM

RPC name: create

Overview:

Create a new VTPM instance, and return its handle.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VTPM ref create (session ref session_ref, VM ref vM, bool is_unique)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`vM`	The VM reference the VTPM will be attached to
`bool`	`is_unique`	Whether the VTPM must be unique

Minimum Role: vm-admin

Return Type: `VTPM ref`

The reference of the newly created VTPM

RPC name: destroy

Overview:

Destroy the specified VTPM instance, along with its state.

Signature:

```
void destroy (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	The reference to the VTPM object

Minimum Role: vm-admin

Return Type: `void`

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getall

Overview:

Return a list of all the VTPMs known to the system.

Signature:

```
VTPM ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VTPM ref set`

references to all objects

RPC name: getallrecords

Overview:

Return a map of VTPM references to VTPM records for all VTPMs known to the system.

Signature:

```
(VTPM ref -> VTPM record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VTPM ref -> VTPM record) map`

records of all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VTPM.

Signature:

```
vtpm_operations set get_allowed_operations (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `vtpm_operations set`

value of the field

RPC name: getbackend

Overview:

Get the backend field of the given VTPM.

Signature:

```
VM ref get_backend (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VTPM instance with the specified UUID.

Signature:

```
VTPM ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VTPM ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VTPM.

Signature:

```
(string -> vtpm_operations) map get_current_operations (session ref
session_ref, VTPM ref self)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vtpm_operations) map`

value of the field

RPC name: getisprotected

Overview:

Get the isprotected field of the given VTPM.

Signature:

```
bool get_is_protected (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getisunique

Overview:

Get the isunique field of the given VTPM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bool get_is_unique (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getpersistencebackend

Overview:

Get the persistencebackend field of the given VTPM.

Signature:

```
persistence_backend get_persistence_backend (session ref session_ref, VTPM
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `persistence_backend`

value of the field

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getrecord

Overview:

Get a record containing the current state of the given VTPM.

Signature:

```
VTPM record get_record (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VTPM record`

all fields from the object

RPC name: getuuid

Overview:

Get the uuid field of the given VTPM.

Signature:

```
string get_uuid (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VTPM.

Signature:

```
VM ref get_VM (session ref session_ref, VTPM ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VTPM ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

Class: VUSB

Describes the vusb device

Fields for class: VUSB

FIELD	TYPE	QUALIFIER	DESCRIPTION
`allowed_operations`	`vusb_operations set`	*RO/runtime*	list of the operations allowed in this state. This list is advisory only and the server state may have changed by the time this field is read by a client.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FIELD	TYPE	QUALIFIER	DESCRIPTION
`current_operations`	`(string -> vusb_operations) map`	*RO/runtime*	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
`currently_attached`	`bool`	*RO/runtime*	is the device currently attached
`other_config`	`(string -> string) map`	*RW*	Additional configuration
`USB_group`	`USB_group ref`	*RO/runtime*	USB group used by the VUSB
`uuid`	`string`	*RO/runtime*	Unique identifier/ object reference
`VM`	`VM ref`	*RO/runtime*	VM that owns the VUSB

RPCs associated with class: VUSB

RPC name: addtootherconfig

Overview:

Add the given keyvalue pair to the otherconfig field of the given VUSB.

Signature:

```
void add_to_other_config (session ref session_ref, VUSB ref self, string key, string value)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object
`string`	`key`	Key to add
`string`	`value`	Value to add

Minimum Role: pool-admin

Return Type: `void`

RPC name: create

Overview:

Create a new VUSB record in the database only

Signature:

```
VUSB ref create (session ref session_ref, VM ref VM, USB_group ref
USB_group, (string -> string) map other_config)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VM ref`	`VM`	The VM
`USB_group ref`	`USB_group`	
`(string -> string) map`	`other_config`	

Minimum Role: pool-admin

Return Type: `VUSB ref`

The ref of the newly created VUSB record.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: destroy

Overview:

Removes a VUSB record from the database

Signature:

```
void destroy (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	The VUSB to destroy about

Minimum Role: pool-admin

Return Type: `void`

RPC name: getall

Overview:

Return a list of all the VUSBs known to the system.

Signature:

```
VUSB ref set get_all (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `VUSB ref set`

references to all objects

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getallrecords

Overview:

Return a map of VUSB references to VUSB records for all VUSBs known to the system.

Signature:

```
(VUSB ref -> VUSB record) map get_all_records (session ref session_ref)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

Minimum Role: read-only

Return Type: `(VUSB ref -> VUSB record) map`

records of all objects

RPC name: getallowedoperations

Overview:

Get the allowedoperations field of the given VUSB.

Signature:

```
vusb_operations set get_allowed_operations (session ref session_ref, VUSB
ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `vusb_operations set`

value of the field

RPC name: getbyuuid

Overview:

Get a reference to the VUSB instance with the specified UUID.

Signature:

```
VUSB ref get_by_uuid (session ref session_ref, string uuid)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`string`	`uuid`	UUID of object to return

Minimum Role: read-only

Return Type: `VUSB ref`

reference to the object

RPC name: getcurrentoperations

Overview:

Get the currentoperations field of the given VUSB.

Signature:

```
(string -> vusb_operations) map get_current_operations (session ref
session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> vusb_operations) map`

value of the field

RPC name: getcurrentlyattached

Overview:

Get the currentlyattached field of the given VUSB.

Signature:

```
bool get_currently_attached (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `bool`

value of the field

RPC name: getotherconfig

Overview:

Get the otherconfig field of the given VUSB.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
(string -> string) map get_other_config (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `(string -> string) map`

value of the field

RPC name: getrecord

Overview:

Get a record containing the current state of the given VUSB.

Signature:

```
VUSB record get_record (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VUSB record`

all fields from the object

© 1997–2026 Citrix Systems, Inc. All rights reserved.

RPC name: getUSBgroup

Overview:

Get the USBgroup field of the given VUSB.

Signature:

```
USB_group ref get_USB_group (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `USB_group ref`

value of the field

RPC name: getuuid

Overview:

Get the uuid field of the given VUSB.

Signature:

```
string get_uuid (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Return Type: `string`

value of the field

RPC name: getVM

Overview:

Get the VM field of the given VUSB.

Signature:

```
VM ref get_VM (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object

Minimum Role: read-only

Return Type: `VM ref`

value of the field

RPC name: removefromotherconfig

Overview:

Remove the given key and its corresponding value from the otherconfig field of the given VUSB. If the key is not in that Map, then do nothing.

Signature:

```
void remove_from_other_config (session ref session_ref, VUSB ref self,
string key)
```

Arguments:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object
`string`	`key`	Key to remove

Minimum Role: pool-admin

Return Type: `void`

RPC name: setotherconfig

Overview:

Set the otherconfig field of the given VUSB.

Signature:

```
void set_other_config (session ref session_ref, VUSB ref self, (string ->
string) map value)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	reference to the object
`(string -> string) map`	`value`	New value to set

Minimum Role: pool-admin

Return Type: `void`

RPC name: unplug

Overview:

Unplug the vusb device from the vm.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
void unplug (session ref session_ref, VUSB ref self)
```

Arguments:

TYPE	NAME	DESCRIPTION
`session ref`	`session_ref`	Reference to a valid session
`VUSB ref`	`self`	vusb deivce

Minimum Role: pool-admin

Return Type: `void`

Error Codes

The following is a list of all possible errors that can be issued by API calls.

ACTIVATIONWHILENOTFREE

An activation key can only be applied when the edition is set to 'free'.

Signature:

```
ACTIVATION_WHILE_NOT_FREE()
```

ADDRESSVIOLATESLOCKINGCONSTRAINT

The specified IP address violates the VIF locking configuration.

Signature:

```
ADDRESS_VIOLATES_LOCKING_CONSTRAINT(address)
```

APPLYGUIDANCEFAILED

Failed to apply guidance on a host after updating.

Signature:

```
APPLY_GUIDANCE_FAILED(ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

APPLYLIVEPATCHFAILED

Failed to apply a livepatch.

Signature:

```
APPLY_LIVEPATCH_FAILED(livepatch)
```

APPLYUPDATESFAILED

Failed to apply updates on a host.

Signature:

```
APPLY_UPDATES_FAILED(ref)
```

APPLYUPDATESINPROGRESS

The operation could not be performed because applying updates is in progress.

Signature:

```
APPLY_UPDATES_IN_PROGRESS()
```

AUTHALREADYENABLED

External authentication for this server is already enabled.

Signature:

```
AUTH_ALREADY_ENABLED(current auth_type, current service_name)
```

AUTHDISABLEFAILED

The host failed to disable external authentication.

Signature:

```
AUTH_DISABLE_FAILED(message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHDISABLEFAILEDPERMISSIONDENIED

The host failed to disable external authentication.

Signature:

```
AUTH_DISABLE_FAILED_PERMISSION_DENIED(message)
```

AUTHDISABLEFAILEDWRONGCREDENTIALS

The host failed to disable external authentication.

Signature:

```
AUTH_DISABLE_FAILED_WRONG_CREDENTIALS(message)
```

AUTHENABLEFAILED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED(message)
```

AUTHENABLEFAILEDDOMAINLOOKUPFAILED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED(message)
```

AUTHENABLEFAILEDINVALIDACCOUNT

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_INVALID_ACCOUNT(message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHENABLEFAILEDINVALIDOU

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_INVALID_OU(message)
```

AUTHENABLEFAILEDPERMISSIONDENIED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_PERMISSION_DENIED(message)
```

AUTHENABLEFAILEDUNAVAILABLE

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_UNAVAILABLE(message)
```

AUTHENABLEFAILEDWRONGCREDENTIALS

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_WRONG_CREDENTIALS(message)
```

AUTHISDISABLED

External authentication is disabled, unable to resolve subject name.

Signature:

```
AUTH_IS_DISABLED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

AUTHSERVICEERROR

Error querying the external directory service.

Signature:

```
AUTH_SERVICE_ERROR(message)
```

AUTHUNKNOWNTYPE

Unknown type of external authentication.

Signature:

```
AUTH_UNKNOWN_TYPE(type)
```

BACKUPSCRIPTFAILED

The backup could not be performed because the backup script failed.

Signature:

```
BACKUP_SCRIPT_FAILED(log)
```

BALLOONINGTIMEOUTBEFOREMIGRATION

Timeout trying to balloon down memory before VM migration. If the error occurs repeatedly, consider increasing the memorydynamicmin value.

Signature:

```
BALLOONING_TIMEOUT_BEFORE_MIGRATION(vm)
```

BLOCKEDREPOURL

Cannot create the repository as the url is blocked, please check your settings.

Signature:

```
BLOCKED_REPO_URL(url)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

BOOTLOADERFAILED

The bootloader returned an error

Signature:

```
BOOTLOADER_FAILED(vm, msg)
```

BOOTDEVICEPASSTHROUGHDISALLOWED

Passing through a PCI device backing a boot disk is disallowed

Signature:

```
BOOT_DEVICE_PASSTHROUGH_DISALLOWED(device)
```

BRIDGENAMEEXISTS

The specified bridge already exists.

Signature:

```
BRIDGE_NAME_EXISTS(bridge)
```

BRIDGENOTAVAILABLE

Could not find bridge required by VM.

Signature:

```
BRIDGE_NOT_AVAILABLE(bridge)
```

BUNDLEREPOSITORYALREADYEXISTS

The bundle repository already exists.

Signature:

```
BUNDLE_REPOSITORY_ALREADY_EXISTS(ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

BUNDLEREPONOTENABLED

Cannot sync bundle as the bundle repository is not enabled.

Signature:

```
BUNDLE_REPO_NOT_ENABLED()
```

BUNDLESYNCFAILED

Syncing with bundle repository failed.

Signature:

```
BUNDLE_SYNC_FAILED()
```

BUNDLEUNPACKFAILED

Failed to unpack bundle file.

Signature:

```
BUNDLE_UNPACK_FAILED(error)
```

CANNOTADDTUNNELTOBONDSLAVE

This PIF is a bond member and cannot have a tunnel on it.

Signature:

```
CANNOT_ADD_TUNNEL_TO_BOND_SLAVE(PIF)
```

CANNOTADDTUNNELTOSRIOVLOGICAL

This is a network SRIOV logical PIF and cannot have a tunnel on it.

Signature:

```
CANNOT_ADD_TUNNEL_TO_SRIOV_LOGICAL(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTADDTUNNELTOVLANONSRIOVLOGICAL

This is a vlan PIF on network SRIOV and cannot have a tunnel on it.

Signature:

```
CANNOT_ADD_TUNNEL_TO_VLAN_ON_SRIOV_LOGICAL(PIF)
```

CANNOTADDVLANTOBONDSLAVE

This PIF is a bond member and cannot have a VLAN on it.

Signature:

```
CANNOT_ADD_VLAN_TO_BOND_SLAVE(PIF)
```

CANNOTCHANGEPIFPROPERTIES

The properties of this PIF cannot be changed. Only the properties of nonbonded physical PIFs, or bond interfaces can be changed.

Signature:

```
CANNOT_CHANGE_PIF_PROPERTIES(PIF)
```

CANNOTCONTACTHOST

Cannot forward messages because the server cannot be contacted. The server may be switched off or there may be network connectivity problems.

Signature:

```
CANNOT_CONTACT_HOST(host)
```

CANNOTCREATESTATEFILE

An HA statefile could not be created, perhaps because no SR with the appropriate capability was found.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
CANNOT_CREATE_STATE_FILE()
```

CANNOTDESTROYDISASTERRECOVERYTASK

The disaster recovery task could not be cleanly destroyed.

Signature:

```
CANNOT_DESTROY_DISASTER_RECOVERY_TASK(reason)
```

CANNOTDESTROYSYSTEMNETWORK

You tried to destroy a system network: these cannot be destroyed.

Signature:

```
CANNOT_DESTROY_SYSTEM_NETWORK(network)
```

CANNOTENABLEREDOLOG

Could not enable redo log.

Signature:

```
CANNOT_ENABLE_REDO_LOG(reason)
```

CANNOTEVACUATEHOST

This server cannot be evacuated.

Signature:

```
CANNOT_EVACUATE_HOST(errors)
```

CANNOTFETCHPATCH

The requested update could not be obtained from the coordinator.

Signature:

```
CANNOT_FETCH_PATCH(uuid)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTFINDOEMBACKUPPARTITION

The backup partition to stream the update to cannot be found.

Signature:

```
CANNOT_FIND_OEM_BACKUP_PARTITION()
```

CANNOTFINDPATCH

The requested update could not be found. This can occur when you designate a new coordinator or xe patchclean. Please upload the update again.

Signature:

```
CANNOT_FIND_PATCH()
```

CANNOTFINDSTATEPARTITION

This operation could not be performed because the state partition could not be found

Signature:

```
CANNOT_FIND_STATE_PARTITION()
```

CANNOTFINDUPDATE

The requested update could not be found. Please upload the update again. This can occur when you run xe updatepoolclean before xe updateapply.

Signature:

```
CANNOT_FIND_UPDATE()
```

CANNOTFORGETSRIOVLOGICAL

This is a network SRIOV logical PIF and cannot do forget on it

Signature:

```
CANNOT_FORGET_SRIOV_LOGICAL(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTPLUGBONDSLAVE

This PIF is a bond member and cannot be plugged.

Signature:

```
CANNOT_PLUG_BOND_SLAVE(PIF)
```

CANNOTPLUGVIF

Cannot plug VIF

Signature:

```
CANNOT_PLUG_VIF(VIF)
```

CANNOTRESETCONTROLDOMAIN

The powerstate of a control domain cannot be reset.

Signature:

```
CANNOT_RESET_CONTROL_DOMAIN(vm)
```

CANNOTRESTARTDEVICEMODEL

Cannot restart device models of paused VMs residing on the host.

Signature:

```
CANNOT_RESTART_DEVICE_MODEL(ref)
```

CANNOTPERIODICSYNCUPDATES

The currently enabled repositories do not support periodic automatic updates.

Signature:

```
CAN_NOT_PERIODIC_SYNC_UPDATES()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CANNOTSYNCUPDATES

The currently enabled repositories do not support synchronization of updates.

Signature:

```
CAN_NOT_SYNC_UPDATES()
```

CERTIFICATEALREADYEXISTS

A certificate already exists with the specified name.

Signature:

```
CERTIFICATE_ALREADY_EXISTS(name)
```

CERTIFICATECORRUPT

The specified certificate is corrupt or unreadable.

Signature:

```
CERTIFICATE_CORRUPT(name)
```

CERTIFICATEDOESNOTEXIST

The specified certificate does not exist.

Signature:

```
CERTIFICATE_DOES_NOT_EXIST(name)
```

CERTIFICATELACKSPURPOSE

No purpose is specified for the provided certificate.

Signature:

```
CERTIFICATE_LACKS_PURPOSE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CERTIFICATELIBRARYCORRUPT

The certificate library is corrupt or unreadable.

Signature:

```
CERTIFICATE_LIBRARY_CORRUPT()
```

CERTIFICATENAMEINVALID

The specified certificate name is invalid.

Signature:

```
CERTIFICATE_NAME_INVALID(name)
```

CHANGEPASSWORDREJECTED

The system rejected the password change request; perhaps the new password was too short?

Signature:

```
CHANGE_PASSWORD_REJECTED(msg)
```

CLUSTEREDSRDEGRADED

An SR is using clustered local storage. It is not safe to reboot a host at the moment.

Signature:

```
CLUSTERED_SR_DEGRADED(sr)
```

CLUSTERINGDISABLED

An operation was attempted while clustering was disabled on the clusterhost.

Signature:

```
CLUSTERING_DISABLED(cluster_host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CLUSTERINGENABLED

An operation was attempted while clustering was enabled on the clusterhost.

Signature:

```
CLUSTERING_ENABLED(cluster_host)
```

CLUSTERALREADYEXISTS

A cluster already exists in the pool.

Signature:

```
CLUSTER_ALREADY_EXISTS()
```

CLUSTERCREATEINPROGRESS

The operation could not be performed because cluster creation is in progress.

Signature:

```
CLUSTER_CREATE_IN_PROGRESS()
```

CLUSTERDOESNOTHAVEONENODE

An operation failed as it expected the cluster to have only one node but found multiple clusterhosts.

Signature:

```
CLUSTER_DOES_NOT_HAVE_ONE_NODE(number_of_nodes)
```

CLUSTERFORCEDESTROYFAILED

Force destroy failed on a Clusterhost while force destroying the cluster.

Signature:

```
CLUSTER_FORCE_DESTROY_FAILED(cluster)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CLUSTERHOSTISLAST

The last cluster host cannot be destroyed. Destroy the cluster instead

Signature:

```
CLUSTER_HOST_IS_LAST(cluster_host)
```

CLUSTERHOSTNOTJOINED

Clusterhost operation failed as the clusterhost has not joined the cluster.

Signature:

```
CLUSTER_HOST_NOT_JOINED(cluster_host)
```

CLUSTERSTACKINUSE

The cluster stack is still in use by at least one plugged PBD.

Signature:

```
CLUSTER_STACK_IN_USE(cluster_stack)
```

CONFIGUREREPOSITORIESINPROGRESS

The operation could not be performed because other repositoryies isare already being configured.

Signature:

```
CONFIGURE_REPOSITORIES_IN_PROGRESS()
```

COULDNOTFINDNETWORKINTERFACEWITHSPECIFIEDDEVICENAMEANDMACADDRESS

Could not find a network interface with the specified device name and MAC address.

Signature:

```
COULD_NOT_FIND_NETWORK_INTERFACE_WITH_SPECIFIED_DEVICE_NAME_AND_MAC_ADDRES
S(device, mac)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

COULDNOTIMPORTDATABASE

An error occurred while attempting to import a database from a metadata VDI

Signature:

```
COULD_NOT_IMPORT_DATABASE(reason)
```

COULDNOTUPDATEIGMPSNOOPINGEVERYWHERE

The IGMP Snooping setting cannot be applied for some of the host, networks.

Signature:

```
COULD_NOT_UPDATE_IGMP_SNOOPING_EVERYWHERE()
```

CPUFEATUREMASKINGNOTSUPPORTED

The CPU does not support masking of features.

Signature:

```
CPU_FEATURE_MASKING_NOT_SUPPORTED(details)
```

CRLALREADYEXISTS

A CRL already exists with the specified name.

Signature:

```
CRL_ALREADY_EXISTS(name)
```

CRLCORRUPT

The specified CRL is corrupt or unreadable.

Signature:

```
CRL_CORRUPT(name)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

CRLDOESNOTEXIST

The specified CRL does not exist.

Signature:

```
CRL_DOES_NOT_EXIST(name)
```

CRLNAMEINVALID

The specified CRL name is invalid.

Signature:

```
CRL_NAME_INVALID(name)
```

DBUNIQUENESSCONSTRAINTVIOLATION

You attempted an operation which would have resulted in duplicate keys in the database.

Signature:

```
DB_UNIQUENESS_CONSTRAINT_VIOLATION(table, field, value)
```

DEFAULTSRNOTFOUND

The default SR reference does not point to a valid SR

Signature:

```
DEFAULT_SR_NOT_FOUND(sr)
```

DEVICEALREADYATTACHED

The device is already attached to a VM

Signature:

```
DEVICE_ALREADY_ATTACHED(device)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DEVICEALREADYDETACHED

The device is not currently attached

Signature:

```
DEVICE_ALREADY_DETACHED(device)
```

DEVICEALREADYEXISTS

A device with the name given already exists on the selected VM

Signature:

```
DEVICE_ALREADY_EXISTS(device)
```

DEVICEATTACHTIMEOUT

A timeout happened while attempting to attach a device to a VM.

Signature:

```
DEVICE_ATTACH_TIMEOUT(type, ref)
```

DEVICEDETACHREJECTED

The VM rejected the attempt to detach the device.

Signature:

```
DEVICE_DETACH_REJECTED(type, ref, msg)
```

DEVICEDETACHTIMEOUT

A timeout happened while attempting to detach a device from a VM.

Signature:

```
DEVICE_DETACH_TIMEOUT(type, ref)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DEVICENOTATTACHED

The operation could not be performed because the VBD was not connected to the VM.

Signature:

```
DEVICE_NOT_ATTACHED(VBD)
```

DISABLESSHFAILED

Failed to disable SSH access.

Signature:

```
DISABLE_SSH_FAILED(host)
```

DISABLESSHPARTIALLYFAILED

Some of hosts failed to disable SSH access.

Signature:

```
DISABLE_SSH_PARTIALLY_FAILED(hosts)
```

DISKVBDMUSTBEREADWRITEFORHVM

All VBDs of type 'disk' must be read/write for HVM guests

Signature:

```
DISK_VBD_MUST_BE_READWRITE_FOR_HVM(vbd)
```

DOMAINBUILDERERROR

An internal error generated by the domain builder.

Signature:

```
DOMAIN_BUILDER_ERROR(function, code, message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

DOMAINEXISTS

The operation could not be performed because a domain still exists for the specified VM.

Signature:

```
DOMAIN_EXISTS(vm, domid)
```

DUPLICATEMACSEED

This MAC seed is already in use by a VM in the pool

Signature:

```
DUPLICATE_MAC_SEED(seed)
```

DUPLICATEPIFDEVICENAME

A PIF with this specified device name already exists.

Signature:

```
DUPLICATE_PIF_DEVICE_NAME(device)
```

DUPLICATEVM

Cannot restore this VM because it would create a duplicate

Signature:

```
DUPLICATE_VM(vm)
```

ENABLESSHFAILED

Failed to enable SSH access.

Signature:

```
ENABLE_SSH_FAILED(host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

ENABLESSHPARTIALLYFAILED

Some of hosts failed to enable SSH access.

Signature:

```
ENABLE_SSH_PARTIALLY_FAILED(hosts)
```

EVENTSLOST

Some events have been lost from the queue and cannot be retrieved.

Signature:

```
EVENTS_LOST()
```

EVENTFROMTOKENPARSEFAILURE

The event.from token could not be parsed. Valid values include: '', and a value returned from a previous event.from call.

Signature:

```
EVENT_FROM_TOKEN_PARSE_FAILURE(token)
```

EVENTSUBSCRIPTIONPARSEFAILURE

The server failed to parse your event subscription. Valid values include: , classname, classname/objectreference.

Signature:

```
EVENT_SUBSCRIPTION_PARSE_FAILURE(subscription)
```

FAILEDTOSTARTEMULATOR

An emulator required to run this VM failed to start

Signature:

```
FAILED_TO_START_EMULATOR(vm, name, msg)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

FEATUREREQUIRESHVM

The VM is set up to use a feature that requires it to boot as HVM.

Signature:

```
FEATURE_REQUIRES_HVM(details)
```

FEATURERESTRICTED

The use of this feature is restricted.

Signature:

```
FEATURE_RESTRICTED()
```

FIELDTYPEERROR

The value specified is of the wrong type

Signature:

```
FIELD_TYPE_ERROR(field)
```

GETHOSTUPDATESFAILED

Failed to get available updates from a host.

Signature:

```
GET_HOST_UPDATES_FAILED(ref)
```

GETUPDATESFAILED

Failed to get available updates from the pool.

Signature:

```
GET_UPDATES_FAILED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

GPUGROUPCONTAINSNOPGPUS

The GPU group does not contain any PGPUs.

Signature:

```
GPU_GROUP_CONTAINS_NO_PGPUS(gpu_group)
```

GPUGROUPCONTAINSPGPU

The GPU group contains active PGPUs and cannot be deleted.

Signature:

```
GPU_GROUP_CONTAINS_PGPU(pgpus)
```

GPUGROUPCONTAINSVGPU

The GPU group contains active VGPUs and cannot be deleted.

Signature:

```
GPU_GROUP_CONTAINS_VGPU(vgpus)
```

HANDLEINVALID

You gave an invalid object reference. The object may have recently been deleted. The class parameter gives the type of reference given, and the handle parameter echoes the bad value given.

Signature:

```
HANDLE_INVALID(class, handle)
```

HAABORTNEWMASTER

This server cannot accept the proposed new coordinator setting at this time.

Signature:

```
HA_ABORT_NEW_MASTER(reason)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HACANNOTCHANGEBONDSTATUSOFMGMTIFACE

This operation cannot be performed because creating or deleting a bond involving the management interface is not allowed while HA is on. In order to do that, disable HA, create or delete the bond then reenable HA.

Signature:

```
HA_CANNOT_CHANGE_BOND_STATUS_OF_MGMT_IFACE()
```

HACONSTRAINTVIOLATIONNETWORKNOTSHARED

This operation cannot be performed because the referenced network is not properly shared. The network must either be entirely virtual or must be physically present via a currentlyattached PIF on every host.

Signature:

```
HA_CONSTRAINT_VIOLATION_NETWORK_NOT_SHARED(network)
```

HACONSTRAINTVIOLATIONSRNOTSHARED

This operation cannot be performed because the referenced SR is not properly shared. The SR must both be marked as shared and a currentlyattached PBD must exist for each host.

Signature:

```
HA_CONSTRAINT_VIOLATION_SR_NOT_SHARED(SR)
```

HADISABLEINPROGRESS

The operation could not be performed because HA disable is in progress

Signature:

```
HA_DISABLE_IN_PROGRESS()
```

HAENABLEINPROGRESS

The operation could not be performed because HA enable is in progress

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
HA_ENABLE_IN_PROGRESS()
```

HAFAILEDTOFORMLIVESET

HA could not be enabled on the Pool because a liveset could not be formed: check storage and network heartbeat paths.

Signature:

```
HA_FAILED_TO_FORM_LIVESET()
```

HAHEARTBEATDAEMONSTARTUPFAILED

The server could not join the liveset because the HA daemon failed to start.

Signature:

```
HA_HEARTBEAT_DAEMON_STARTUP_FAILED()
```

HAHOSTCANNOTACCESSSTATEFILE

The server could not join the liveset because the HA daemon could not access the heartbeat disk.

Signature:

```
HA_HOST_CANNOT_ACCESS_STATEFILE()
```

HAHOSTCANNOTSEEPEERS

The operation failed because the HA software on the specified server could not see a subset of other servers. Check your network connectivity.

Signature:

```
HA_HOST_CANNOT_SEE_PEERS(host, all, subset)
```

HAHOSTISARMED

The operation could not be performed while the server is still armed; it must be disarmed first.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
HA_HOST_IS_ARMED(host)
```

HAISENABLED

The operation could not be performed because HA is enabled on the Pool

Signature:

```
HA_IS_ENABLED()
```

HALOSTSTATEFILE

This server lost access to the HA statefile.

Signature:

```
HA_LOST_STATEFILE()
```

HANOTENABLED

The operation could not be performed because HA is not enabled on the Pool

Signature:

```
HA_NOT_ENABLED()
```

HANOTINSTALLED

The operation could not be performed because the HA software is not installed on this server.

Signature:

```
HA_NOT_INSTALLED(host)
```

HANOPLAN

Cannot find a plan for placement of VMs as there are no other servers available.

Signature:

```
HA_NO_PLAN()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HAOPERATIONWOULDBREAKFAILOVERPLAN

This operation cannot be performed because it would invalidate VM failover planning such that the system would be unable to guarantee to restart protected VMs after a Host failure.

Signature:

```
HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN()
```

HAPOOLISENABLEDBUTHOSTISDISABLED

This server cannot join the pool because the pool has HA enabled but this server has HA disabled.

Signature:

```
HA_POOL_IS_ENABLED_BUT_HOST_IS_DISABLED()
```

HASHOULDBEFENCED

Server cannot rejoin pool because it should have fenced it is not in the coordinator's partition.

Signature:

```
HA_SHOULD_BE_FENCED(host)
```

HATOOFEWHOSTS

HA can only be enabled for 2 servers or more. Note that 2 servers requires a preconfigured quorum tiebreak script.

Signature:

```
HA_TOO_FEW_HOSTS()
```

HOSTSNOTCOMPATIBLE

The hosts in this pool are not compatible.

Signature:

```
HOSTS_NOT_COMPATIBLE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTSNOTHOMOGENEOUS

The hosts in this pool are not homogeneous.

Signature:

```
HOSTS_NOT_HOMOGENEOUS(reason)
```

HOSTBROKEN

This server failed in the middle of an automatic failover operation and needs to retry the failover action.

Signature:

```
HOST_BROKEN()
```

HOSTCANNOTATTACHNETWORK

Server cannot attach network in the case of NIC bonding, this may be because attaching the network on this server would require other networks that are currently active to be taken down.

Signature:

```
HOST_CANNOT_ATTACH_NETWORK(host, network)
```

HOSTCANNOTDESTROYSELF

The pool coordinator host cannot be removed.

Signature:

```
HOST_CANNOT_DESTROY_SELF(host)
```

HOSTCANNOTREADMETRICS

The metrics of this server could not be read.

Signature:

```
HOST_CANNOT_READ_METRICS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTCDDRIVEEMPTY

The host CDROM drive does not contain a valid CD

Signature:

```
HOST_CD_DRIVE_EMPTY()
```

HOSTDISABLED

The specified server is disabled.

Signature:

```
HOST_DISABLED(host)
```

HOSTDISABLEDINDEFINITELY

The specified server is disabled and can only be reenabled manually with Host.enable.

Signature:

```
HOST_DISABLED_INDEFINITELY(host)
```

HOSTDISABLEDUNTILREBOOT

The specified server is disabled and cannot be reenabled until after it has rebooted.

Signature:

```
HOST_DISABLED_UNTIL_REBOOT(host)
```

HOSTDRIVERNOHARDWARE

No hardware present for this host driver variant

Signature:

```
HOST_DRIVER_NO_HARDWARE(driver variant)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTEVACUATEINPROGRESS

This host is being evacuated.

Signature:

```
HOST_EVACUATE_IN_PROGRESS(host)
```

HOSTEVACUATIONISREQUIRED

Host evacuation is required before applying updates.

Signature:

```
HOST_EVACUATION_IS_REQUIRED(host)
```

HOSTHASNOMANAGEMENTIP

The server failed to acquire an IP address on its management interface and therefore cannot contact the coordinator.

Signature:

```
HOST_HAS_NO_MANAGEMENT_IP()
```

HOSTHASRESIDENTVMS

This server cannot be forgotten because there are user VMs still running.

Signature:

```
HOST_HAS_RESIDENT_VMS(host)
```

HOSTINEMERGENCYMODE

Cannot perform operation as the host is running in emergency mode.

Signature:

```
HOST_IN_EMERGENCY_MODE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

HOSTINUSE

This operation cannot be completed as the host is in use by at least the object of type and ref echoed below.

Signature:

```
HOST_IN_USE(host, type, ref)
```

HOSTISLIVE

This operation cannot be completed because the server is still live.

Signature:

```
HOST_IS_LIVE(host)
```

HOSTISSLAVE

You cannot make regular API calls directly on a supporter. Please pass API calls via the coordinator host.

Signature:

```
HOST_IS_SLAVE(Master IP address)
```

HOSTITSOWNSLAVE

The host is its own supporter. Please use poolemergencytransitiontomaster or poolemergencyresetmaster.

Signature:

```
HOST_ITS_OWN_SLAVE()
```

HOSTMASTERCANNOTTALKBACK

The coordinator reports that it cannot talk back to the supporter on the supplied management IP address.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
HOST_MASTER_CANNOT_TALK_BACK(ip)
```

HOSTNAMEINVALID

The server name is invalid.

Signature:

```
HOST_NAME_INVALID(reason)
```

HOSTNOTDISABLED

This operation cannot be performed because the host is not disabled. Please disable the host and then try again.

Signature:

```
HOST_NOT_DISABLED()
```

HOSTNOTENOUGHFREEMEMORY

Not enough server memory is available to perform this operation.

Signature:

```
HOST_NOT_ENOUGH_FREE_MEMORY(needed, available)
```

HOSTNOTENOUGHPCPUS

The host does not have enough pCPUs to run the VM. It needs at least as many as the VM has vCPUs.

Signature:

```
HOST_NOT_ENOUGH_PCPUS(vcpus, pcpus)
```

HOSTNOTLIVE

This operation cannot be completed as the server is not live.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
HOST_NOT_LIVE()
```

HOSTOFFLINE

You attempted an operation which involves a host which could not be contacted.

Signature:

```
HOST_OFFLINE(host)
```

HOSTPENDINGMANDATORYGUIDANCENOTEMPTY

Operation could not be performed on the host because there is pending mandatory update guidance on it.

Signature:

```
HOST_PENDING_MANDATORY_GUIDANCE_NOT_EMPTY(host)
```

HOSTPOWERONMODEDISABLED

This operation cannot be completed because the server power on mode is disabled.

Signature:

```
HOST_POWER_ON_MODE_DISABLED()
```

HOSTSTILLBOOTING

The host toolstack is still initialising. Please wait.

Signature:

```
HOST_STILL_BOOTING()
```

HOSTUNKNOWNTOMASTER

The coordinator says the server is not known to it. Is the server in the coordinator's database and pointing to the correct coordinator? Are all servers using the same pool secret?

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
HOST_UNKNOWN_TO_MASTER(host)
```

HOSTXAPIVERSIONHIGHERTHANCOORDINATOR

The host xapi version is higher than the one in the coordinator

Signature:

```
HOST_XAPI_VERSION_HIGHER_THAN_COORDINATOR(host_xapi_version)
```

ILLEGALVBDDEVICE

The specified VBD device is not recognized: please use a nonnegative integer

Signature:

```
ILLEGAL_VBD_DEVICE(vbd, device)
```

IMPORTERROR

The VM could not be imported.

Signature:

```
IMPORT_ERROR(msg)
```

IMPORTERRORATTACHEDDISKSNOTFOUND

The VM could not be imported because attached disks could not be found.

Signature:

```
IMPORT_ERROR_ATTACHED_DISKS_NOT_FOUND()
```

IMPORTERRORCANNOTHANDLECHUNKED

Cannot import VM using chunked encoding.

Signature:

```
IMPORT_ERROR_CANNOT_HANDLE_CHUNKED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

IMPORTERRORFAILEDTOFINDOBJECT

The VM could not be imported because a required object could not be found.

Signature:

```
IMPORT_ERROR_FAILED_TO_FIND_OBJECT(id)
```

IMPORTERRORPREMATUREEOF

The VM could not be imported; the end of the file was reached prematurely.

Signature:

```
IMPORT_ERROR_PREMATURE_EOF()
```

IMPORTERRORSOMECHECKSUMSFAILED

Some data checksums were incorrect; the VM may be corrupt.

Signature:

```
IMPORT_ERROR_SOME_CHECKSUMS_FAILED()
```

IMPORTERRORUNEXPECTEDFILE

The VM could not be imported because the XVA file is invalid: an unexpected file was encountered.

Signature:

```
IMPORT_ERROR_UNEXPECTED_FILE(filename_expected, filename_found)
```

IMPORTINCOMPATIBLEVERSION

The import failed because this export has been created by a different incompatible product version

Signature:

```
IMPORT_INCOMPATIBLE_VERSION()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INCOMPATIBLECLUSTERSTACKACTIVE

This operation cannot be performed, because it is incompatible with the currently active HA cluster stack.

Signature:

```
INCOMPATIBLE_CLUSTER_STACK_ACTIVE(cluster_stack)
```

INCOMPATIBLEPIFPROPERTIES

These PIFs cannot be bonded, because their properties are different.

Signature:

```
INCOMPATIBLE_PIF_PROPERTIES()
```

INCOMPATIBLESTATEFILESR

The specified SR is incompatible with the selected HA cluster stack.

Signature:

```
INCOMPATIBLE_STATEFILE_SR(SR type)
```

INTERFACEHASNOIP

The specified interface cannot be used because it has no IP address

Signature:

```
INTERFACE_HAS_NO_IP(interface)
```

INTERNALERROR

The server failed to handle your request, due to an internal error. The given message may give details useful for debugging the problem.

Signature:

```
INTERNAL_ERROR(message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDBASEURL

The base url in the repository is invalid.

Signature:

```
INVALID_BASE_URL(url)
```

INVALIDCIDRADDRESSSPECIFIED

A required parameter contained an invalid CIDR address <addr>/<prefix length>

Signature:

```
INVALID_CIDR_ADDRESS_SPECIFIED(parameter)
```

INVALIDCLUSTERSTACK

The cluster stack provided is not supported.

Signature:

```
INVALID_CLUSTER_STACK(cluster_stack)
```

INVALIDDEVICE

The device name is invalid

Signature:

```
INVALID_DEVICE(device)
```

INVALIDEDITION

The edition you supplied is invalid.

Signature:

```
INVALID_EDITION(edition)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDFEATURESTRING

The given feature string is not valid.

Signature:

```
INVALID_FEATURE_STRING(details)
```

INVALIDGPGKEYPATH

The GPG public key file name in the repository is invalid.

Signature:

```
INVALID_GPGKEY_PATH(gpgkey_path)
```

INVALIDIPADDRESSSPECIFIED

A required parameter contained an invalid IP address

Signature:

```
INVALID_IP_ADDRESS_SPECIFIED(parameter)
```

INVALIDNTPCONFIG

The NTP configuration is invalid.

Signature:

```
INVALID_NTP_CONFIG(reason)
```

INVALIDPATCH

The uploaded patch file is invalid

Signature:

```
INVALID_PATCH()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDPATCHWITHLOG

The uploaded patch file is invalid. See attached log for more details.

Signature:

```
INVALID_PATCH_WITH_LOG(log)
```

INVALIDREPOMDXML

The repomd.xml is invalid.

Signature:

```
INVALID_REPOMD_XML()
```

INVALIDREPOSITORYDOMAINALLOWLIST

The repository domain allowlist has some invalid domains.

Signature:

```
INVALID_REPOSITORY_DOMAIN_ALLOWLIST(domains)
```

INVALIDREPOSITORYPROXYCREDENTIAL

The repository proxy username/password is invalid.

Signature:

```
INVALID_REPOSITORY_PROXY_CREDENTIAL()
```

INVALIDREPOSITORYPROXYURL

The repository proxy URL is invalid.

Signature:

```
INVALID_REPOSITORY_PROXY_URL(url)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

INVALIDUPDATE

The uploaded update package is invalid.

Signature:

```
INVALID_UPDATE(info)
```

INVALIDUPDATEINFOXML

The updateinfo.xml is invalid.

Signature:

```
INVALID_UPDATEINFO_XML()
```

INVALIDUPDATESYNCDAY

Invalid day of the week chosen for weekly update sync.

Signature:

```
INVALID_UPDATE_SYNC_DAY(day)
```

INVALIDVALUE

The value given is invalid

Signature:

```
INVALID_VALUE(field, value)
```

ISTUNNELACCESSPIF

Cannot create a VLAN or tunnel on top of a tunnel access PIF use the underlying transport PIF instead.

Signature:

```
IS_TUNNEL_ACCESS_PIF(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

JOININGHOSTCANNOTBEMASTEROFOTHERHOSTS

The server joining the pool cannot already be a coordinator of another pool.

Signature:

```
JOINING_HOST_CANNOT_BE_MASTER_OF_OTHER_HOSTS()
```

JOININGHOSTCANNOTCONTAINSHAREDSRS

The server joining the pool cannot contain any shared storage.

Signature:

```
JOINING_HOST_CANNOT_CONTAIN_SHARED_SRS()
```

JOININGHOSTCANNOTHAVERUNNINGORSUSPENDEDVMS

The server joining the pool cannot have any running or suspended VMs.

Signature:

```
JOINING_HOST_CANNOT_HAVE_RUNNING_OR_SUSPENDED_VMS()
```

JOININGHOSTCANNOTHAVERUNNINGVMS

The server joining the pool cannot have any running VMs.

Signature:

```
JOINING_HOST_CANNOT_HAVE_RUNNING_VMS()
```

JOININGHOSTCANNOTHAVEVMSWITHCURRENTOPERATIONS

The host joining the pool cannot have any VMs with active tasks.

Signature:

```
JOINING_HOST_CANNOT_HAVE_VMS_WITH_CURRENT_OPERATIONS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

JOININGHOSTCONNECTIONFAILED

There was an error connecting to the host while joining it in the pool.

Signature:

```
JOINING_HOST_CONNECTION_FAILED()
```

JOININGHOSTSERVICEFAILED

There was an error connecting to the server. The service contacted didn't reply properly.

Signature:

```
JOINING_HOST_SERVICE_FAILED()
```

LICENCERESTRICTION

This operation is not allowed because your license lacks a needed feature. Please contact your support representative.

Signature:

```
LICENCE_RESTRICTION(feature)
```

LICENSECANNOTDOWNGRADEWHILEINPOOL

Cannot downgrade license while in pool. Please disband the pool first, then downgrade licenses on hosts separately.

Signature:

```
LICENSE_CANNOT_DOWNGRADE_WHILE_IN_POOL()
```

LICENSECHECKOUTERROR

The license for the edition you requested is not available.

Signature:

```
LICENSE_CHECKOUT_ERROR(code, message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

LICENSEDOESNOTSUPPORTPOOLING

This server cannot join a pool because its license does not support pooling.

Signature:

```
LICENSE_DOES_NOT_SUPPORT_POOLING()
```

LICENSEDOESNOTSUPPORTXHA

HA cannot be enabled because this server's license does not allow it.

Signature:

```
LICENSE_DOES_NOT_SUPPORT_XHA()
```

LICENSEEXPIRED

Your license has expired. Please contact your support representative.

Signature:

```
LICENSE_EXPIRED()
```

LICENSEFILEDEPRECATED

This type of license file is for previous versions of the server. Please upgrade to the new licensing system.

Signature:

```
LICENSE_FILE_DEPRECATED()
```

LICENSEHOSTPOOLMISMATCH

Host and pool have incompatible licenses editions.

Signature:

```
LICENSE_HOST_POOL_MISMATCH()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

LICENSEPROCESSINGERROR

There was an error processing your license. Please contact your support representative.

Signature:

```
LICENSE_PROCESSING_ERROR()
```

LOCATIONNOTUNIQUE

A VDI with the specified location already exists within the SR

Signature:

```
LOCATION_NOT_UNIQUE(SR, location)
```

MACDOESNOTEXIST

The MAC address specified does not exist on this server.

Signature:

```
MAC_DOES_NOT_EXIST(MAC)
```

MACINVALID

The MAC address specified is not valid.

Signature:

```
MAC_INVALID(MAC)
```

MACSTILLEXISTS

The MAC address specified still exists on this server.

Signature:

```
MAC_STILL_EXISTS(MAC)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

MAPDUPLICATEKEY

You tried to add a keyvalue pair to a map, but that key is already there.

Signature:

```
MAP_DUPLICATE_KEY(type, param_name, uuid, key)
```

MEMORYCONSTRAINTVIOLATION

The dynamic memory range does not satisfy the following constraint.

Signature:

```
MEMORY_CONSTRAINT_VIOLATION(constraint)
```

MEMORYCONSTRAINTVIOLATIONMAXPIN

The dynamic memory range violates constraint staticmin = dynamicmin = dynamicmax = staticmax.

Signature:

```
MEMORY_CONSTRAINT_VIOLATION_MAXPIN(reason)
```

MEMORYCONSTRAINTVIOLATIONORDER

The dynamic memory range violates constraint staticmin <= dynamicmin <= dynamicmax <= staticmax.

Signature:

```
MEMORY_CONSTRAINT_VIOLATION_ORDER()
```

MESSAGEDEPRECATED

This message has been deprecated.

Signature:

```
MESSAGE_DEPRECATED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

MESSAGEMETHODUNKNOWN

You tried to call a method that does not exist. The method name that you used is echoed.

Signature:

```
MESSAGE_METHOD_UNKNOWN(method)
```

MESSAGEPARAMETERCOUNTMISMATCH

You tried to call a method with the incorrect number of parameters. The fullyqualified method name that you used, and the number of received and expected parameters are returned.

Signature:

```
MESSAGE_PARAMETER_COUNT_MISMATCH(method, expected, received)
```

MESSAGEREMOVED

This function is no longer available.

Signature:

```
MESSAGE_REMOVED()
```

MIRRORFAILED

The VDI mirroring cannot be performed

Signature:

```
MIRROR_FAILED(vdi)
```

MISSINGCONNECTIONDETAILS

The licenseserver connection details address or port were missing or incomplete.

Signature:

```
MISSING_CONNECTION_DETAILS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

MULTIPLEUPDATEREPOSITORIESENABLED

There is more than one update repository being enabled.

Signature:

```
MULTIPLE_UPDATE_REPOSITORIES_ENABLED()
```

NETWORKALREADYCONNECTED

You tried to create a PIF, but the network you tried to attach it to is already attached to some other PIF, and so the creation failed.

Signature:

```
NETWORK_ALREADY_CONNECTED(network, connected PIF)
```

NETWORKCONTAINSPIF

The network contains active PIFs and cannot be deleted.

Signature:

```
NETWORK_CONTAINS_PIF(pifs)
```

NETWORKCONTAINSVIF

The network contains active VIFs and cannot be deleted.

Signature:

```
NETWORK_CONTAINS_VIF(vifs)
```

NETWORKHASINCOMPATIBLESRIOVPIFS

The PIF is not compatible with the selected SRIOV network

Signature:

```
NETWORK_HAS_INCOMPATIBLE_SRIOV_PIFS(PIF, network)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKHASINCOMPATIBLEVLANONSRIOVPIFS

VLAN on the PIF is not compatible with the selected SRIOV VLAN network

Signature:

```
NETWORK_HAS_INCOMPATIBLE_VLAN_ON_SRIOV_PIFS(PIF, network)
```

NETWORKINCOMPATIBLEPURPOSES

You tried to add a purpose to a network but the new purpose is not compatible with an existing purpose of the network or other networks.

Signature:

```
NETWORK_INCOMPATIBLE_PURPOSES(new_purpose, conflicting_purpose)
```

NETWORKINCOMPATIBLEWITHBOND

The network is incompatible with bond

Signature:

```
NETWORK_INCOMPATIBLE_WITH_BOND(network)
```

NETWORKINCOMPATIBLEWITHSRIOV

The network is incompatible with sriov

Signature:

```
NETWORK_INCOMPATIBLE_WITH_SRIOV(network)
```

NETWORKINCOMPATIBLEWITHTUNNEL

The network is incompatible with tunnel

Signature:

```
NETWORK_INCOMPATIBLE_WITH_TUNNEL(network)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKINCOMPATIBLEWITHVLANONBRIDGE

The network is incompatible with vlan on bridge

Signature:

```
NETWORK_INCOMPATIBLE_WITH_VLAN_ON_BRIDGE(network)
```

NETWORKINCOMPATIBLEWITHVLANONSRIOV

The network is incompatible with vlan on sriov

Signature:

```
NETWORK_INCOMPATIBLE_WITH_VLAN_ON_SRIOV(network)
```

NETWORKSRIOVALREADYENABLED

The PIF selected for the SRIOV network is already enabled

Signature:

```
NETWORK_SRIOV_ALREADY_ENABLED(PIF)
```

NETWORKSRIOVDISABLEFAILED

Failed to disable SRIOV on PIF

Signature:

```
NETWORK_SRIOV_DISABLE_FAILED(PIF, msg)
```

NETWORKSRIOVENABLEFAILED

Failed to enable SRIOV on PIF

Signature:

```
NETWORK_SRIOV_ENABLE_FAILED(PIF, msg)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NETWORKSRIOVINSUFFICIENTCAPACITY

There is insufficient capacity for VF reservation

Signature:

```
NETWORK_SRIOV_INSUFFICIENT_CAPACITY(network)
```

NETWORKUNMANAGED

The network is not managed by xapi.

Signature:

```
NETWORK_UNMANAGED(network)
```

NOTALLOWEDONOEMEDITION

This command is not allowed on the OEM edition.

Signature:

```
NOT_ALLOWED_ON_OEM_EDITION(command)
```

NOTALLOWEDWHENNTPISENABLED

The operation is not allowed on the host when the NTP is enabled.

Signature:

```
NOT_ALLOWED_WHEN_NTP_IS_ENABLED(host)
```

NOTIMPLEMENTED

The function is not implemented

Signature:

```
NOT_IMPLEMENTED(function)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTINEMERGENCYMODE

This pool is not in emergency mode.

Signature:

```
NOT_IN_EMERGENCY_MODE()
```

NOTSUPPORTEDDURINGUPGRADE

This operation is not supported during an upgrade.

Signature:

```
NOT_SUPPORTED_DURING_UPGRADE()
```

NOTSYSTEMDOMAIN

The given VM is not registered as a system domain. This operation can only be performed on a registered system domain.

Signature:

```
NOT_SYSTEM_DOMAIN(vm)
```

NOTTRUSTEDCERTIFICATE

The provided certificate is not a trusted certificate.

Signature:

```
NOT_TRUSTED_CERTIFICATE(ref)
```

NOCLUSTERHOSTSREACHABLE

No other cluster host was reachable when joining

Signature:

```
NO_CLUSTER_HOSTS_REACHABLE(cluster)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOCOMPATIBLECLUSTERHOST

Clustering is not enabled on this host or pool.

Signature:

```
NO_COMPATIBLE_CLUSTER_HOST(host)
```

NOHOSTSAVAILABLE

There were no servers available to complete the specified operation.

Signature:

```
NO_HOSTS_AVAILABLE()
```

NOMOREREDOLOGSALLOWED

The upper limit of active redo log instances was reached.

Signature:

```
NO_MORE_REDO_LOGS_ALLOWED()
```

NOREPOSITORIESCONFIGURED

No update repositories have been configured.

Signature:

```
NO_REPOSITORIES_CONFIGURED()
```

NOREPOSITORYENABLED

There is no repository being enabled.

Signature:

```
NO_REPOSITORY_ENABLED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NVIDIASRIOVMISCONFIGURED

The NVidia GPU is not configured for SRIOV as expected

Signature:

```
NVIDIA_SRIOV_MISCONFIGURED(host, device_name)
```

NVIDIATOOLSERROR

Nvidia tools error. Please ensure that the latest Nvidia tools are installed

Signature:

```
NVIDIA_TOOLS_ERROR(host)
```

OBJECTNOLONGEREXISTS

The specified object no longer exists.

Signature:

```
OBJECT_NOLONGER_EXISTS()
```

ONLYALLOWEDONOEMEDITION

This command is only allowed on the OEM edition.

Signature:

```
ONLY_ALLOWED_ON_OEM_EDITION(command)
```

OPENVSWITCHNOTACTIVE

This operation needs the OpenVSwitch networking backend to be enabled on all hosts in the pool.

Signature:

```
OPENVSWITCH_NOT_ACTIVE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

OPERATIONBLOCKED

You attempted an operation that was explicitly blocked see the blockedoperations field of the given object.

Signature:

```
OPERATION_BLOCKED(ref, code)
```

OPERATIONNOTALLOWED

You attempted an operation that was not allowed.

Signature:

```
OPERATION_NOT_ALLOWED(reason)
```

OPERATIONPARTIALLYFAILED

Some VMs belonging to the appliance threw an exception while carrying out the specified operation

Signature:

```
OPERATION_PARTIALLY_FAILED(operation)
```

OTHEROPERATIONINPROGRESS

Another operation involving the object is currently in progress

Signature:

```
OTHER_OPERATION_IN_PROGRESS(class, object, operation_type, operation_ref)
```

OUTOFSPACE

There is not enough space to upload the update

Signature:

```
OUT_OF_SPACE(location)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PATCHALREADYAPPLIED

This patch has already been applied

Signature:

```
PATCH_ALREADY_APPLIED(patch)
```

PATCHALREADYEXISTS

The uploaded patch file already exists

Signature:

```
PATCH_ALREADY_EXISTS(uuid)
```

PATCHAPPLYFAILED

The patch apply failed. Please see attached output.

Signature:

```
PATCH_APPLY_FAILED(output)
```

PATCHAPPLYFAILEDBACKUPFILESEXIST

The patch apply failed: there are backup files created while applying patch. Please remove these backup files before applying patch again.

Signature:

```
PATCH_APPLY_FAILED_BACKUP_FILES_EXIST(output)
```

PATCHISAPPLIED

The specified patch is applied and cannot be destroyed.

Signature:

```
PATCH_IS_APPLIED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PATCHPRECHECKFAILEDISOMOUNTED

Tools ISO must be ejected from all running VMs.

Signature:

```
PATCH_PRECHECK_FAILED_ISO_MOUNTED(patch)
```

PATCHPRECHECKFAILEDOUTOFSPACE

The patch precheck stage failed: the server does not have enough space.

Signature:

```
PATCH_PRECHECK_FAILED_OUT_OF_SPACE(patch, found_space, required_required)
```

PATCHPRECHECKFAILEDPREREQUISITEMISSING

The patch precheck stage failed: prerequisite patches are missing.

Signature:

```
PATCH_PRECHECK_FAILED_PREREQUISITE_MISSING(patch,
prerequisite_patch_uuid_list)
```

PATCHPRECHECKFAILEDUNKNOWNERROR

The patch precheck stage failed with an unknown error. See attached info for more details.

Signature:

```
PATCH_PRECHECK_FAILED_UNKNOWN_ERROR(patch, info)
```

PATCHPRECHECKFAILEDVMRUNNING

The patch precheck stage failed: there are one or more VMs still running on the server. All VMs must be suspended before the patch can be applied.

Signature:

```
PATCH_PRECHECK_FAILED_VM_RUNNING(patch)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PATCHPRECHECKFAILEDWRONGSERVERBUILD

The patch precheck stage failed: the server is of an incorrect build.

Signature:

```
PATCH_PRECHECK_FAILED_WRONG_SERVER_BUILD(patch, found_build,
required_build)
```

PATCHPRECHECKFAILEDWRONGSERVERVERSION

The patch precheck stage failed: the server is of an incorrect version.

Signature:

```
PATCH_PRECHECK_FAILED_WRONG_SERVER_VERSION(patch, found_version,
required_version)
```

PBDEXISTS

A PBD already exists connecting the SR to the server.

Signature:

```
PBD_EXISTS(sr, host, pbd)
```

PERMISSIONDENIED

Caller not allowed to perform this operation.

Signature:

```
PERMISSION_DENIED(message)
```

PGPUINSUFFICIENTCAPACITYFORVGPU

There is insufficient capacity on this PGPU to run the VGPU.

Signature:

```
PGPU_INSUFFICIENT_CAPACITY_FOR_VGPU(pgpu, vgpu_type)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PGPUINUSEBYVM

This PGPU is currently in use by running VMs.

Signature:

```
PGPU_IN_USE_BY_VM(VMs)
```

PGPUNOTCOMPATIBLEWITHGPUGROUP

PGPU type not compatible with destination group.

Signature:

```
PGPU_NOT_COMPATIBLE_WITH_GPU_GROUP(type, group_types)
```

PIFALLOWSUNPLUG

The operation you requested cannot be performed because the specified PIF allows unplug.

Signature:

```
PIF_ALLOWS_UNPLUG(PIF)
```

PIFALREADYBONDED

This operation cannot be performed because the pif is bonded.

Signature:

```
PIF_ALREADY_BONDED(PIF)
```

PIFBONDMORETHANONEIP

Only one PIF on a bond is allowed to have an IP configuration.

Signature:

```
PIF_BOND_MORE_THAN_ONE_IP()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFBONDNEEDSMOREMEMBERS

A bond must consist of at least two member interfaces

Signature:

```
PIF_BOND_NEEDS_MORE_MEMBERS()
```

PIFCANNOTBONDCROSSHOST

You cannot bond interfaces across different servers.

Signature:

```
PIF_CANNOT_BOND_CROSS_HOST()
```

PIFCONFIGURATIONERROR

An unknown error occurred while attempting to configure an interface.

Signature:

```
PIF_CONFIGURATION_ERROR(PIF, msg)
```

PIFDEVICENOTFOUND

The specified device was not found.

Signature:

```
PIF_DEVICE_NOT_FOUND()
```

PIFDOESNOTALLOWUNPLUG

The operation you requested cannot be performed because the specified PIF does not allow unplug.

Signature:

```
PIF_DOES_NOT_ALLOW_UNPLUG(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFHASFCOESRINUSE

The operation you requested cannot be performed because the specified PIF has FCoE SR in use.

Signature:

```
PIF_HAS_FCOE_SR_IN_USE(PIF, SR)
```

PIFHASNONETWORKCONFIGURATION

PIF has no IP configuration mode currently set to 'none'

Signature:

```
PIF_HAS_NO_NETWORK_CONFIGURATION(PIF)
```

PIFHASNOV6NETWORKCONFIGURATION

PIF has no IPv6 configuration mode currently set to 'none'

Signature:

```
PIF_HAS_NO_V6_NETWORK_CONFIGURATION(PIF)
```

PIFINCOMPATIBLEPRIMARYADDRESSTYPE

The primary address types are not compatible

Signature:

```
PIF_INCOMPATIBLE_PRIMARY_ADDRESS_TYPE(PIF)
```

PIFISMANAGEMENTINTERFACE

The operation you requested cannot be performed because the specified PIF is the management interface.

Signature:

```
PIF_IS_MANAGEMENT_INTERFACE(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFISNOTPHYSICAL

You tried to perform an operation which is only available on physical PIF

Signature:

```
PIF_IS_NOT_PHYSICAL(PIF)
```

PIFISNOTSRIOVCAPABLE

The selected PIF is not capable of network SRIOV

Signature:

```
PIF_IS_NOT_SRIOV_CAPABLE(PIF)
```

PIFISPHYSICAL

You tried to destroy a PIF, but it represents an aspect of the physical host configuration, and so cannot be destroyed. The parameter echoes the PIF handle you gave.

Signature:

```
PIF_IS_PHYSICAL(PIF)
```

PIFISSRIOVLOGICAL

You tried to create a bond on top of a network SRIOV logical PIF use the underlying physical PIF instead

Signature:

```
PIF_IS_SRIOV_LOGICAL(PIF)
```

PIFISVLAN

You tried to create a VLAN on top of another VLAN use the underlying physical PIF/bond instead

Signature:

```
PIF_IS_VLAN(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFNOTATTACHEDTOHOST

Clusterhost creation failed as the PIF provided is not attached to the host.

Signature:

```
PIF_NOT_ATTACHED_TO_HOST(pif, host)
```

PIFNOTPRESENT

This host has no PIF on the given network.

Signature:

```
PIF_NOT_PRESENT(host, network)
```

PIFSRIOVSTILLEXISTS

The PIF is still related with a network SRIOV

Signature:

```
PIF_SRIOV_STILL_EXISTS(PIF)
```

PIFTUNNELSTILLEXISTS

Operation cannot proceed while a tunnel exists on this interface.

Signature:

```
PIF_TUNNEL_STILL_EXISTS(PIF)
```

PIFUNMANAGED

The operation you requested cannot be performed because the specified PIF is not managed by xapi.

Signature:

```
PIF_UNMANAGED(PIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

PIFVLANEXISTS

You tried to create a PIF, but it already exists.

Signature:

```
PIF_VLAN_EXISTS(PIF)
```

PIFVLANSTILLEXISTS

Operation cannot proceed while a VLAN exists on this interface.

Signature:

```
PIF_VLAN_STILL_EXISTS(PIF)
```

POOLAUTHALREADYENABLED

External authentication is already enabled for at least one server in this pool.

Signature:

```
POOL_AUTH_ALREADY_ENABLED(host)
```

POOLAUTHDISABLEFAILED

The pool failed to disable the external authentication of at least one host.

Signature:

```
POOL_AUTH_DISABLE_FAILED(host, message)
```

POOLAUTHDISABLEFAILEDINVALIDACCOUNT

External authentication has been disabled with errors: Some AD machine accounts were not disabled on the AD server due to invalid account.

Signature:

```
POOL_AUTH_DISABLE_FAILED_INVALID_ACCOUNT(host, message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLAUTHDISABLEFAILEDPERMISSIONDENIED

External authentication has been disabled with errors: Your AD machine account was not disabled on the AD server as permission was denied.

Signature:

```
POOL_AUTH_DISABLE_FAILED_PERMISSION_DENIED(host, message)
```

POOLAUTHDISABLEFAILEDWRONGCREDENTIALS

External authentication has been disabled with errors: Some AD machine accounts were not disabled on the AD server due to invalid credentials.

Signature:

```
POOL_AUTH_DISABLE_FAILED_WRONG_CREDENTIALS(host, message)
```

POOLAUTHENABLEFAILED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED(host, message)
```

POOLAUTHENABLEFAILEDDOMAINLOOKUPFAILED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED(host, message)
```

POOLAUTHENABLEFAILEDDUPLICATEHOSTNAME

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_DUPLICATE_HOSTNAME(host, message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLAUTHENABLEFAILEDINVALIDACCOUNT

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_INVALID_ACCOUNT(host, message)
```

POOLAUTHENABLEFAILEDINVALIDOU

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_INVALID_OU(host, message)
```

POOLAUTHENABLEFAILEDPERMISSIONDENIED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_PERMISSION_DENIED(host, message)
```

POOLAUTHENABLEFAILEDUNAVAILABLE

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_UNAVAILABLE(host, message)
```

POOLAUTHENABLEFAILEDWRONGCREDENTIALS

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_WRONG_CREDENTIALS(host, message)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGEXTERNALAUTHMISMATCH

Cannot join pool whose external authentication configuration is different.

Signature:

```
POOL_JOINING_EXTERNAL_AUTH_MISMATCH()
```

POOLJOININGHOSTCACERTIFICATESCONFLICT

The host joining the pool has different CA certificates from the pool coordinator while using the same name, uninstall them and try again.

Signature:

```
POOL_JOINING_HOST_CA_CERTIFICATES_CONFLICT()
```

POOLJOININGHOSTHASBONDS

The host joining the pool must not have any bonds.

Signature:

```
POOL_JOINING_HOST_HAS_BONDS()
```

POOLJOININGHOSTHASNETWORKSRIOVS

The host joining the pool must not have any network SRIOVs.

Signature:

```
POOL_JOINING_HOST_HAS_NETWORK_SRIOVS()
```

POOLJOININGHOSTHASNONMANAGEMENTVLANS

The host joining the pool must not have any nonmanagement vlans.

Signature:

```
POOL_JOINING_HOST_HAS_NON_MANAGEMENT_VLANS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGHOSTHASTUNNELS

The host joining the pool must not have any tunnels.

Signature:

```
POOL_JOINING_HOST_HAS_TUNNELS()
```

POOLJOININGHOSTMANAGEMENTVLANDOESNOTMATCH

The host joining the pool must have the same management vlan.

Signature:

```
POOL_JOINING_HOST_MANAGEMENT_VLAN_DOES_NOT_MATCH(local, remote)
```

POOLJOININGHOSTMUSTHAVEONLYONEIPONCLUSTERINGNETWORK

The host joining the pool must have one and only one IP on the clustering network

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_ONLY_ONE_IP_ON_CLUSTERING_NETWORK()
```

POOLJOININGHOSTMUSTHAVEPHYSICALMANAGEMENTNIC

The server joining the pool must have a physical management NIC i.e. the management NIC must not be on a VLAN or bonded PIF.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_PHYSICAL_MANAGEMENT_NIC()
```

POOLJOININGHOSTMUSTHAVESAMEAPIVERSION

The host joining the pool must have the same API version as the pool coordinator.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_SAME_API_VERSION(host_api_version,
master_api_version)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

POOLJOININGHOSTMUSTHAVESAMEDBSCHEMA

The host joining the pool must have the same database schema as the pool coordinator.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_SAME_DB_SCHEMA(host_db_schema,
master_db_schema)
```

POOLJOININGHOSTMUSTHAVESAMEPRODUCTVERSION

The server joining the pool must have the same product version as the pool coordinator.

Signature:

```
POOL_JOINING_HOST_MUST_HAVE_SAME_PRODUCT_VERSION()
```

POOLJOININGHOSTMUSTONLYHAVEPHYSICALPIFS

The host joining the pool must not have any bonds, VLANs or tunnels.

Signature:

```
POOL_JOINING_HOST_MUST_ONLY_HAVE_PHYSICAL_PIFS()
```

POOLJOININGHOSTTLSVERIFICATIONMISMATCH

The TLS verification check failed when the joining host attempted to open a verified connection to the pool coordinator using the imported pool certificate bundle.

Signature:

```
POOL_JOINING_HOST_TLS_VERIFICATION_MISMATCH()
```

POOLJOININGHOSTTRUSTEDCERTIFICATESCONFLICT

The joining host has a trusted certificate identical to one on the pool coordinator but with different purpose. Uninstall it then install it on the host again with the poolcompatible purpose, and try again.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
POOL_JOINING_HOST_TRUSTED_CERTIFICATES_CONFLICT(ref_in_pool, ref_on_host)
```

POOLJOININGMASTERCERTIFICATENOTINPOOLBUNDLE

The pool coordinator's own certificate is absent from the pool certificate bundle sent to the joining host. Run 'xe poolcertificatesync' on the coordinator and retry.

Signature:

```
POOL_JOINING_MASTER_CERTIFICATE_NOT_IN_POOL_BUNDLE(master_uuid)
```

POOLJOININGPOOLBUNDLEEMPTYAFTERIMPORT

The pool certificate bundle is empty or missing after import on the joining host. The bundle generation script updatecabundle.sh likely failed silently.

Signature:

```
POOL_JOINING_POOL_BUNDLE_EMPTY_AFTER_IMPORT(bundle_path)
```

POOLJOININGPOOLCANNOTENABLECLUSTERINGONVLANNETWORK

The remote pool cannot enable clustering on vlan network

Signature:

```
POOL_JOINING_POOL_CANNOT_ENABLE_CLUSTERING_ON_VLAN_NETWORK(vlan)
```

POOLJOININGSMFEATURESINCOMPATIBLE

The host joining the pool has an incompatible set of sm features from the pool coordinator. Make sure the sm are of the same versions and try again.

Signature:

```
POOL_JOINING_SM_FEATURES_INCOMPATIBLE(pool_sm_ref, candidate_sm_ref)
```

PROVISIONFAILEDOUTOFSPACE

The provision call failed because it ran out of space.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Signature:

```
PROVISION_FAILED_OUT_OF_SPACE()
```

PROVISIONONLYALLOWEDONTEMPLATE

The provision call can only be invoked on templates, not regular VMs.

Signature:

```
PROVISION_ONLY_ALLOWED_ON_TEMPLATE()
```

PUSBVDICONFLICT

The VDI corresponding to this PUSB has existing VBDs.

Signature:

```
PUSB_VDI_CONFLICT(PUSB, VDI)
```

PVSCACHESTORAGEALREADYPRESENT

The PVS site already has cache storage configured for the host.

Signature:

```
PVS_CACHE_STORAGE_ALREADY_PRESENT(site, host)
```

PVSCACHESTORAGEISINUSE

The PVS cache storage is in use by the site and cannot be removed.

Signature:

```
PVS_CACHE_STORAGE_IS_IN_USE(PVS_cache_storage)
```

PVSPROXYALREADYPRESENT

The VIF is already associated with a PVS proxy

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_PROXY_ALREADY_PRESENT(proxies)
```

PVSPROXYPRESENTONHIGHERVIFDEVICE

The VM has a VIF, with a higher device number than the new VIF, that uses a PVS proxy. The VIF used by PVS proxy must be the one with the lowest device number.

Signature:

```
PVS_PROXY_PRESENT_ON_HIGHER_VIF_DEVICE(device)
```

PVSSERVERADDRESSINUSE

The address specified is already in use by an existing PVSserver object

Signature:

```
PVS_SERVER_ADDRESS_IN_USE(address)
```

PVSSITECONTAINSRUNNINGPROXIES

The PVS site contains running proxies.

Signature:

```
PVS_SITE_CONTAINS_RUNNING_PROXIES(proxies)
```

PVSSITECONTAINSSERVERS

The PVS site contains servers and cannot be forgotten.

Signature:

```
PVS_SITE_CONTAINS_SERVERS(servers)
```

PVSVIFMUSTBEFIRSTDEVICE

The VIF used by PVS proxy must be the one with the lowest device number

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
PVS_VIF_MUST_BE_FIRST_DEVICE()
```

RBACPERMISSIONDENIED

RBAC permission denied.

Signature:

```
RBAC_PERMISSION_DENIED(permission, message)
```

REDOLOGISENABLED

The operation could not be performed because a redo log is enabled on the Pool.

Signature:

```
REDO_LOG_IS_ENABLED()
```

REPOSITORYALREADYEXISTS

The repository already exists.

Signature:

```
REPOSITORY_ALREADY_EXISTS(ref)
```

REPOSITORYCLEANUPFAILED

Failed to clean up local repository on coordinator.

Signature:

```
REPOSITORY_CLEANUP_FAILED()
```

REPOSITORYISINUSE

The repository is in use.

Signature:

```
REPOSITORY_IS_IN_USE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

REPOSYNCFAILED

Syncing with remote YUM repository failed.

Signature:

```
REPOSYNC_FAILED()
```

REPOSHOULDBESINGLEONEENABLED

If the bundle repository or remotepool repository is enabled, it should be the only enabled repository of the pool.

Signature:

```
REPO_SHOULD_BE_SINGLE_ONE_ENABLED()
```

REQUIREDPIFISUNPLUGGED

The operation you requested cannot be performed because the specified PIF is currently unplugged.

Signature:

```
REQUIRED_PIF_IS_UNPLUGGED(PIF)
```

RESTOREINCOMPATIBLEVERSION

The restore could not be performed because this backup has been created by a different incompatible product version

Signature:

```
RESTORE_INCOMPATIBLE_VERSION()
```

RESTORESCRIPTFAILED

The restore could not be performed because the restore script failed. Is the file corrupt?

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
RESTORE_SCRIPT_FAILED(log)
```

RESTORETARGETMGMTIFNOTINBACKUP

The restore could not be performed because the server's current management interface is not in the backup. The interfaces mentioned in the backup are:

Signature:

```
RESTORE_TARGET_MGMT_IF_NOT_IN_BACKUP()
```

RESTORETARGETMISSINGDEVICE

The restore could not be performed because a network interface is missing

Signature:

```
RESTORE_TARGET_MISSING_DEVICE(device)
```

ROLEALREADYEXISTS

Role already exists.

Signature:

```
ROLE_ALREADY_EXISTS()
```

ROLENOTFOUND

Role cannot be found.

Signature:

```
ROLE_NOT_FOUND()
```

SERVERCERTIFICATECHAININVALID

The provided intermediate certificates are not in a PEMencoded X509 format.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SERVER_CERTIFICATE_CHAIN_INVALID()
```

SERVERCERTIFICATEEXPIRED

The provided certificate has expired.

Signature:

```
SERVER_CERTIFICATE_EXPIRED(now, not_after)
```

SERVERCERTIFICATEINVALID

The provided certificate is not in a PEMencoded X509 format.

Signature:

```
SERVER_CERTIFICATE_INVALID()
```

SERVERCERTIFICATEKEYALGORITHMNOTSUPPORTED

The provided key uses an unsupported algorithm.

Signature:

```
SERVER_CERTIFICATE_KEY_ALGORITHM_NOT_SUPPORTED(algorithm_oid)
```

SERVERCERTIFICATEKEYINVALID

The provided key is not in a PEMencoded PKCS8 format.

Signature:

```
SERVER_CERTIFICATE_KEY_INVALID()
```

SERVERCERTIFICATEKEYMISMATCH

The provided key does not match the provided certificate's public key.

Signature:

```
SERVER_CERTIFICATE_KEY_MISMATCH()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SERVERCERTIFICATEKEYRSALENGTHNOTSUPPORTED

The provided RSA key does not have a length between 2048 and 4096.

Signature:

```
SERVER_CERTIFICATE_KEY_RSA_LENGTH_NOT_SUPPORTED(length)
```

SERVERCERTIFICATEKEYRSAMULTINOTSUPPORTED

The provided RSA key is using more than 2 primes, expecting only 2.

Signature:

```
SERVER_CERTIFICATE_KEY_RSA_MULTI_NOT_SUPPORTED()
```

SERVERCERTIFICATENOTVALIDYET

The provided certificate is not valid yet.

Signature:

```
SERVER_CERTIFICATE_NOT_VALID_YET(now, not_before)
```

SERVERCERTIFICATESIGNATURENOTSUPPORTED

The provided certificate is not using one of the following SHA2 signature algorithms: SHA256, SHA512.

Signature:

```
SERVER_CERTIFICATE_SIGNATURE_NOT_SUPPORTED()
```

SESSIONAUTHENTICATIONFAILED

The credentials given by the user are incorrect, so access has been denied, and you have not been issued a session handle.

Signature:

```
SESSION_AUTHENTICATION_FAILED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SESSIONAUTHORIZATIONFAILED

The credentials given by the user are correct, but the user could not be authorized, so access has been denied, and you have not been issued a session handle.

Signature:

```
SESSION_AUTHORIZATION_FAILED(username, msg)
```

SESSIONINVALID

You gave an invalid session reference. It may have been invalidated by a server restart, or timed out. You should get a new session handle, using one of the session.login calls. This error does not invalidate the current connection. The handle parameter echoes the bad value given.

Signature:

```
SESSION_INVALID(handle)
```

SESSIONNOTREGISTERED

This session is not registered to receive events. You must call event.register before event.next. The session handle you are using is echoed.

Signature:

```
SESSION_NOT_REGISTERED(handle)
```

SETCONSOLETIMEOUTPARTIALLYFAILED

Some hosts failed to set console timeout.

Signature:

```
SET_CONSOLE_TIMEOUT_PARTIALLY_FAILED(hosts)
```

SETSSHAUTOMODEPARTIALLYFAILED

Some hosts failed to set SSH auto mode.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SET_SSH_AUTO_MODE_PARTIALLY_FAILED(hosts)
```

SETSSHTIMEOUTPARTIALLYFAILED

Some hosts failed to set SSH timeout.

Signature:

```
SET_SSH_TIMEOUT_PARTIALLY_FAILED(hosts)
```

SLAVEREQUIRESMANAGEMENTINTERFACE

The management interface on a supporter cannot be disabled because the supporter would enter emergency mode.

Signature:

```
SLAVE_REQUIRES_MANAGEMENT_INTERFACE()
```

SMPLUGINCOMMUNICATIONFAILURE

The SM plugin did not respond to a query.

Signature:

```
SM_PLUGIN_COMMUNICATION_FAILURE(sm)
```

SRATTACHFAILED

Attaching this SR failed.

Signature:

```
SR_ATTACH_FAILED(sr)
```

SRBACKENDFAILURE

There was an SR backend failure.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR_BACKEND_FAILURE(status, stdout, stderr)
```

SRDEVICEINUSE

The SR operation cannot be performed because a device underlying the SR is in use by the server.

Signature:

```
SR_DEVICE_IN_USE()
```

SRDOESNOTSUPPORTMIGRATION

Cannot migrate a VDI to or from an SR that doesn't support migration.

Signature:

```
SR_DOES_NOT_SUPPORT_MIGRATION(sr, capabilities)
```

SRFULL

The SR is full. Requested new size exceeds the maximum size

Signature:

```
SR_FULL(requested, maximum)
```

SRHASMULTIPLEPBDS

The SR.shared flag cannot be set to false while the SR remains connected to multiple servers.

Signature:

```
SR_HAS_MULTIPLE_PBDS(PBD)
```

SRHASNOPBDS

The SR has no attached PBDs

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
SR_HAS_NO_PBDS(sr)
```

SRHASPBD

The SR is still connected to a host via a PBD. It cannot be destroyed or forgotten.

Signature:

```
SR_HAS_PBD(sr)
```

SRINDESTRUCTIBLE

The SR could not be destroyed because the 'indestructible' flag was set on it.

Signature:

```
SR_INDESTRUCTIBLE(sr)
```

SRISCACHESR

The SR is currently being used as a local cache SR.

Signature:

```
SR_IS_CACHE_SR(host)
```

SRNOTATTACHED

The SR is not attached.

Signature:

```
SR_NOT_ATTACHED(sr)
```

SRNOTEMPTY

The SR operation cannot be performed because the SR is not empty.

Signature:

```
SR_NOT_EMPTY()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SRNOTSHARABLE

The PBD could not be plugged because the SR is in use by another host and is not marked as sharable.

Signature:

```
SR_NOT_SHARABLE(sr, host)
```

SROPERATIONNOTSUPPORTED

The SR backend does not support the operation check the SR's allowed operations

Signature:

```
SR_OPERATION_NOT_SUPPORTED(sr)
```

SRREQUIRESUPGRADE

The operation cannot be performed until the SR has been upgraded

Signature:

```
SR_REQUIRES_UPGRADE(SR)
```

SRSOURCESPACEINSUFFICIENT

The source SR does not have sufficient temporary space available to proceed the operation.

Signature:

```
SR_SOURCE_SPACE_INSUFFICIENT(sr)
```

SRSUSPENDSPACEINSUFFICIENT

The suspend SR does not have sufficient free space to store the VM suspend image required to complete a snapshot with memory.

Signature:

```
SR_SUSPEND_SPACE_INSUFFICIENT(sr)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SRUNHEALTHY

The SR is currently unhealthy. See the suggestion on how to fix it.

Signature:

```
SR_UNHEALTHY(sr, health, fix)
```

SRUNKNOWNDRIVER

The SR could not be connected because the driver was not recognised.

Signature:

```
SR_UNKNOWN_DRIVER(driver)
```

SRUUIDEXISTS

An SR with that uuid already exists.

Signature:

```
SR_UUID_EXISTS(uuid)
```

SRVDILOCKINGFAILED

The operation could not proceed because necessary VDIs were already locked at the storage level.

Signature:

```
SR_VDI_LOCKING_FAILED()
```

SSLVERIFYERROR

The remote system's SSL certificate failed to verify against our certificate library.

Signature:

```
SSL_VERIFY_ERROR(reason)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SUBJECTALREADYEXISTS

Subject already exists.

Signature:

```
SUBJECT_ALREADY_EXISTS()
```

SUBJECTCANNOTBERESOLVED

Subject cannot be resolved by the external directory service.

Signature:

```
SUBJECT_CANNOT_BE_RESOLVED()
```

SUSPENDIMAGENOTACCESSIBLE

The suspend image of a checkpoint is not accessible from the host on which the VM is running

Signature:

```
SUSPEND_IMAGE_NOT_ACCESSIBLE(vdi)
```

SYNCBUNDLEINPROGRESS

The operation could not be performed because syncing bundle is in progress.

Signature:

```
SYNC_BUNDLE_IN_PROGRESS()
```

SYNCUPDATESINPROGRESS

The operation could not be performed because syncing updates is in progress.

Signature:

```
SYNC_UPDATES_IN_PROGRESS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

SYSPREP

VM.sysprep error with details in the message

Signature:

```
SYSPREP(vm, message)
```

SYSTEMSTATUSMUSTUSETARONOEM

You must use tar output to retrieve system status from an OEM server.

Signature:

```
SYSTEM_STATUS_MUST_USE_TAR_ON_OEM()
```

SYSTEMSTATUSRETRIEVALFAILED

Retrieving system status from the host failed. A diagnostic reason suitable for support organisations is also returned.

Signature:

```
SYSTEM_STATUS_RETRIEVAL_FAILED(reason)
```

TASKCANCELLED

The request was asynchronously canceled.

Signature:

```
TASK_CANCELLED(task)
```

TELEMETRYNEXTCOLLECTIONTOOLATE

The next scheduled telemetry data collection is too far into the future. Pick a timestamp within two telemetry intervals starting from now.

Signature:

```
TELEMETRY_NEXT_COLLECTION_TOO_LATE(timestamp)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TLSCONNECTIONFAILED

Cannot contact the other host using TLS on the specified address and port

Signature:

```
TLS_CONNECTION_FAILED(address, port)
```

TLSVERIFICATIONNOTENABLEDINPOOL

TLS verification has not been enabled in the pool successfully, please enable it in XC or run xe poolenabletlsverification instead.

Signature:

```
TLS_VERIFICATION_NOT_ENABLED_IN_POOL()
```

TOOBUSY

The request was rejected because the server is too busy.

Signature:

```
TOO_BUSY()
```

TOOMANYGROUPS

VM can only belong to one group.

Signature:

```
TOO_MANY_GROUPS()
```

TOOMANYPENDINGTASKS

The request was rejected because there are too many pending tasks on the server.

Signature:

```
TOO_MANY_PENDING_TASKS()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TOOMANYSTORAGEMIGRATES

You reached the maximal number of concurrently migrating VMs.

Signature:

```
TOO_MANY_STORAGE_MIGRATES(number)
```

TOOMANYVUSBS

The VM has too many VUSBs.

Signature:

```
TOO_MANY_VUSBS(number)
```

TRANSPORTPIFNOTCONFIGURED

The tunnel transport PIF has no IP configuration set.

Signature:

```
TRANSPORT_PIF_NOT_CONFIGURED(PIF)
```

TRUSTEDCERTIFICATEALREADYEXISTS

A trusted certificate already exists with the same purpose.

Signature:

```
TRUSTED_CERTIFICATE_ALREADY_EXISTS(fingerprint)
```

TRUSTEDCERTIFICATEEXPIRED

The provided certificate has expired.

Signature:

```
TRUSTED_CERTIFICATE_EXPIRED(now, not_after)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

TRUSTEDCERTIFICATEINVALID

The provided certificate is not in a PEMencoded X509 format.

Signature:

```
TRUSTED_CERTIFICATE_INVALID()
```

TRUSTEDCERTIFICATENOTVALIDYET

The provided certificate is not valid yet.

Signature:

```
TRUSTED_CERTIFICATE_NOT_VALID_YET(now, not_before)
```

UNIMPLEMENTEDINSMBACKEND

You have attempted a function which is not implemented

Signature:

```
UNIMPLEMENTED_IN_SM_BACKEND(message)
```

UNKNOWNBOOTLOADER

The requested bootloader is unknown

Signature:

```
UNKNOWN_BOOTLOADER(vm, bootloader)
```

UPDATEINFOHASHMISMATCH

The hash of updateinfo doesn't match with current one. There may be newer available updates.

Signature:

```
UPDATEINFO_HASH_MISMATCH()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATEALREADYAPPLIED

This update has already been applied.

Signature:

```
UPDATE_ALREADY_APPLIED(update)
```

UPDATEALREADYAPPLIEDINPOOL

This update has already been applied to all hosts in the pool.

Signature:

```
UPDATE_ALREADY_APPLIED_IN_POOL(update)
```

UPDATEALREADYEXISTS

The uploaded update already exists

Signature:

```
UPDATE_ALREADY_EXISTS(uuid)
```

UPDATEAPPLYFAILED

The update failed to apply. Please see attached output.

Signature:

```
UPDATE_APPLY_FAILED(output)
```

UPDATEISAPPLIED

The specified update has been applied and cannot be destroyed.

Signature:

```
UPDATE_IS_APPLIED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATEPOOLAPPLYFAILED

The update cannot be applied for the following hosts.

Signature:

```
UPDATE_POOL_APPLY_FAILED(hosts)
```

UPDATEPRECHECKFAILEDCONFLICTPRESENT

The update precheck stage failed: conflicting updates are present.

Signature:

```
UPDATE_PRECHECK_FAILED_CONFLICT_PRESENT(update, conflict_update)
```

UPDATEPRECHECKFAILEDGPGKEYNOTIMPORTED

The update precheck stage failed: RPM package validation requires a GPG key that is not present on the host.

Signature:

```
UPDATE_PRECHECK_FAILED_GPGKEY_NOT_IMPORTED(update)
```

UPDATEPRECHECKFAILEDOUTOFSPACE

The update precheck stage failed: the server does not have enough space.

Signature:

```
UPDATE_PRECHECK_FAILED_OUT_OF_SPACE(update, available_space,
required_space )
```

UPDATEPRECHECKFAILEDPREREQUISITEMISSING

The update precheck stage failed: prerequisite updates are missing.

Signature:

```
UPDATE_PRECHECK_FAILED_PREREQUISITE_MISSING(update, prerequisite_update)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UPDATEPRECHECKFAILEDUNKNOWNERROR

The update precheck stage failed with an unknown error.

Signature:

```
UPDATE_PRECHECK_FAILED_UNKNOWN_ERROR(update, info)
```

UPDATEPRECHECKFAILEDWRONGSERVERVERSION

The update precheck stage failed: the server is of an incorrect version.

Signature:

```
UPDATE_PRECHECK_FAILED_WRONG_SERVER_VERSION(update, installed_version,
required_version )
```

UPDATESYNCINGREMOTEPOOLCOORDINATORCONNECTIONFAILED

There was an error connecting to the remote pool coordinator while syncing updates from it.

Signature:

```
UPDATE_SYNCING_REMOTE_POOL_COORDINATOR_CONNECTION_FAILED()
```

UPDATESYNCINGREMOTEPOOLCOORDINATORSERVICEFAILED

There was an error connecting to the server while syncing updates from it. The service contacted didn't reply properly.

Signature:

```
UPDATE_SYNCING_REMOTE_POOL_COORDINATOR_SERVICE_FAILED()
```

USBALREADYATTACHED

The USB device is currently attached to a VM.

Signature:

```
USB_ALREADY_ATTACHED(PUSB, VM)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

USBGROUPCONFLICT

USBgroups are currently restricted to contain no more than one VUSB.

Signature:

```
USB_GROUP_CONFLICT(USB_group)
```

USBGROUPCONTAINSNOPUSBS

The USB group does not contain any PUSBs.

Signature:

```
USB_GROUP_CONTAINS_NO_PUSBS(usb_group)
```

USBGROUPCONTAINSPUSB

The USB group contains active PUSBs and cannot be deleted.

Signature:

```
USB_GROUP_CONTAINS_PUSB(pusbs)
```

USBGROUPCONTAINSVUSB

The USB group contains active VUSBs and cannot be deleted.

Signature:

```
USB_GROUP_CONTAINS_VUSB(vusbs)
```

USERISNOTLOCALSUPERUSER

Only the local superuser can perform this operation.

Signature:

```
USER_IS_NOT_LOCAL_SUPERUSER(msg)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

UUIDINVALID

The uuid you supplied was invalid.

Signature:

```
UUID_INVALID(type, uuid)
```

V6DFAILURE

There was a problem with the license daemon v6d.

Signature:

```
V6D_FAILURE()
```

VALUENOTSUPPORTED

You attempted to set a value that is not supported by this implementation. The fullyqualified field name and the value that you tried to set are returned. Also returned is a developeronly diagnostic reason.

Signature:

```
VALUE_NOT_SUPPORTED(field, value, reason)
```

VBDCDSMUSTBEREADONLY

Read/write CDs are not supported

Signature:

```
VBD_CDS_MUST_BE_READONLY()
```

VBDISEMPTY

Operation could not be performed because the drive is empty

Signature:

```
VBD_IS_EMPTY(vbd)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VBDNOTEMPTY

Operation could not be performed because the drive is not empty

Signature:

```
VBD_NOT_EMPTY(vbd)
```

VBDNOTREMOVABLEMEDIA

Media could not be ejected because it is not removable

Signature:

```
VBD_NOT_REMOVABLE_MEDIA(vbd)
```

VBDNOTUNPLUGGABLE

Drive could not be hotunplugged because it is not marked as unpluggable

Signature:

```
VBD_NOT_UNPLUGGABLE(vbd)
```

VBDTRAYLOCKED

This VM has locked the DVD drive tray, so the disk cannot be ejected

Signature:

```
VBD_TRAY_LOCKED(vbd)
```

VCPUMAXNOTCORESPERSOCKETMULTIPLE

VCPUsmax must be a multiple of corespersocket

Signature:

```
VCPU_MAX_NOT_CORES_PER_SOCKET_MULTIPLE(vcpu_max, cores_per_socket)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDICBTENABLED

The requested operation is not allowed for VDIs with CBT enabled or VMs having such VDIs, and CBT is enabled for the specified VDI.

Signature:

```
VDI_CBT_ENABLED(vdi)
```

VDICONTAINSMETADATAOFTHISPOOL

The VDI could not be opened for metadata recovery as it contains the current pool's metadata.

Signature:

```
VDI_CONTAINS_METADATA_OF_THIS_POOL(vdi, pool)
```

VDICOPYFAILED

The VDI copy action has failed

Signature:

```
VDI_COPY_FAILED()
```

VDIHASRRDS

The operation cannot be performed because this VDI has rrd stats

Signature:

```
VDI_HAS_RRDS(vdi)
```

VDIINCOMPATIBLETYPE

This operation cannot be performed because the specified VDI is of an incompatible type eg: an HA statefile cannot be attached to a guest

Signature:

```
VDI_INCOMPATIBLE_TYPE(vdi, type)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDIINUSE

This operation cannot be performed because this VDI is in use by some other operation

Signature:

```
VDI_IN_USE(vdi, operation)
```

VDIISAPHYSICALDEVICE

The operation cannot be performed on physical device

Signature:

```
VDI_IS_A_PHYSICAL_DEVICE(vdi)
```

VDIISENCRYPTED

The requested operation is not allowed because the specified VDI is encrypted.

Signature:

```
VDI_IS_ENCRYPTED(vdi)
```

VDIISNOTISO

This operation can only be performed on CD VDIs iso files or CDROM drives

Signature:

```
VDI_IS_NOT_ISO(vdi, type)
```

VDILOCATIONMISSING

This operation cannot be performed because the specified VDI could not be found in the specified SR

Signature:

```
VDI_LOCATION_MISSING(sr, location)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDIMISSING

This operation cannot be performed because the specified VDI could not be found on the storage substrate

Signature:

```
VDI_MISSING(sr, vdi)
```

VDINEEDSVMFORMIGRATE

Cannot migrate a VDI which is not attached to a running VM.

Signature:

```
VDI_NEEDS_VM_FOR_MIGRATE(vdi)
```

VDINOTAVAILABLE

This operation cannot be performed because this VDI could not be properly attached to the VM.

Signature:

```
VDI_NOT_AVAILABLE(vdi)
```

VDINOTINMAP

This VDI was not mapped to a destination SR in VM.migratesend operation

Signature:

```
VDI_NOT_IN_MAP(vdi)
```

VDINOTMANAGED

This operation cannot be performed because the system does not manage this VDI

Signature:

```
VDI_NOT_MANAGED(vdi)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDINOTSPARSE

The VDI is not stored using a sparse format. It is not possible to query and manipulate only the changed blocks or 'block differences' or 'disk deltas' between two VDIs. Please select a VDI which uses a sparseaware technology such as VHD.

Signature:

```
VDI_NOT_SPARSE(vdi)
```

VDINOCBTMETADATA

The requested operation is not allowed because the specified VDI does not have changed block tracking metadata.

Signature:

```
VDI_NO_CBT_METADATA(vdi)
```

VDIONBOOTMODEINCOMPATIBLEWITHOPERATION

This operation is not permitted on VDIs in the 'onboot=reset' mode, or on VMs having such VDIs.

Signature:

```
VDI_ON_BOOT_MODE_INCOMPATIBLE_WITH_OPERATION()
```

VDIREADONLY

The operation required write access but this VDI is readonly

Signature:

```
VDI_READONLY(vdi)
```

VDITOOLARGE

The VDI is too large.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VDI_TOO_LARGE(vdi, maximum size)
```

VDITOOSMALL

The VDI is too small. Please resize it to at least the minimum size.

Signature:

```
VDI_TOO_SMALL(vdi, minimum size)
```

VGPUDESTINATIONINCOMPATIBLE

The VGPU is not compatible with any PGPU in the destination.

Signature:

```
VGPU_DESTINATION_INCOMPATIBLE(reason, vgpu, host)
```

VGPUGUESTDRIVERLIMIT

The guest driver does not support VGPU migration.

Signature:

```
VGPU_GUEST_DRIVER_LIMIT(reason, vm, host)
```

VGPUSUSPENSIONNOTSUPPORTED

The VGPU configuration does not support suspension.

Signature:

```
VGPU_SUSPENSION_NOT_SUPPORTED(reason, vgpu, host)
```

VGPUTYPENOTCOMPATIBLE

Cannot create a virtual GPU that is incompatible with the existing types on the VM.

Signature:

```
VGPU_TYPE_NOT_COMPATIBLE(type)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VGPUTYPENOTCOMPATIBLEWITHRUNNINGTYPE

The VGPU type is incompatible with one or more of the VGPU types currently running on this PGPU

Signature:

```
VGPU_TYPE_NOT_COMPATIBLE_WITH_RUNNING_TYPE(pgpu, type, running_type)
```

VGPUTYPENOTENABLED

VGPU type is not one of the PGPU's enabled types.

Signature:

```
VGPU_TYPE_NOT_ENABLED(type, enabled_types)
```

VGPUTYPENOTSUPPORTED

VGPU type is not one of the PGPU's supported types.

Signature:

```
VGPU_TYPE_NOT_SUPPORTED(type, supported_types)
```

VGPUTYPENOLONGERSUPPORTED

VGPU type is no longer supported

Signature:

```
VGPU_TYPE_NO_LONGER_SUPPORTED(type)
```

VIFINUSE

Network has active VIFs

Signature:

```
VIF_IN_USE(network, VIF)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VIFNOTINMAP

This VIF was not mapped to a destination Network in VM.migratesend operation

Signature:

```
VIF_NOT_IN_MAP(vif)
```

VLANINUSE

Operation cannot be performed because this VLAN is already in use. Please check your network configuration.

Signature:

```
VLAN_IN_USE(device, vlan)
```

VLANTAGINVALID

You tried to create a VLAN, but the tag you gave was invalid it must be between 0 and 4094. The parameter echoes the VLAN tag you gave.

Signature:

```
VLAN_TAG_INVALID(VLAN)
```

VMPPARCHIVEMOREFREQUENTTHANBACKUP

Archive more frequent than backup.

Signature:

```
VMPP_ARCHIVE_MORE_FREQUENT_THAN_BACKUP()
```

VMPPHASVM

There is at least one VM assigned to this protection policy.

Signature:

```
VMPP_HAS_VM()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMSSHASVM

There is at least one VM assigned to snapshot schedule.

Signature:

```
VMSS_HAS_VM()
```

VMSFAILEDTOCOOPERATE

The given VMs failed to release memory when instructed to do so

Signature:

```
VMS_FAILED_TO_COOPERATE()
```

VMASSIGNEDTOPROTECTIONPOLICY

This VM is assigned to a protection policy.

Signature:

```
VM_ASSIGNED_TO_PROTECTION_POLICY(vm, vmpp)
```

VMASSIGNEDTOSNAPSHOTSCHEDULE

This VM is assigned to a snapshot schedule.

Signature:

```
VM_ASSIGNED_TO_SNAPSHOT_SCHEDULE(vm, vmss)
```

VMATTACHEDTOMORETHANONEVDIWITHTIMEOFFSETMARKEDASRESETONBOOT

You attempted to start a VM that's attached to more than one VDI with a timeoffset marked as resetonboot.

Signature:

```
VM_ATTACHED_TO_MORE_THAN_ONE_VDI_WITH_TIMEOFFSET_MARKED_AS_RESET_ON_BOOT(v
m)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMBADPOWERSTATE

You attempted an operation on a VM that was not in an appropriate power state at the time; for example, you attempted to start a VM that was already running. The parameters returned are the VM's handle, and the expected and actual VM state at the time of the call.

Signature:

```
VM_BAD_POWER_STATE(vm, expected, actual)
```

VMBIOSSTRINGSALREADYSET

The BIOS strings for this VM have already been set and cannot be changed.

Signature:

```
VM_BIOS_STRINGS_ALREADY_SET()
```

VMCALLPLUGINRATELIMIT

There is a minimal interval required between consecutive plugin calls made on the same VM, please wait before retry.

Signature:

```
VM_CALL_PLUGIN_RATE_LIMIT(VM, interval, wait)
```

VMCANNOTDELETEDEFAULTTEMPLATE

You cannot delete the specified default template.

Signature:

```
VM_CANNOT_DELETE_DEFAULT_TEMPLATE(vm)
```

VMCHECKPOINTRESUMEFAILED

An error occured while restoring the memory image of the specified virtual machine

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_CHECKPOINT_RESUME_FAILED(vm)
```

VMCHECKPOINTSUSPENDFAILED

An error occured while saving the memory image of the specified virtual machine

Signature:

```
VM_CHECKPOINT_SUSPEND_FAILED(vm)
```

VMCRASHED

The VM crashed

Signature:

```
VM_CRASHED(vm)
```

VMDUPLICATEVBDDEVICE

The specified VM has a duplicate VBD device and cannot be started.

Signature:

```
VM_DUPLICATE_VBD_DEVICE(vm, vbd, device)
```

VMFAILEDSHUTDOWNACKNOWLEDGMENT

VM didn't acknowledge the need to shutdown.

Signature:

```
VM_FAILED_SHUTDOWN_ACKNOWLEDGMENT(vm)
```

VMFAILEDSUSPENDACKNOWLEDGMENT

VM didn't acknowledge the need to suspend.

Signature:

```
VM_FAILED_SUSPEND_ACKNOWLEDGMENT(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMHALTED

The VM unexpectedly halted

Signature:

```
VM_HALTED(vm)
```

VMHASCHECKPOINT

Cannot migrate a VM which has a checkpoint.

Signature:

```
VM_HAS_CHECKPOINT(vm)
```

VMHASNOSUSPENDVDI

VM cannot be resumed because it has no suspend VDI

Signature:

```
VM_HAS_NO_SUSPEND_VDI(vm)
```

VMHASPCIATTACHED

This operation could not be performed, because the VM has one or more PCI devices passed through.

Signature:

```
VM_HAS_PCI_ATTACHED(vm)
```

VMHASSRIOVVIF

This operation could not be performed, because the VM has one or more SRIOV VIFs.

Signature:

```
VM_HAS_SRIOV_VIF(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMHASTOOMANYSNAPSHOTS

Cannot migrate a VM with more than one snapshot.

Signature:

```
VM_HAS_TOO_MANY_SNAPSHOTS(vm)
```

VMHASVGPU

This operation could not be performed, because the VM has one or more virtual GPUs.

Signature:

```
VM_HAS_VGPU(vm)
```

VMHASVUSBS

The operation is not allowed when the VM has VUSBs.

Signature:

```
VM_HAS_VUSBS(VM)
```

VMHOSTINCOMPATIBLEVERSION

This VM operation cannot be performed on an olderversioned host during an upgrade.

Signature:

```
VM_HOST_INCOMPATIBLE_VERSION(host, vm)
```

VMHOSTINCOMPATIBLEVERSIONMIGRATE

Cannot migrate a VM to a destination host which is older than the source host.

Signature:

```
VM_HOST_INCOMPATIBLE_VERSION_MIGRATE(host, vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMHOSTINCOMPATIBLEVIRTUALHARDWAREPLATFORMVERSION

You attempted to run a VM on a host that cannot provide the VM's required Virtual Hardware Platform version.

Signature:

```
VM_HOST_INCOMPATIBLE_VIRTUAL_HARDWARE_PLATFORM_VERSION(host,
host_versions, vm, vm_version)
```

VMHVMREQUIRED

HVM is required for this operation

Signature:

```
VM_HVM_REQUIRED(vm)
```

VMINCOMPATIBLEWITHTHISHOST

The VM is incompatible with the CPU features of this host.

Signature:

```
VM_INCOMPATIBLE_WITH_THIS_HOST(vm, host, reason)
```

VMISIMMOBILE

The VM is configured in a way that prevents it from being mobile.

Signature:

```
VM_IS_IMMOBILE(VM)
```

VMISPARTOFANAPPLIANCE

This operation is not allowed as the VM is part of an appliance.

Signature:

```
VM_IS_PART_OF_AN_APPLIANCE(vm, appliance)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMISPROTECTED

This operation cannot be performed because the specified VM is protected by HA

Signature:

```
VM_IS_PROTECTED(vm)
```

VMISTEMPLATE

The operation attempted is not valid for a template VM

Signature:

```
VM_IS_TEMPLATE(vm)
```

VMISUSINGNESTEDVIRT

This operation is illegal because the VM is using nested virtualization.

Signature:

```
VM_IS_USING_NESTED_VIRT(VM)
```

VMLACKSFEATURE

You attempted an operation on a VM which lacks the feature.

Signature:

```
VM_LACKS_FEATURE(vm)
```

VMMEMORYSIZETOOLOW

The specified VM has too little memory to be started.

Signature:

```
VM_MEMORY_SIZE_TOO_LOW(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMMIGRATECONTACTREMOTESERVICEFAILED

Failed to contact service on the destination host.

Signature:

```
VM_MIGRATE_CONTACT_REMOTE_SERVICE_FAILED()
```

VMMIGRATEFAILED

An error occurred during the migration process.

Signature:

```
VM_MIGRATE_FAILED(vm, source, destination, msg)
```

VMMISSINGPVDRIVERS

You attempted an operation on a VM which requires PV drivers to be installed but the drivers were not detected.

Signature:

```
VM_MISSING_PV_DRIVERS(vm)
```

VMNONSUSPENDABLE

You attempted an operation on a VM which is not suspendable.

Signature:

```
VM_NON_SUSPENDABLE(vm, reason)
```

VMNOTRESIDENTHERE

The specified VM is not currently resident on the specified server.

Signature:

```
VM_NOT_RESIDENT_HERE(vm, host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMNOCRASHDUMPSR

This VM does not have a crash dump SR specified.

Signature:

```
VM_NO_CRASHDUMP_SR(vm)
```

VMNOEMPTYCDVBD

The VM has no empty CD drive VBD.

Signature:

```
VM_NO_EMPTY_CD_VBD(vm)
```

VMNOSUSPENDSR

This VM does not have a suspend SR specified.

Signature:

```
VM_NO_SUSPEND_SR(vm)
```

VMNOVCPUS

You need at least 1 VCPU to start a VM

Signature:

```
VM_NO_VCPUS(vm)
```

VMOLDPVDRIVERS

You attempted an operation on a VM which requires a more recent version of the PV drivers. Please upgrade your PV drivers.

Signature:

```
VM_OLD_PV_DRIVERS(vm, major, minor)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMPCIBUSFULL

The VM does not have any free PCI slots

Signature:

```
VM_PCI_BUS_FULL(VM)
```

VMPVDRIVERSINUSE

VM PV drivers still in use

Signature:

```
VM_PV_DRIVERS_IN_USE(vm)
```

VMREBOOTED

The VM unexpectedly rebooted

Signature:

```
VM_REBOOTED(vm)
```

VMREQUIRESGPU

You attempted to run a VM on a host which doesn't have a pGPU available in the GPU group needed by the VM. The VM has a vGPU attached to this GPU group.

Signature:

```
VM_REQUIRES_GPU(vm, GPU_group)
```

VMREQUIRESIOMMU

You attempted to run a VM on a host which doesn't have I/O virtualization IOMMU/VTd enabled, which is needed by the VM.

Signature:

```
VM_REQUIRES_IOMMU(host)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMREQUIRESNETWORK

You attempted to run a VM on a host which doesn't have a PIF on a Network needed by the VM. The VM has at least one VIF attached to the Network.

Signature:

```
VM_REQUIRES_NETWORK(vm, network)
```

VMREQUIRESSR

You attempted to run a VM on a host which doesn't have access to an SR needed by the VM. The VM has at least one VBD attached to a VDI in the SR.

Signature:

```
VM_REQUIRES_SR(vm, sr)
```

VMREQUIRESVDI

VM cannot be started because it requires a VDI which cannot be attached

Signature:

```
VM_REQUIRES_VDI(vm, vdi)
```

VMREQUIRESVGPU

You attempted to run a VM on a host on which the vGPU required by the VM cannot be allocated on any pGPUs in the GPUgroup needed by the VM.

Signature:

```
VM_REQUIRES_VGPU(vm, GPU_group, vGPU_type)
```

VMREQUIRESVUSB

You attempted to run a VM on a host on which the VUSB required by the VM cannot be allocated on any PUSBs in the USBgroup needed by the VM.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_REQUIRES_VUSB(vm, USB_group)
```

VMREVERTFAILED

An error occured while reverting the specified virtual machine to the specified snapshot

Signature:

```
VM_REVERT_FAILED(vm, snapshot)
```

VMSHUTDOWNTIMEOUT

VM failed to shutdown before the timeout expired

Signature:

```
VM_SHUTDOWN_TIMEOUT(vm, timeout)
```

VMSNAPSHOTWITHQUIESCEFAILED

The quiescedsnapshot operation failed for an unexpected reason

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_FAILED(vm)
```

VMSNAPSHOTWITHQUIESCENOTSUPPORTED

The VSS plugin is not installed on this virtual machine

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_NOT_SUPPORTED(vm, error)
```

VMSNAPSHOTWITHQUIESCEPLUGINDEOSNOTRESPOND

The VSS plugin cannot be contacted

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_PLUGIN_DEOS_NOT_RESPOND(vm)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VMSNAPSHOTWITHQUIESCETIMEOUT

The VSS plugin has timed out

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_TIMEOUT(vm)
```

VMSUSPENDTIMEOUT

VM failed to suspend before the timeout expired

Signature:

```
VM_SUSPEND_TIMEOUT(vm, timeout)
```

VMTOOMANYVCPUS

Too many VCPUs to start this VM

Signature:

```
VM_TOO_MANY_VCPUS(vm)
```

VMTOIMPORTISNOTNEWERVERSION

The VM cannot be imported unforced because it is either the same version or an older version of an existing VM.

Signature:

```
VM_TO_IMPORT_IS_NOT_NEWER_VERSION(vm, existing_version, version_to_import)
```

VMUNSAFEBOOT

You attempted an operation on a VM that was judged to be unsafe by the server. This can happen if the VM would run on a CPU that has a potentially incompatible set of feature flags to those the VM requires. If you want to override this warning then use the 'force' option.

Signature:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
VM_UNSAFE_BOOT(vm)
```

VTPMMAXAMOUNTREACHED

The VM cannot be associated with more VTPMs.

Signature:

```
VTPM_MAX_AMOUNT_REACHED(amount)
```

WLBAUTHENTICATIONFAILED

Failed to authenticate with the WLB server, the provided credentials are invalid.

Signature:

```
WLB_AUTHENTICATION_FAILED()
```

WLBCONNECTIONREFUSED

WLB refused a connection to the server.

Signature:

```
WLB_CONNECTION_REFUSED()
```

WLBCONNECTIONRESET

The connection to the WLB server was reset.

Signature:

```
WLB_CONNECTION_RESET()
```

WLBDISABLED

This pool has wlbenabled set to false.

Signature:

```
WLB_DISABLED()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

WLBINTERNALERROR

WLB reported an internal error.

Signature:

```
WLB_INTERNAL_ERROR()
```

WLBMALFORMEDREQUEST

WLB rejected the server's request as malformed.

Signature:

```
WLB_MALFORMED_REQUEST()
```

WLBMALFORMEDRESPONSE

WLB said something that the server wasn't expecting or didn't understand. The method called on WLB, a diagnostic reason, and the response from WLB are returned.

Signature:

```
WLB_MALFORMED_RESPONSE(method, reason, response)
```

WLBNOTINITIALIZED

No WLB connection is configured.

Signature:

```
WLB_NOT_INITIALIZED()
```

WLBTIMEOUT

The communication with the WLB server timed out.

Signature:

```
WLB_TIMEOUT(configured_timeout)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

WLBUNKNOWNHOST

The configured WLB server name failed to resolve in DNS.

Signature:

```
WLB_UNKNOWN_HOST()
```

WLBURLINVALID

The WLB URL is invalid. Ensure it is in the format: <ipaddress>:<port>. The configured/given URL is returned.

Signature:

```
WLB_URL_INVALID(url)
```

WLBXENSERVERAUTHENTICATIONFAILED

WLB reported that the server rejected its configured authentication details.

Signature:

```
WLB_XENSERVER_AUTHENTICATION_FAILED()
```

WLBXENSERVERCONNECTIONREFUSED

WLB reported that the server refused to let it connect even though we're connecting perfectly fine in the other direction.

Signature:

```
WLB_XENSERVER_CONNECTION_REFUSED()
```

WLBXENSERVERMALFORMEDRESPONSE

WLB reported that the server said something to it that WLB wasn't expecting or didn't understand.

Signature:

```
WLB_XENSERVER_MALFORMED_RESPONSE()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

WLBXENSERVERTIMEOUT

WLB reported that communication with the server timed out.

Signature:

```
WLB_XENSERVER_TIMEOUT()
```

WLBXENSERVERUNKNOWNHOST

WLB reported that its configured server name for this server instance failed to resolve in DNS.

Signature:

```
WLB_XENSERVER_UNKNOWN_HOST()
```

XAPIHOOKFAILED

3rd party xapi hook failed

Signature:

```
XAPI_HOOK_FAILED(hook_name, reason, stdout, exit_code)
```

XENAPIMISSINGPLUGIN

The requested plugin could not be found.

Signature:

```
XENAPI_MISSING_PLUGIN(name)
```

XENAPIPLUGINFAILURE

There was a failure communicating with the plugin.

Signature:

```
XENAPI_PLUGIN_FAILURE(status, stdout, stderr)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XENINCOMPATIBLE

The current version of Xen or its control libraries is incompatible with the Toolstack.

Signature:

```
XEN_INCOMPATIBLE()
```

XENVSSREQERRORADDINGVOLUMETOSNAPSETFAILED

Some volumes to be snapshot could not be added to the VSS snapshot set

Signature:

```
XEN_VSS_REQ_ERROR_ADDING_VOLUME_TO_SNAPSET_FAILED(vm, error_code)
```

XENVSSREQERRORCREATINGSNAPSHOT

An attempt to create the snapshots failed

Signature:

```
XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT(vm, error_code)
```

XENVSSREQERRORCREATINGSNAPSHOTXMLSTRING

Could not create the XML string generated by the transportable snapshot

Signature:

```
XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT_XML_STRING(vm, error_code)
```

XENVSSREQERRORINITFAILED

Initialization of the VSS requester failed

Signature:

```
XEN_VSS_REQ_ERROR_INIT_FAILED(vm, error_code)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XENVSSREQERRORNOVOLUMESSUPPORTED

Could not find any volumes supported by the VSS Provider

Signature:

```
XEN_VSS_REQ_ERROR_NO_VOLUMES_SUPPORTED(vm, error_code)
```

XENVSSREQERRORPREPARINGWRITERS

An attempt to prepare VSS writers for the snapshot failed

Signature:

```
XEN_VSS_REQ_ERROR_PREPARING_WRITERS(vm, error_code)
```

XENVSSREQERRORPROVNOTLOADED

The VSS Provider is not loaded

Signature:

```
XEN_VSS_REQ_ERROR_PROV_NOT_LOADED(vm, error_code)
```

XENVSSREQERRORSTARTSNAPSHOTSETFAILED

An attempt to start a new VSS snapshot failed

Signature:

```
XEN_VSS_REQ_ERROR_START_SNAPSHOT_SET_FAILED(vm, error_code)
```

XMLRPCUNMARSHALFAILURE

The server failed to unmarshal the XMLRPC message; it was expecting one element and received something else.

Signature:

```
XMLRPC_UNMARSHAL_FAILURE(expected, received)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

XenServer® Changed Block Tracking Guide

Changed block tracking provides a set of features and APIs that enable you to develop fast and space-efficient incremental backup solutions for XenServer.

Changed block tracking is available only to customers with XenServer Premium Edition. If a customer without Premium Edition attempts to use an incremental backup solution for XenServer that uses changed block tracking, they are prevented from enabling changed block tracking on new VDIs. However, if the customer has existing VDIs with changed block tracking enabled, they can still perform other changed block tracking actions on these VDIs.

How changed block tracking works

When changed block tracking is enabled for a virtual disk image (VDI), any blocks that are changed in that VDI are recorded in a log file. Every time the VDI is snapshotted, this log file can be used to identify the blocks that have changed since the VDI was last snapshotted. This provides the capability to backup only those blocks that have changed.

After the changed blocks have been exported, the full VDI snapshots can now be changed into metadata-only snapshots by destroying the data associated with them and leaving only the changed block information. These metadata only snapshots are linked both to the preceding metadata-only snapshot and to the following metadata-only snapshot. This provides a chain of metadata that records the full history of changes to this VDI since changed block tracking was enabled.

The changed block tracking feature also takes advantage of network block device (NBD) capabilities to perform the export of data from the changed blocks.

Benefits of changed block tracking

Unlike some other incremental backup solutions, changed block tracking does not require that the customer keep a snapshot of the last known good state of a VDI available on the host or a storage repository (SR) to compare the current state to. The customer needs less disk space because,

© 1997–2026 Citrix Systems, Inc. All rights reserved.

instead of handling and storing large VDIs, with changed block tracking they instead can choose to store space-efficient metadata-only snapshot files.

Changed block tracking also saves the customer time as well as space. Other backup solutions export a snapshot of the whole VDI every single time the VDI is backed up. This is a time-consuming process and the customer has to pay that time cost every time they take a backup. With changed block tracking enabled, the first back up exports a snapshot of the whole VDI. However, subsequent backups only export the blocks in the VDI that have changed since the previous backup. This decreases the time required to export the backup in proportion to the percentage of blocks that have changed.

For example, it can take around 10 hours to export a backup of a full 1 TB VDI. If, after a week, 5% of the blocks in that VDI have changed, exporting the backup takes 5% of the time - 30 minutes. A backup taken after a day has even fewer changed blocks and takes even less time to export.

The savings in time and space that changed block tracking provides makes it a preferable backup solution for customers using XenServer. The simple API that XenServer exposes makes it easy for you to develop an incremental backup solution that delivers these benefits to the end user. You can use this API through the language-agnostic remote procedure calls (RPCs) or take advantage of the language bindings provided for C, C#, Java, Python 3, and PowerShell.

Getting started using changed block tracking

This section steps through the process of using changed block tracking to create incremental backups.

Before getting started with changed block tracking, we recommend that you read the XenServer Software Developer Kit Guide. This document contains information to help you become familiar with developing for XenServer.

The examples provided in these steps use the Python binding for the Management API.

- For more information about the individual RPC calls, see the XenServer Management API
- For more detailed information about individual steps in this process, see the following chapters.

Full Python examples are provided on GitHub.

The NBD connection examples provided in these steps use the Linux nbd-client. However, you can use any NBD client that supports the "fixed newstyle" version of the NBD protocol. For more information, see the NBD protocol documentation.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTE:

If you are using the Linux upstream NBD client, a minimum version of 3.15 is required to support TLS.

Prerequisites

Before you start, set up or implement an NBD client at the backup location that supports the "fixed newstyle" version of the NBD protocol. For more information, see Exporting the changed blocks using an NBD client.

Enable NBD connections on your network. For more information, see Enabling NBD connections on XenServer.

Procedure

This procedure is broken down into three sections:

- **Setting up changed block tracking** Perform the steps in this section once, when you start using changed block tracking, to enable the changed block tracking capability and export a base snapshot that the incremental, changed block exported data is compared to.
- **Taking incremental backups** Perform the steps in this section every time you want to take an incremental back up of the changed blocks in a VDI.
- **Restoring a VDI from exported changed block data** Perform the steps in this section if you want to use your backed up data to restore a VDI to an earlier state.

Setting up changed block tracking

Before you can take incremental backups of a VDI using changed block tracking, you must first enable changed block tracking on the VDI and export a base snapshot. To set up changed block tracking for a VDI, complete the following steps

1. Use the Management API to establish a XenAPI session on the XenServer host:

```
import XenAPI

import shutil

import urllib3

import requests
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
session = XenAPI.xapi_local()

session.xenapi.login_with_password("&lt;user>", "&lt;password>",
"&lt;version>", "&lt;originator>")
```

2. **Optional**: If you intend to create a new VM and new VDIs to restore your backed up data to, you must also export your VM metadata. Ensure that you export a copy of the VM metadata any time your VM properties change. This can be done by using HTTPS or by using the command line.

```
session_id = session._session

url = ("https://%s/export_metadata?session_id=%s&uuid=%s"

        "&export_snapshots=false"

        % (&lt;xs_host>, session_id, &lt;vm_uuid>))



with requests.Session() as session:

    urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

    request = session.get(url, verify=False, stream=True)

    with open(&lt;export_path>, 'wb') as filehandle:

        shutil.copyfileobj(request.raw, filehandle)

    request.raise_for_status()
```

Where *<export_path>* is the location to save the VM metadata to.

The export URL includes the parameter `export_snapshots=false`. This parameter ensures that the snapshot history is not included in the VM metadata backup. The VM metadata is used to create a new VM and this snapshot history does not apply to the new VM.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

If you intend to use your backed up data only to restore existing VDIs and VMs, you can skip this step.

3. Get a reference for the VDI you want to snapshot:

```
vdi_ref = session.xenapi.VDI.get_by_uuid("&lt;vdi_uuid>")
```

4. Enable changed block tracking for the VDI:

```
session.xenapi.VDI.enable_cbt(&lt;vdi_ref>)
```

For more information, see Using changed block tracking with a virtual disk image.

5. Take a snapshot of the VDI:

```
base_snapshot_vdi_ref = session.xenapi.VDI.snapshot(&lt;vdi_ref>)
```

This VDI snapshot is the base snapshot.

6. Export the base VDI snapshot to the backup location. This can be done by using HTTPS or by using the command line.

For example, at the xe command line run:

```
xe vdi-export uuid=&lt;base-snapshot-vdi-uuid> filename=&lt;name of
export>
```

Or, in Python, you can use the following code:

```
session_id = session._session

url = ('https://%s/export_raw_vdi?session_id=%s&vdi=%s&format=raw'

       % (&lt;xs_host>, session_id, &lt;base_snapshot_vdi_uuid>))

with requests.Session() as http_session:

    urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

    request = http_session.get(url, verify=False, stream=True)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
    with open(&lt;export_path>, 'wb') as filehandle:

        shutil.copyfileobj(request.raw, filehandle)

    request.raise_for_status()
```

Where *<export_path>* is the location you want to write the exported VDI to.

7. Optional: For each VDI snapshot, delete the snapshot data, but retain the metadata:

```
session.xenapi.VDI.data_destroy(&lt;base_snapshot_vdi_ref>)
```

This frees up space on the host or SR.

For more information, see Deleting VDI snapshot data and retaining the snapshot metadata.

Taking incremental backups

After taking the initial VDI snapshot and exporting all the data, the following steps can be repeated every time an incremental backup is taken of the VDI. These incremental backups export only the blocks that have changed since the previous snapshot was taken.

To take an incremental backup, complete the following steps:

1. Check that changed block tracking is enabled:

```
is_cbt_enabled = session.xenapi.VDI.get_cbt_enabled(&lt;vdi_ref>)
```

If the value of `is_cbt_enabled` is not `true`, you must complete the steps in the *Setting up changed block tracking* section, before taking incremental backups. For more information, see Incremental backup sets.

If changed block tracking is disabled and this is unexpected, this state might indicate that the host or SR has crashed since you last took an incremental backup or that a XenServer user has disabled changed block tracking.

2. Take a snapshot of the VDI:

```
snapshot_vdi_ref = session.xenapi.VDI.snapshot(&lt;vdi_ref>)
```

3. Compare this snapshot to a previous snapshot to find the changed blocks:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
bitmap =
session.xenapi.VDI.list_changed_blocks(&lt;base_snapshot_vdi_ref>,
&lt;snapshot_vdi_ref>)
```

This call returns a base64-encoded bitmap that indicates which blocks have changed. For more information, see Get the list of blocks that changed between VDIs.

4. Get details for a list of connections that can be used to use to access the VDI snapshot over the NBD protocol.

```
connections = session.xenapi.VDI.get_nbd_info(&lt;snapshot_vdi_ref>)
```

This call returns a list of connection details that are specific to this session. Each set of connection details in the list contains a dictionary of the parameters required for an NBD client connection. For more information, see Getting NBD connection information for a VDI.

NOTE:

Ensure that this session with the host remains logged in until after you have finished reading from the network block device.

5. From your NBD client, complete the following steps to export the changed blocks to the backup location. For example, when using the Linux `nbd-client`:

   a. Connect to the NBD server.

```
nbd-client &lt;address> &lt;port> -N &lt;exportname> -cacertfile
&lt;cacert> -tlshostname &lt;subject>
```

   - The <*address*>, <*port*>, <*exportname*>, and <*subject*> values passed as parameters to the connection command are the values returned by the `get_nbd_info` call.
   - The <*cacert*> is a file containing one or more trusted Certificate Authority certificates of which at least one has signed the NBD server's TLS certificate. That TLS certificate is included in the values returned by the `get_nbd_info` call. If the TLS certificate returned by the `get_nbd_info` call is self-signed, it can be used as the value of `cacert` here to authenticate itself.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

For more information about using these values, see Getting NBD connection information for a VDI.

b. Read off the blocks that are marked as changed in the bitmap returned from step 3.

c. Disconnect from the block device:

```
nbd-client -d &lt;block_device>
```

d. Optional: We recommend that you retain the bitmap associated with each changed block export at your backup location.

To complete the preceding steps, you can use any NBD client implementation that supports the “fixed newstyle” version of the NBD protocol. For more information, see Exporting the changed blocks using an NBD clientl)l).

6. Optional: On the host, delete the VDI snapshot, but retain the metadata:

```
session.xenapi.vdi.data_destroy(&lt;snapshot_vdi_ref>)
```

This frees up space on the host or SR.

For more information, see Deleting VDI snapshot data and retaining the snapshot metadata.

Restoring a VDI from exported changed block data

When you want to use your incremental backups to restore or import data from a VDI, you cannot use individual exports of changed blocks to do this. You must first coalesce the exported changed blocks onto a base snapshot. Use this coalesced VDI to restore or import backed up data.

1. Create a coalesced VDI.

For each set of exported changed blocks between the base snapshot and the snapshot you want to restore to, create a coalesced VDI from a previous base VDI and the subsequent set of exported changed blocks. Ensure that you apply sets of the changed blocks to the base VDI in the order that they were snapshotted.

To create a coalesced VDI from a base VDI and the subsequent set of exported changed blocks, complete the following steps:

a. Get the bitmap that was used in step 3 to derive this set of exported changed blocks.

b. For each block in the VDI:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- If the bitmap indicates that the block has changed, read the block data from the set of exported changed blocks and append that data to the coalesced VDI.
- If the bitmap indicates that the block has not changed, read the block data from the base VDI and append that data to the coalesced VDI.

c. Use the coalesced VDI as the base VDI for the next iteration of this step. Or, if you have reached the target snapshot level, use this coalesced VDI in the next step to restore a VDI in XenServer.

For more information, see Coalescing changed blocks onto a base VDI.

You can now use this coalesced VDI to either import backed up data into a new VDI or to restore an existing VDI.

2. Optional: Create a new VM and new VDI.

   You can create a new VM and new VDI to import the coalesced VDI into. However, if you intend to use the coalesced VDI to restore an existing VDI, you can skip this step.

   To create a new VM and new VDI, complete the following steps

   a. Create a new VDI:

```
vdi_record = {

     "SR": &lt;sr>,

     "virtual_size": &lt;size>,

     "type": "user",

     "sharable": False,

     "read_only": False,

     "other_config": {},

     "name_label": "&lt;name_label>"

}

vdi_ref = session.xenapi.VDI.create(vdi_record)
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
vdi_uuid = session.xenapi.VDI.get_uuid(vdi_ref)
```

Where <*sr*> is a reference to the SR that the original VDI was located on and <*size*> is the size of the original VDI.

b. To create a new VM that uses the VDI created in the previous step, import the VM metadata associated with the snapshot level you are using to restore the VDI:

```
vdi_string = "&vdi:%s=%s" % (&lt;original_vdi_uuid>,
&lt;new_vdi_uuid>)



task_ref = session.xenapi.task.create("import_vm", "Task to track vm
import")



url = ('https://%s/import_metadata?session_id=%s&task_id=%s%s'

       % (host, session._session, task_ref, vdi_string))



with open(&lt;vm_import_path>, 'r') as filehandle:


urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

    with requests.Session() as http_session:

        request = http_session.put(url, filehandle, verify=False)

        request.raise_for_status()
```

Where <*vm_import_path*> is the location of the VM metadata.

The `vdi:` query parameter changes the VM from pointing to its original VDI to pointing to the new VDI created in the previous step. You might want to create multiple new VDIs. If you want to change multiple VDI references for your new VM, add a `vdi:` query parameter for each VDI to the import URL.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The new VM is created from the imported metadata and its VDI reference is updated to point at the VDI created in the previous step. You can extract a reference to this new VM from the result of the task. For more information, see the samples on GitHub.

3. Import the coalesced VDI snapshot to the XenServer host at the UUID of the VDI you want to replace with the restored version. This VDI can be either an existing VDI or the VDI created in the previous step.

   In Python, you can use the following code:

```
session_id = session._session

url = ('https://%s/import_raw_vdi?session_id=%s&vdi=%s&format=%s'

       % (&lt;xs_host>, session_id, &lt;vdi_uuid>, 'raw'))

with open(&lt;import_path>, 'r') as filehandle:

    urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

    with requests.Session() as http_session:

        request = http_session.put(url, filehandle, verify=False)

        request.raise_for_status()
```

   Where <*vdi_uuid*> is the UUID of the VDI you want to overwrite with the restored data from the coalesced VDI and <*import_path*> is the location of the coalesced VDI.

Enabling NBD connections on XenServer®

XenServer acts as a network block device (NBD) server and makes VDI snapshots available over NBD connections. However, to connect to XenServer over an NBD connection, you must enable NBD connections for one or more networks.

IMPORTANT

We recommend that you use a dedicated network for your NBD traffic.

By default, NBD connections are not enabled on any networks.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NOTE:

Networks associated with a XenServer pool that have NBD connections enabled must either all have the `nbd` purpose or all have the `insecure_nbd` purpose. You cannot have a mix of normal NBD networks (`FORCEDTLS`) and insecure NBD networks (`NOTLS`). To switch the purpose of all networks, you must first disable normal NBD connections on all networks before enabling either normal or insecure NBD connections on any networks.

Enabling an NBD connection for a network (`FORCEDTLS` mode)

We recommend that you use TLS in your NBD connections. When NBD connections with TLS are enabled, any NBD clients that attempt to connect to XenServer must use TLSv1.2. The NBD server runs in `FORCEDTLS` mode with the "fixed newstyle" NBD handshake. For more information, see the NBD protocol documentation.

To enable NBD connections with TLS, use the `purpose` parameter of the network. Set this parameter to include the value `nbd`. Ensure that you wait for the setting to propagate before attempting to use this network for NBD connections. The time it takes for the setting to propagate depends on your network and is at least 10 seconds. We recommend that you use a retry loop when making the NBD connection.

Examples

You can use any of our supported language bindings to enable NBD connections. The following examples show how to do it in Python and at the xe command line.

Python:

```
session.xenapi.network.add_purpose(<network_ref>, "nbd")
```

xe command line:

```
xe network-param-add param-name=purpose param-key=nbd uuid=<network-uuid>
```

Enabling an insecure NBD connection for a network (`NOTLS` mode)

We recommend that you do not enable insecure NBD connections. Instead use `FORCEDTLS` NBD connections. However, the ability to connect to the XenServer over an insecure NBD connection is

© 1997–2026 Citrix Systems, Inc. All rights reserved.

provided for development and testing with the NBD server operating in `NOTLS` mode as described in the NBD protocol.

To enable insecure NBD connections, use the `purpose` parameter of the network. Set this parameter to include the value `insecure_nbd`. Ensure that you wait for the setting to propagate before attempting to use this network for NBD connections. The time it takes for the setting to propagate depends on your network and is at least 10 seconds. We recommend that you use a retry loop when making the NBD connection.

Examples

You can use any of our supported language bindings to enable insecure NBD connections. The following examples show how to do it in Python and at the xe command line.

Python:

```
session.xenapi.network.add_purpose(<network_ref>, "insecure_nbd")
```

xe command line:

```
xe network-param-add param-name=purpose param-key=insecure_nbd
uuid=<network-uuid>
```

Disabling NBD connections for a network

To disable NBD connections for a network, remove the NBD values from the `purpose` parameter of the network.

Examples

You can use any of our supported language bindings to disable NBD connections. The following examples show how to do it in Python and at the xe command line.

Python:

```
session.xenapi.network.remove_purpose(<network_ref>, "nbd")
```

Or, for insecure NBD connections:

```
session.xenapi.network.remove_purpose(<network_ref>, "insecure_nbd")
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

xe command line:

```
xe network-param-remove param-name=purpose param-key=nbd uuid=<network-
uuid>
```

Or, for insecure NBD connections:

```
xe network-param-remove param-name=purpose param-key=insecure_nbd
uuid=<network-uuid>
```

Using changed block tracking with a virtual disk image

The changed block tracking capability can be enabled and disabled for individual virtual disk images (VDIs).

Incremental backup sets

When you enable changed block tracking for a VDI you start a new set of incremental backups for that VDI. The first action you must take when starting a set of incremental backups is to create a baseline snapshot and to backup its full data.

After you disable changed block tracking, or after changed block tracking is disabled by XenServer® or a user, no further incremental backups can be added to this set. If changed block tracking is enabled again, you must take another baseline snapshot and start a new set of incremental backups.

You cannot compare VDI snapshots taken as part of one set of incremental backups with VDI snapshots taken as part of a different set of incremental backups. If you attempt to list the changed blocks between snapshots that are part of different sets, you get an error with the message `Source and target VDI are unrelated`.

You can use some or all of the data in previous incremental backup sets to create VDIs that you can use to restore the state of a VDI. For more information, see Coalescing changed blocks onto a base VDI.

Enabling changed block tracking for a VDI

By default, changed block tracking is not enabled for a VDI. You can enable changed block tracking by using the `enable_cbt` call.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

When changed block tracking is enabled for a VDI, additional log files are created on the SR to list the changes since the last backup. Blocks of 64 kB within the VDI are tracked and changes to these blocks recorded in the log layer.

The associated VM remains in the same state as before changed block tracking was enabled.

To enable changed block tracking, an SR must be attached, be writable, and have enough free space for the log files to be created on it. The associated VM can be in any state when changed block tracking is enabled or disabled. It is not required that the VM be offline.

Changed block tracking can only be enabled for a VDI that is one of the following types:

- user
- system

In addition, if the `VDI.on_boot` field is set to `reset`, you cannot enable changed block tracking for the VDI.

Examples

You can use any of our supported language bindings to enable changed block tracking for a VDI. The following examples show how to do it in Python and at the xe command line.

Python:

```
session.xenapi.VDI.enable_cbt(<vdi_ref>)
```

xe command line:

```
xe vdi-enable-cbt uuid=<vdi-uuid>
```

Errors

You might see the following errors when using this call:

VDI_MISSING:

- The call cannot find the VDI.

  Check that the reference or UUID you are using to refer to the VDI is correct. Check that the VDI exists.

VDI_INCOMPATIBLE_TYPE:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- The VDI is of a type that does not support changed block tracking.

  Check that the type of the VDI is `system` or `user`. You can use the `get_type` call to find out the type of a VDI. If your VDI is an incompatible type, you cannot enable changed block tracking. For more information, see Checking the type of a VDI or VDI snapshot.

VDI_ON_BOOT_MODE_INCOMPATIBLE_WITH_OPERATION:

- The value of the `on_boot` field of the VDI is set to `reset`.

  Check the value of the `on_boot` field by using the `get_on_boot` call. If appropriate, you can use the `set_on_boot` call to change the value of this field to `persist`.

SR_NOT_ATTACHED, SR_HAS_NO_PBDS:

- The call cannot find an attached SR.

  Check that there is an SR attached to the host and that the SR is writable. You cannot enable changed block tracking unless the host has access to an SR to which the changed block information can be written.

If you attempt to enable changed block tracking for a VDI that already has changed block tracking enabled, no error is thrown.

Disabling changed block tracking for a VDI

You can disable changed block tracking for a VDI by using the `disable_cbt` call.

When changed block tracking is disabled for a VDI, the active disks are detached and reattached without the log layer. The associated VM remains in the same state as before changed block tracking was disabled.

Examples

You can use any of our supported language bindings to disable changed block tracking for a VDI. The following examples show how to do it in Python and at the xe command line.

Python:

```
session.xenapi.VDI.disable_cbt(<vdi_ref>)
```

xe command line:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
xe vdi-disable-cbt uuid=<vdi-uuid>
```

Errors

You might see the same sorts of errors for this call and you might for the `enable_cbt` call.

If you attempt to disable changed block tracking for a VDI that already has changed block tracking disabled, no error is thrown.

Checking whether changed block tracking is enabled

The value of the boolean `cbt_enabled` VDI field shows whether changed block tracking is enabled for that VDI. You can query the value of this field by using the `get_cbt_enabled` call.

A return value of `true` indicated that changed block tracking is enabled for this VDI.

Examples

You can use any of our supported languages to check whether a VDI has changed block tracking enabled. The following examples show how to do it in Python and at the xe command line.

Python:

```
is_cbt_enabled = session.xenapi.VDI.get_cbt_enabled(<vdi_ref>)
```

xe command line:

```
xe vdi-param-get param-name=cbt-enabled uuid=<vdi-uuid>
```

Deleting VDI snapshot data and retaining the snapshot metadata

A VDI snapshot is made up of both data and metadata. The data is the full image of the disk at the time the snapshot was taken. The metadata includes the changed block tracking information.

After the snapshot data on the host has been exported to the backup location, you can use the `data_destroy` call to delete only the snapshot data and retain only the snapshot metadata on the host. This action converts the snapshot that is stored on the host or SR into a smaller metadata-only snapshot. The `type` field of the snapshot changes to be `cbt_metadata`.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Metadata-only snapshots are linked to the metadata-only snapshots that precede and follow them in time.

You can use the `data_destroy` call only for snapshots for VDIs that have changed block tracking enabled.

NOTE:

The API also provides a `destroy` call, which deletes both the data in the snapshot and the metadata in the snapshot.

Do not use the `destroy` call to delete snapshots that are part of a set of changed block tracking backups unless you are sure that you no longer need the changed block tracking metadata.

For example, use `destroy` to remove a metadata-only snapshot that is older than age allowed by your retention policy.

Examples

You can use any of our supported language bindings to delete the data in a snapshot and convert the snapshot to a metadata only snapshot. The following examples show how to do it in Python and at the xe command line.

Python:

```
session.xenapi.VDI.data_destroy(<snapshot_vdi_ref>)
```

xe command line:

```
xe vdi-data-destroy uuid=<snapshot_vdi_uuid>
```

Errors

You might see the following errors when using this call:

VDI_MISSING:

- The call cannot find the VDI snapshot.

  Check that the reference or UUID you are using to refer to the VDI snapshot is correct. Check that the VDI exists.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

VDI_NO_CBT_METADATA:

- No changed block tracking metadata exists for this VDI snapshot.

  Check that changed block tracking is enabled for the VDI. You cannot use the `data_destroy` call on VDIs that do not have changed block tracking enabled. For more information, see Using changed block tracking with a virtual disk image.

VDI_IN_USE:

- The VDI snapshot is currently in use by another operation.

  Check that the VDI snapshot is not being accessed by another client or operation. Check that the VDI is not attached to a VM.

  If the VDI snapshot is connected to a VM snapshot by a VBD, you receive this error. Before you can run `VDI.data_destroy` on this VDI snapshot, you must remove the VM snapshot. Use `VM.destroy` to remove the VM snapshot.

Checking the type of a VDI or VDI snapshot

The value of the `type` VDI field shows what type of VDI or VDI snapshot an object is. The values this field can have are stored in the `vdi_type` enum. You can query the value of this field by using the `get_type` call.

A metadata-only VDI snapshot has the type `cbt_metadata`.

Examples

You can use any of our supported language bindings to query the VDI type of a VDI or VDI snapshot. The following examples show how to do it in Python and at the xe command line.

Python:

```
vdi_type = session.xenapi.VDI.get_type(<snapshot_vdi_ref>)
```

xe command line:

```
xe vdi-param-get param-name=type uuid=<snapshot_vdi_uuid>
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Getting the list of blocks that changed between VDIs

You can use the `list_changed_blocks` call to get a list of the blocks that have changed between two VDIs. Both VDI snapshots must be taken after changed block tracking is enabled on the VDI.

This call takes as parameters references to two VDI snapshots:

- `VDI_from`: The earlier VDI snapshot.
- `VDI_to`: The later VDI snapshot. This VDI *cannot* be attached to a VM at the time this comparison is made.

This operation does not require the VM associated with the VDIs to be offline at the time the comparison is made.

The changed blocks are listed in a base64-encoded bitmap. Each bit in the bitmap indicates whether a 64 kB block in the VDI has been changed in comparison to an earlier snapshot. A bit set to 0 indicates that the block is the same. A bit set to 1 indicates that the block has changed.

The bit in the first position in the bitmap represents the first block in the VDI. For example, if the bitmap is 01100000, this indicates that the first block has not changed, the second and third blocks have changed, and all other blocks have not changed.

Examples

You can use any of our supported languages to get the bitmap that lists the changed blocks between two VDI snapshots. The following examples show how to do it in Python and at the xe command line.

Python:

```
bitmap =
session.xenapi.VDI.list_changed_blocks(<previous_snapshot_vdi_ref>,
<new_snapshot_vdi_ref>)
```

You can convert the base64-encoded bitmap this call returns into a human-readable string of 1s and 0s:

```
from bitstring import BitStream

import base64
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
data = BitStream(bytes=base64.b64decode(bitmap))
```

xe command line:

```
xe vdi-list-changed-blocks vdi-from-uuid=<previous_snapshot_vdi_uuid> vdi-to-uuid=<new_snapshot_vdi_uuid>
```

Errors

You might see the following errors when using this call:

VDI_MISSING:

- The call cannot find one or both of the VDI snapshots.

  Check that the reference or UUID you are using to refer to the VDI snapshot is correct. Check that the VDI snapshot exists.

VDI_IN_USE:

- The VDI snapshot is currently in use by another operation.

  Check that the VDI snapshot is not being accessed by another client or operation. Check that the more recent VDI snapshot is not attached to a VM. The newer VDI in the comparison cannot be attached to a VM at the time of the comparison.

Source and target VDI are unrelated:

- The VDI snapshots are not linked by changed block metadata.

  You can only list changed blocks between snapshots that are taken as part of the same set of incremental backups. For more information, see Incremental backup sets.

Export changed blocks over a network block device connection

XenServer® runs an NBD server on the host that can make VDI snapshots accessible as a network block device to NBD clients. The NBD server listens on port 10809 and uses the "fixed newstyle" NBD protocol. For more information, see the NBD protocol documentation.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

NBD connections must be enabled for one or more of the XenServer networks before you can export the changed blocks over NBD. For more information, see Enabling NBD connections on XenServer.

Getting NBD connection information for a VDI

From a logged in XenAPI session, you can use the `get_nbd_info` call to get a list of connection details for a VDI snapshot made available as a network block device.

These connection details are specific to the session that creates them and the NBD client uses this logged in session when making its connection. Any set of connection details in the list can be used by the NBD client when accessing the VDI snapshot.

Each set of connection details in the list is provided as a dictionary containing the following information:

`address`:

- The IP address (IPv4 or IPv6) of the NBD server.

`port`:

- The TCP port to connect to the XenServer NBD server on.

`cert`:

- The TLS certificate used by the NBD server encoded as a string in PEM format. When XenServer is configured to enable NBD connections in `FORCEDTLS` mode, the server presents this certificate during the TLS handshake and the NBD client must verify the server TLS certificate against this TLS certificate. For more information, see "Verifying TLS certificates for NBD connections".

`exportname`:

- A token that the NBD client can use to request the export of a VDI from the NBD server. The NBD client provides the value of this token to the NBD server using the `NBD_OPT_EXPORT_NAME` option during the NBD option haggling phase of an NBD connection.

  This token contains a reference to a logged in XenAPI session. The XenAPI session must remain logged in for this token to continue to be valid. Because the token contains a reference to a XenAPI session, you must handle the token securely to prevent the session being hijacked.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The format of this token is not guaranteed and might change in future releases of XenServer. Treat the export name as an opaque token.

`subject`:

- A subject of the TLS certificate returned as the value of `cert`. This field is provided as a convenience.

Examples

You can use any of our supported languages to get the list of NBD connection details for a VDI snapshot. The following examples show how to do it in Python.

Python:

```
connection_list = session.xenapi.VDI.get_nbd_info(<snapshot_vdi_ref>)
```

This call requires a logged in XenAPI session that remains logged in while the VDI snapshot is accessed over NBD. This means that this command is not available at the xe command line.

Errors

You might see the following errors when using this call:

VDI_INCOMPATIBLE_TYPE:

- The VDI is of a type that does not support being accessed as a network block device.

  Check that the type of the VDI is not `cbt_metadata`. You can use the `get_type` call to find out the type of a VDI. If your VDI is `cbt_metadata`, you cannot access it as a network block device. For more information, see Checking the type of a VDI or VDI snapshot.

An empty list of connection details:

- The VDI cannot be accessed.

  Check that the XenServer host that runs the NBD server has a PIF with an IP address.

  Check that you have at least one network in your pool with the purpose `nbd` or `insecure_nbd`. For more information, see Enabling NBD connections on XenServer.

  Check that storage repository the VDI is on is attached to a host that is connected to one of the NBD-enabled networks.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

Exporting the changed blocks using an NBD client

An NBD client running in the backup location can connect to the NBD server that runs on the XenServer host and access the VDI snapshot by using the provided connection details.

The NBD client that you use to connect to the XenServer NBD server can be any implementation that supports the “fixed newstyle” version of the NBD protocol.

When choosing or developing an NBD client implementation, consider the following requirements:

- The NBD client must support the “fixed newstyle” version of the NBD protocol. For more information, see the NBD protocol documentation.
- The NBD client must request an export name returned by the `get_nbd_info` call that corresponds to an existing logged in XenAPI session. The client makes this request by using the `NBD_OPT_EXPORT_NAME` option during the NBD option haggling phase of the NBD connection.
- The NBD client must verify the TLS certificate presented by the NBD server by using the information returned by the `get_nbd_info` call. For more information, see “Verifying TLS certificates for NBD connections”.

NOTE:

If you are using the Linux upstream NBD client, a minimum version of 3.15 is required to support TLS.

After the NBD client has made a connection to the XenServer host and accessed the VDI snapshot, you can use the bitmap provided by the `list_changed_blocks` call to select which blocks to read. For more information, see Getting the list of blocks that changed between VDIs.

NOTE:

XenServer supports up to 16 concurrent NBD connections.

Verifying TLS certificates for NBD connections

When connecting to the NBD server using TLS, the NBD client must verify the certificate that the server presents as part of the TLS handshake.

We recommend that you use one of the following methods of verification depending on your NBD client implementation:

© 1997–2026 Citrix Systems, Inc. All rights reserved.

- Verify that the server certificate matches the certificate returned by the `get_nbd_info` call.
- Verify that the public key of the server certificate matches the public key of the certificate returned by the `get_nbd_info` call.

Alternative approach

As a less preferred option, it is possible for the NBD client to verify the certificate that the server presents during the TLS handshake by checking that the certificate meets all of the following criteria:

- It is signed by a trusted Certificate Authority
- It has an `Alternative Subject Name` (or, if absent, a `Subject`) that matches the subject returned by the `get_nbd_info` call.

Coalescing changed blocks onto a base VDI

When using backed up data to restore the state of a VDI, you must import a full VDI into XenServer. You cannot import only the sets of changed blocks. To import incremental backups created with changed block tracking data into XenServer, you must first use these incremental backups to create a full VDI.

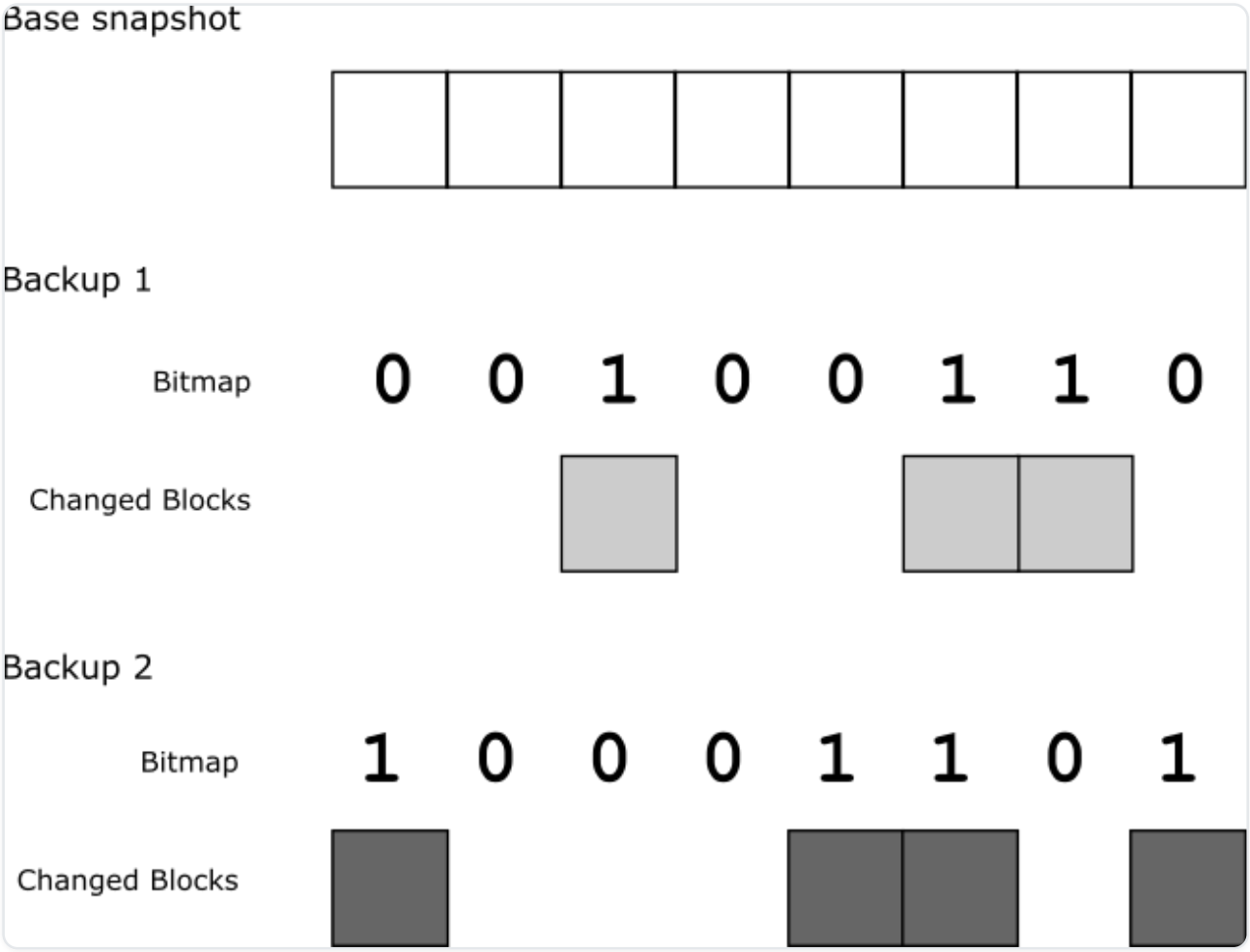
A set of incremental backups created with changed block tracking can be used to create a full VDI whose data is identical to the source VDI at the time an incremental backup was taken.

For more information about incremental backup sets, see Incremental backup sets.

For example, you have a set of incremental backups that comprises:

- A base snapshot that captures the data for the full VDI.
- Backup 1: The first incremental backup, which consists of a bitmap list of blocks changed since the base snapshot and the data for only those changed blocks.
- Backup 2: The second incremental backup, which consists of a bitmap list of blocks changed since backup 1 and the data for only those changed blocks.

© 1997–2026 Citrix Systems, Inc. All rights reserved.



If you want to restore a VDI on XenServer to the state it was at when backup 2 was taken, you must create a VDI that takes blocks from the base snapshot, changed blocks from backup 1, and changed blocks from backup 2. To do this, you can apply each set of changed blocks in sequence to the base snapshot of the VDI.

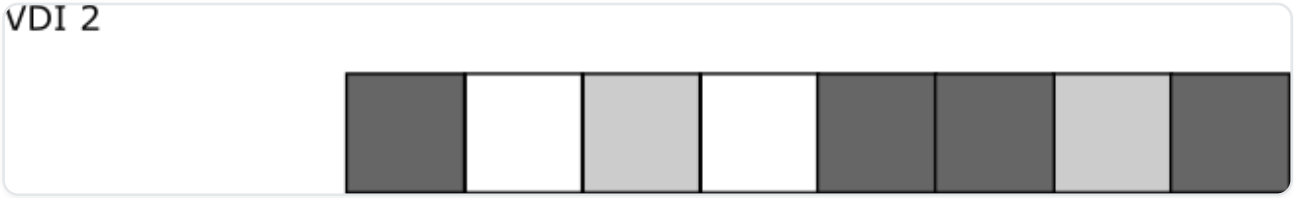
First build up a coalesced VDI by taking unchanged blocks from the base snapshot and changed blocks from those exported at backup 1. The bitmap list of changed blocks that was used to create backup 1 defines which blocks are changed.

After coalescing the base snapshot with the changed blocks exported at backup 1, you have a full VDI whose data is identical to that of the source VDI at the time backup 1 was taken. Call this coalesced VDI "VDI 1".

Next, use this coalesced VDI, VDI 1, to create another coalesced VDI by taking unchanged blocks from VDI 1 and changed blocks from those exported at backup 2. The bitmap list of changed blocks that was used to create backup 2 defines which blocks are changed.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

After coalescing VDI 1 with the changed blocks exported at backup 2, you have a full VDI whose data is identical to that of the source VDI at the time backup 2 was taken. Call this coalesced VDI "VDI 2".



This coalesced VDI, VDI 2, can be used to restore the state of the VDI on XenServer at the time that a snapshot was taken for backup 2.

When creating a coalesced VDI, ensure that you work with your VDIs and changed blocks as binary.

Ensure that you verify the integrity of the backed up and restored VDIs. For example, you can do this by computing the checksums of the data.

Examples

The following example shows how to create a coalesced VDI. The example shows applying a single set of changed blocks to the base VDI snapshot. To apply multiple sets of changed blocks, you must repeat this process for each set of changed blocks in order from oldest to most recent, using the output from the previous iteration as the base VDI for the next iteration.

Python:

```
def write_changed_blocks_to_base_VDI(vdi_path, changed_block_path,
bitmap_path, output_path):

    bitmap = open(bitmap_path, 'r')

    vdi = open(vdi_path, 'r+b')

    blocks = open(changed_block_path, 'r+b')

    combined_vdi = open(output_path, 'wb')



    try:

        bitmap_r = bitmap.read()
```

© 1997–2026 Citrix Systems, Inc. All rights reserved.

```
        cb_offset = 0

        for x in range(0, len(bitmap_r)):

            offset = x * changed_block_size

            if bitmap_r[x] == "1":

                blocks.seek(cb_offset)

                blocks_r = blocks.read(changed_block_size)

                combined_vdi.write(blocks_r)

                cb_offset += changed_block_size

            else:

                vdi.seek(offset)

                vdi_r = vdi.read(changed_block_size)

                combined_vdi.write(vdi_r)
```

Troubleshoot Changed Block Tracking

This article includes some common error scenarios you might encounter while enabling and using changed block tracking.

Common error scenarios

You can't enable changed block tracking on a VDI

- Ensure that the VDI is not a snapshot or a raw VDI. You can't enable changed block tracking on snapshots or raw VDIs.

You can't data destroy a VDI

- Ensure that the VDI you are referring to is a snapshot.
- Ensure that changed block tracking is enabled. Otherwise, you might see the `VDI_NO_CBT_METADATA` error.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

  - In XenCenter®, check the **Storage** tab to see if changed block tracking is enabled for the VDI you took a snapshot of.
  - Or, use the xe CLI to check the `cbt_enabled` field of the VDI snapshot: `xe vdi-param-list uuid=<snapshot_uuid>`

- Check whether a `<snapshot_uuid>.cbtlog` file exists in the storage repository:
  - For LVM-based storage repositories, run the command `lvs` to check whether a `<snapshot_uuid>.cbtlog` file exists in the storage repository.
  - For file-based storage repositories, look for the files in the location `/run/sr-mount/<sr-uuid>/<snapshot-uuid>.cbtlog`.

You can't list changed blocks between two VDI snapshots

- Ensure that the VDIs are snapshots. If the `vdi_to` VDI is not a snapshot, ensure it is not attached.
- Ensure that both VDI snapshots have changed block tracking enabled.
- Ensure that the VDI snapshots are related and in the right order.

  You can use the `cbt-util` utility, which helps establish chain relationship. If the VDI snapshots are not linked by changed block metadata, you get errors like "SR_BACKEND_FAILURE_460", "Failed to calculate changed blocks for given VDIs", and "Source and target VDI are unrelated".

  Example usage of cbt-util:

  ```
  cbt-util get –c –n &lt;name of cbt log file>
  ```

  The `-c` option prints the child log file UUID.

Notable error conditions

Changed block tracking is disabled when certain errors are encountered, for example:

- The changed block tracking log is found to be inconsistent at the time of attaching a VDI. This situation can occur when a XenServer® host or SR crashes.
- The resize of a changed block tracking log file was unsuccessful on the source VDI resize.
- Insufficient space remains on disk to create a changed block tracking log file when changed block tracking is activated or a snapshot is created.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

However, the primary action (for example, attach, resize, or snapshot) does succeed in those error conditions and a log message is logged in SMlog. Also, an alert is generated in XenCenter to notify you that changed block tracking is disabled.

Troubleshooting the NBD server

XenServer acts as network block device (NBD) server and makes VDI snapshots available over NBD connections. For more information, see Enabling NBD connections on XenServer.

Notes:

- The NBD server logs are in `/var/log/daemon.log`.
- NBD connections work with all network configurations, including VLAN, bond network, and VLAN on bond.
- NBD connections don't work when an NBD client is in dom0 on the same host as the NBD server.
- Networks associated with a XenServer pool that have NBD connections enabled must either all have the purpose `nbd` or all have the purpose `insecure_nbd`. You cannot have a mix of normal NBD networks (FORCEDTLS) and insecure NBD networks (NOTLS).
- The service (xapi-nbd) shows the state "failed" after it stopped. This state does not indicate an error.

Common NBD connection issues

The command `get_nbd_info` returns an empty list of connection details

- Ensure that the XenServer host that runs the NBD server has a PIF with an IP address.
- Ensure that you have at least one network in your pool with the purpose `nbd` or `insecure_nbd`.
- Ensure that the storage repository that the VDI is on is attached to a host that is connected to one of the NBD-enabled networks.

You can't access the IP address returned by the command `get_nbd_info`

- Check whether NBD is enabled on a network that is not reachable by the client.
- Check whether multiple networks are mixed on the same subnet and NBD is allowed on some of them but blocked on others.
- Check the NBD network configuration.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The connection refused or closed, or the NBD client hangs

- Verify whether the session that the client passes to the NBD server is valid. If the session has expired or is not valid, you see the error "SESSION_INVALID". The session has to be valid for the time of the backup, otherwise, the NBD server refuses the connection or the client might hang.
- The maximum parallel connection limit might have been exceeded. To work around this, you can restart xapi-nbd.
- If the NBD server is configured to use TLS, ensure that the client is able to use TLS.

Appendices

Constraints

The following section lists advisories and constraints to consider when using changed block tracking.

- GFS2 and XFS SR types are supported with changed block tracking in XenServer 8.4 and later.
- Changed block tracking is available only to customers with a Premium Edition license for XenServer. If a customer without a Premium Edition license attempts to use an incremental backup solution for XenServer that uses changed block tracking, they are prevented from enabling changed block tracking on new VDIs. However, if the customer has existing VDIs with changed block tracking enabled, they can still perform other changed block tracking actions on these VDIs.
- Changed block tracking information is lost on storage live migration. If you attempt to migrate a VM that has VDIs with changed block tracking enabled, you are prevented from doing so. You must disable changed block tracking before storage live migration is allowed.
- If a host or an SR crashes, XenServer disables changed block tracking for all VDIs on that host or SR. Before taking a VDI snapshot, we recommend that you check whether changed block tracking is enabled. If changed block tracking is disabled and this is not expected, this can indicate that a crash has occurred or that a XenServer user has disabled changed block tracking.

  To continue using changed block tracking, you must enable changed block tracking again and create a new baseline by taking a full VDI snapshot. Subsequent changed block tracking metadata uses this snapshot as a new baseline.

© 1997–2026 Citrix Systems, Inc. All rights reserved.

The set of snapshots and changed block tracking data captured before the crash cannot be used as a baseline or comparison for any snapshots taken after the crash. However, the set of incremental backups taken before the crash can be used to create a VDI image to use to restore the VDI to a previous state.

For more information, see Incremental backup sets.

- XenServer supports a maximum of 16 concurrent NBD connections.

Third-party solutions

For a list of third-party solutions that use the changed block tracking feature, see Third-party backup solutions.

Additional Resources

The following resources provide additional information:

- GitHub repository of sample code
- XenServer Management API Guide
- XenServer Software Development Kit Guide
- NBD protocol documentation
- nbd-client manpage

© 1997–2026 Citrix Systems, Inc. All rights reserved.

fix

This is the content for the sample heading topic. It is hidden from the main table of contents but remains accessible via links.

© 1997–2026 Citrix Systems, Inc. All rights reserved.