

CA Server Automation

Reference Guide

Release 12.8.1



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document may reference the following CA Technologies products and components or third-party components:

- CA Configuration Automation, formerly CA Application Configuration Manager (CA ACM)
- CA eHealth®
- CA Embedded Entitlements Manager (CA EEM)
- CA IT Asset Manager (CA ITAM)
- CA IT Client Manager (CA ITCM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Process Automation, formerly CA IT Process Automation Manager (CA IT PAM)
- CA Server Automation
- CA Service Desk Manager (CA SDM)
- CA Software Delivery, a component of CA IT Client Manager
- CA Spectrum® Infrastructure Manager (CA Spectrum)
- CA SystemEDGE

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Introduction	17
Scope of This Guide	17
Related Publications	17
Conventions	18
Chapter 2: AutoShell	21
About AutoShell	21
Manager Shell and Client Shell	23
Valid AutoShell User	23
Accessing AutoShell	24
Common Information Model (CIM) Objects	24
Retrieve CIM Classes	24
Retrieve Antecedents and Dependents From CIM Objects	25
Retrieve CIM Objects	26
Invoking AutoShell	28
Use Interactive Mode	30
Use Remote Interactive Mode	31
Batch Mode	31
Remote Batch Mode	32
Stringification	33
Writing Data to the Console (stdout)	34
Reading Data from the Keyboard (stdin)	35
Accessing the Operating System	36
Using the OSRedirect Class	36
Running Scripts	37
Using the run-local Command	38
Using the run-remote Command	40
AutoShell Core Reference	46
Commands, Functions, and Classes According to Categories	46
General AutoShell Commands	49
AutoShell Functions	111
AutoShell Classes	135
AutoShell Loadable Modules' Command Reference	160
CA VMware vCenter Server AutoShell Commands	160
CA Microsoft Hyper-V AutoShell Commands	200
CA IBM LPAR AutoShell Commands	300

CA Oracle Solaris Zones AutoShell Commands	327
Deployment and Policy Configuration AutoShell Commands	347

Chapter 3: Command Line Scripting 383

Command-Line Instructions (CLIs)	383
Amazon Elastic Compute Cloud Commands	384
dpmec2 createImage--Create AMI Image	384
dpmec2 ec2jobcheck Command--Get Amazon EC2 Job Status.....	385
dpmec2 getimages Command--Get Amazon EC2 Images	387
dpmec2 getinstances Command--Get Amazon EC2 Instances.....	388
dpmec2 getlist Command--Get Amazon EC2 List.....	389
dpmec2 imgjobcheck Command--Get Amazon EC2 Imaging Job Status.....	390
dpmec2 manage Command--Manage Amazon Elastic Compute Cloud Instances	392
dpmec2 run Command--Create Amazon EC2 Instance	393
CA Cisco UCS Commands	395
dpmucs associateserviceprofile--Associate Service Profile With a Blade	397
dpmucs cloneserviceprofile--Clone a Service Profile	398
dpmucs createmacpool Command--Create MAC Address Pool.....	399
dpmucs createorg Command--Create Organization	401
dpmucs createserverpool Command--Create Server Pool	402
dpmucs createserviceprofile Command--Create a Service Profile	404
dpmucs createtemplatefromserviceprofile Command--Create a Template from a Service Profile	405
dpmucs createuuidpool Command--Create UUID Pool	407
dpmucs createwwpool Command--Create WWNN Pool	408
dpmucs deletemacpool Command--Delete MAC Address Pool.....	410
dpmucs deleteorg Command--Delete Organization	411
dpmucs deleteserverpool Command--Delete Server Pool.....	413
dpmucs deleteserviceprofile--Delete a Service Profile	414
dpmucs deleteuuidpool Command--Delete UUID Pool	415
dpmucs deletewwpool Command--Delete WWNN Pool	416
dpmucs deployimage Command--Deploy a Cisco UCS Image.....	418
dpmucs disassociateserviceprofile Command--Disassociate Service Profile	419
dpmucs failoverserviceprofile Command--Associate a Service Profile With a Blade for Failover	421
dpmucs getblades Command--Get Blade Information	422
dpmucs getchassis Command--Get Chassis Information	423
dpmucs getpolicy Command--Gets Policy from a UCS Manager	425
dpmucs getserviceprofilestatus Command--Get Service Profile Status.....	426
dpmucs getswitches Command--Get Switch Information	427
dpmucs gettopsystem Command--Get System Information	428
dpmucs listserviceprofiles Command--List Service Profiles	429
dpmucs powercycle Command--Power Cycle Blades	431

dpmucs renamepool Command--Rename Pool	433
dpmucs serviceprofileupdate Command--Execute Service Profile Operations	434
CA Hyper-V CLI Commands	436
dpmhyperv AddVMNic Command--Add a Network Adapter Controller to a VM	436
dpmhyperv AddVMSCSIController Command--Create a SCSI Controller for a VM	438
dpmhyperv AgentGet Command--Return an Agent Value	439
dpmhyperv AgentSet Command--Set an Agent Value	440
dpmhyperv ChangeVMState Command--Change the State of a VM	442
dpmhyperv CreateTemplateFromVM Command--Create a Template from a VM	444
dpmhyperv CreateVirtDisk Command--Create a Virtual Disk Image	445
dpmhyperv CreateVirtFloppy Command--Create a Virtual Floppy Disk Image.....	447
dpmhyperv CreateVM Command--Create a VM	448
dpmhyperv CreateVMFromTemplate Command--Create a VM from a SCVMM Server Template	452
dpmhyperv CreateVMFromTemplate Command--Create a VM from a Template	456
dpmhyperv DeleteVM Command--Delete a VM	463
dpmhyperv ExpandVirtDisk Command--Expand the Size of a Virtual Disk	464
dpmhyperv ExportVM Command--Export a VM	466
dpmhyperv ExportVMEx Command--Export a VM	467
dpmhyperv GetHostSwitches Command--Get Switches of a Host	469
dpmhyperv GetJobErrorInfo Command--Get the Error Information of a Job.....	470
dpmhyperv GetJobInfo Command--Get Information of a Job	471
dpmhyperv GetJobState Command--Get the State of a Job	473
dpmhyperv GetServerInfo Command--Get all Managed Servers	474
dpmhyperv GetVersion Command--Get the PMM Version	475
dpmhyperv GetVMProperties Command--Get VM Properties	476
dpmhyperv GetVMState Command--Get a VM State	477
dpmhyperv ImportVM Command--Import a VM	478
dpmhyperv ImportVMEx Command--Copy and Import a VM	480
dpmhyperv SetClustered Command--Add or Remove a VM from a Cluster.....	481
dpmhyperv SetSysPrepProperties Command--Sets Sysprep Properties.....	483
dpmhyperv SetVMDisk Command--Assign a Drive or Disk to a Drive Controller.....	486
dpmhyperv SetVMProperties Command--Set Properties of a VM	488
dpmhyperv SetVMVirtFloppy Command--Assign a Virtual Floppy Image to a VM Floppy Drive	491
dpmhyperv ShowClusterSharedVolumes Command--Show all Cluster Shared Volumes	492
dpmhyperv ShowDirectories Command--Show Directory Properties on a Hyper-V Server	493
dpmhyperv ShowLogicalDisks Command--Show all Logical Disks	494
dpmhyperv ShowPhysDisks Command--Show Physical Disks of a Host	495
dpmhyperv ShowSCVMMHardwareProfiles Command--Show all SCVMM Hardware Profiles	496
dpmhyperv ShowSCVMMGuestOSProfiles Command--Show Guest Operating System Profiles.....	497
dpmhyperv ShowTemplates Command--Show all VM Templates.....	498
dpmhyperv ShowVirtFloppy Command--Show the List of Virtual Floppy Drives.....	499
dpmhyperv ShowVMDisks Command--Show all Virtual Disks	501

dpmhyperv ShowVMNics Command--Show all Network Interface Cards	502
dpmhyperv ShowVMs Command--Show all VMs.....	503
dpmhyperv ShowVMSCSIControllers Command--Show all SCSI Controllers	504
dpmhyperv ValidateAgentSNMPAccess Command--Validate SNMP Access to an Agent	505
dpmhyperv ValidateServerAccess Command--Check Server Access	506
dpmhyperv DeleteTemplate Command--Deletes a Template	507
CA IBM LPAR CLI Commands	508
dpmlpar cycle Command--Cycle a Logical Partition	509
dpmlpar delete Command--Delete a Logical Partition	513
dpmlpar getresources Command--Get LPAR Resources	515
dpmlpar image Command--Create an IBM AIX LPAR.....	517
dpmlpar imgjobcheck Command--Retrieve Status of IBM AIX Imaging Job	529
dpmlpar list Command--List HMC Resources.....	530
dpmlpar setresources Command--Add Memory Resources	533
dpmlpar setresources Command--Add Processor Resources	535
dpmlpar setresources Command--Subtract Memory Resources	538
dpmlpar setresources Command--Subtract Processor Resources	540
dpmlpar setresources Command--Update Memory Resources	543
dpmlpar setresources Command--Update Processor Resources.....	544
CA Microsoft Cluster Server CLI Commands	547
dpmmscs getclusterinfo Command--Get Information of a Cluster.....	547
dpmmscs getgroupresourcesinfo Command--Get Resource Information of a Cluster group	548
dpmmscs getnetworkinterfacesinfo Command--Get Network Interface Information	550
dpmmscs getnetworksinfo Command--Get Network Information	551
dpmmscs getnodesinfo Command--Get Cluster Nodes Information.....	552
dpmmscs getresourcegroupsinfo Command--Get Cluster Resource Groups Information	553
dpmmscs getresourcesdependenciesinfo Command--Get Dependency Information of Resources	554
dpmmscs getresourcetypesinfo Command--Get Cluster Resource Types Information.....	555
dpmmscs getservicesinfo Command--Get Cluster Services Information	556
dpmmscs movegroup Command--Move a Resource Group to another Node	557
dpmmscs pauseservice Command--Pause a Cluster Service	558
dpmmscs bringgrouponline Command--Bring a Resource Group Online	559
dpmmscs bringresourceonline Command--Bring a Resource Online	560
dpmmscs resumeservice Command--Resume a Cluster Service	562
dpmmscs startservice Command--Start a Cluster Service	563
dpmmscs stopservice Command--Stop a Cluster Service	564
dpmmscs takegroupoffline Command--Take a Resource Group Offline	565
dpmmscs takeresourceoffline Command--Take a Resource Offline.....	566
CA Solaris Zones CLI Commands.....	567
dpmzone associateproject Command--Associate a Task with a Project.....	567
dpmzone clonezone Command--Clone a Zone	568
dpmzone createandinstallzone Command--Create and Install a Zone	569

dpmzone createpool Command--Create Resource Pool.....	572
dpmzone createproject Command--Create a Project	574
dpmzone createzone Command--Create a Zone	575
dpmzone customcommand Command--Run a Custom Command.....	578
dpmzone deleteproject Command--Delete a Project	579
dpmzone deletezone Command--Delete a Zone	580
dpmzone installzone Command--Install a Zone	581
dpmzone movezone Command--Move a Zone	582
dpmzone rebootzone Command--Reboot a Zone	583
dpmzone renamezone Command--Rename a Zone	584
dpmzone setpoolscheduler Command--Set the Pool Scheduler	585
dpmzone setprocessorset Command--Set the Maximum Number of CPUs for a Processor Set.....	586
dpmzone showinterfaces Command--Show Network Interfaces	587
dpmzone showpools Command--Show Resource Pools	588
dpmzone showprojects Command--List the Properties of a Project	589
dpmzone showresourceset Command--Show Resource Sets	589
dpmzone showtasks Command--List the Tasks of a Project	590
dpmzone showzoneinterfaces Command--Show Network Interfaces of a Zone.....	591
dpmzone showzoneprocess Command--Show Processes of a Zone	592
dpmzone showzoneprojects Command--Show Projects of a Zone.....	592
dpmzone showzones Command--Show Zones	593
dpmzone startzone Command--Start a Zone.....	594
dpmzone stopzone Command--Stop a Zone.....	595
dpmzone uninstallzone Command--Uninstall a Zone	596
CA VMware vCenter Server CLI Commands	596
dpmvc addesxhost Command--Add an ESX Host	597
dpmvc addvmvdisk Command--Add Virtual Disk.....	599
dpmvc addvmvnic Command--Add Virtual NIC.....	601
dpmvc clone Command--Clone a VM.....	604
dpmvc createdatastore Command--Create a Datastore.....	612
dpmvc cycle Command--Cycle a VM	614
dpmvc datastore Command--Get Data Store Properties.....	618
dpmvc delete Command--Destroy a VM.....	620
dpmvc distributedswitch Command--Manage Virtual Distributed Switches.....	623
dpmvc entermaintenancemode Command--Set to the Maintenance Mode	628
dpmvc exitmaintenancemode Command--Exit the Maintenance Mode.....	630
dpmvc faulttolerance Command--Specify Fault Tolerant Operations	632
dpmvc getavailablescsidisks Command--Get Available SCSI Disks	634
dpmvc gethosthba Command--Get Host Bus Adapters	635
dpmvc getresources Command--Get VM Resources	636
dpmvc getshares Command--Get VM Shares	639
dpmvc getsnapshots Command--Get VM Snapshots.....	641

dpmvc imgjobcheck Command--Get Clone Job Status	643
dpmvc migrate Command--Migrate a VM	646
dpmvc removedatastore Command--Remove a Datastore	649
dpmvc removevmvmdisk Command--Remove Virtual Disk.....	650
dpmvc removevmvnic Command--Remove Virtual NIC.....	652
dpmvc setcpu Command-- Set CPUs for a VM	654
dpmvc setmemory Command-- Set memory size for a VM	656
dpmvc setresources Command--Set VM Resources	658
dpmvc setshares Command--Set VM Shares	661
dpmvc snapshot Command--Manage Snapshots.....	665
dpmvc templatetovm Command--Convert Template to VM.....	670
dpmvc vapp Command--Manage vApp.....	673
dpmvc virtualswitch Command--Manage Virtual Switches	677
dpmvc vmtotemplate Command--Convert VM to Template	681
dpmovf import Command--Import an OVF Package.....	684
General Shell Commands	685
caaipsecurity Command--Specifies Credentials for Web Service Security Check	686
dpmcmd run Command--Run Scripts.....	686
Verification Commands	687
dpmccm addacmserver Command--Add a Server Instance	687
dpmccm addcompliancejob Command--Create a Compliance Job	689
dpmccm addremediationjob Command--Add a CA Configuration Automation Remediation Job	690
dpmccm compare Command--Compare Systems.....	691
dpmccm createsnapshot Command--Create a Snapshot	692
dpmccm deleteacmserver Command--Delete a CCA Server.....	693
dpmccm deletecompliancejob Command--Delete a Compliance Job	694
dpmccm deletesnapshot Command--Delete a Snapshot.....	695
dpmccm detect Command--Detect Server Configuration Changes	696
dpmccm getaccessprofiles Command--Get Access Profiles.....	697
dpmccm getacmagentinfo Command--Get Agent Information	698
dpmccm getallcompliancejobs Command--Get All Compliance Jobs from CCA Server.....	699
dpmccm getallcomponents Command--Get Server Components	700
dpmccm getallremediationjobs Command--Get All CA Configuration Automation Remediation Jobs.....	701
dpmccm getallremediationprofiles Command--Get All CA Configuration Automation Remediation Profiles	702
dpmccm getallrulegroups Command--Get All Compliance Rule Groups from CCA Server.....	703
dpmccm getallsnapshot Command--Get all Snapshots on a Server	704
dpmccm getallsources Command--Get Servers.....	705
dpmccm getallsystems Command--Get Server Names from CA Configuration Automation Server	706
dpmccm getcompliancejobbyname Command--Get a Compliance Job By Job Name	707
dpmccm getcompliancejobhistory Command--Get Compliance Job History for a Specific Job	707
dpmccm getcurrentactivity Command--Get Server Activity	708

dpmccm getmanagementprofiles Command--Get Management Profiles.....	709
dpmccm getremediationjobhistory Command--Get CA Configuration Automation Remediation Job History.....	710
dpmccm getremediationjobinfo Command--Get CA Configuration Automation Remediation Job Details	711
dpmccm getremediationprofile Command--Get CA Configuration Automation Remediation Profile	712
dpmccm getsystemaccessprofile Command--Get Access Profile	713
dpmccm getsystemdefaultsource Command--Get Assigned Source	714
dpmccm getsystemmanagementprofile Command--Get System Management Profile.....	715
dpmccm redoremmediationjob Command--Redo a CA Configuration Automation Remediation Job.....	716
dpmccm removesystemdefaultsource Command--Remove Assigned Sources.....	716
dpmccm runacmdiscovery Command--Run Discovery	717
dpmccm runcompliancejob Command--Run a Compliance Job	718
dpmccm runremediationjob Command--Run a CA Configuration Automation Remediation Job.....	719
dpmccm runsystemmanagementprofile Command--Run a Management Profile.....	720
dpmccm setacmdefaultmanagementprofile Command--Set the Default Management Profile	722
dpmccm setdefaultacmserver Command--Set the Default CA Configuration Automation Server	723
dpmccm setsnapshotttype Command--Set Snapshot Type	724
dpmccm setsystemaccessprofile Command--Set Access Profile	725
dpmccm setsystemdefaultsource Command--Add a Server Source.....	726
dpmccm setsystemmanagementprofile Command--Set System Management Profile.....	727
dpmccm syncacmserver Command--Synchronize a CA Configuration Automation Server.....	728
dpmccm testsystemagent Command--Ping Agent.....	729
dpmccm undoremmediationjob Command--Undo a CA Configuration Automation Remediation Job.....	730
dpmccm updatecompliancejob Command--Update a Compliance Job.....	731
dpmccm updateremediationjob Command--Update a CA Configuration Automation Remediation Job	732
dpmccm validateacminfo Command--Validate CCA Server Credentials.....	733
Discovery Commands.....	733
dpmcda cancelnetworkscan Command--Cancel a Network Discovery.....	734
dpmcda deletenetwork Command--Delete a Network.....	734
dpmcda discoverhost Command--Discover a Host	735
dpmcda discovernetwork Command--Discover a Network	737
Collection Engine Commands.....	738
dpmce ceconfig -configuremetric Command--Configure Metric	738
dpmce ceconfig -configuremetricfilter Command--Configure Metric Filter	741
dpmce ceconfig -disablemetric Command--Disable a Metric	743
dpmce ceconfig -disablemetricforoverall Command--Disable and Exclude a Metric From Overall Calculation.....	744
dpmce ceconfig -enablemetric Command--Enable a Metric	745
dpmce ceconfig -enablemetricforoverall Command--Enable and Include a Metric for Overall Calculation.....	746
dpmce ceconfig -getglobal Command--Retrieve Data Center Intervals	747

dpmce ceconfig -getnode Command--Retrieve Host Collection Intervals	748
dpmce ceconfig -getmetriclist Command--Retrieve Metric List	749
dpmce ceconfig -refreshmetriclist Command--Retry Metric Collection	750
dpmce ceconfig -setalias Command--Set Alias	751
dpmce ceconfig -setglobal Command--Set Data Center Intervals	753
dpmce ceconfig -setnode Command--Set Intervals for a Host	755
dpmce ceconfig -updatemetric Command--Update a Metric	756
Event Commands	758
dpmeventmanager create_event Command--Create an Event	758
dpmeventmanager get_events Command--Retrieve Events	759
Help Desk Commands	762
dpmhd addcomment Command--Update a Ticket with a Comment	762
dpmhd close Command--Close a Help Desk Ticket	763
dpmhd getattribute Command--Get an Attribute Value of a Ticket	764
dpmhd getstatus Command--Get the Status of a Ticket	765
dpmhd open Command--Open a Help Desk Ticket	765
dpmhd setattribute Command--Set an Attribute of a Ticket	767
dpmhd setstatus Command--Set the Status of a Ticket	768
Solaris Imaging Commands	769
dpmimg image Command--Deploy a Solaris Image	770
dpmimg imgjobcheck Command--Get Solaris Imaging Job Status	774
dpmimg refresh_images Command--Refresh Boot and Install Images from Solaris JumpStart Servers	776
Network Installation Management Commands	777
dpmnim image Command—Deploy an IBM AIX Image Using Individual Resources	777
dpmnim image Command—Deploy an IBM AIX Image Using a Resource Group	781
dpmnim image Command—Deploy an IBM AIX Image to an Existing LPAR Using Individual Resources	785
dpmnim image Command—Deploy an IBM AIX Image to an Existing LPAR Using a Resource Group	790
dpmnim imgjobcheck Command—Retrieve NIM IBM AIX Imaging Job Status	793
Object Model Utility Commands	795
caaipaomwsclient Command--Enumerating Objects	795
caaipaomwsclient Command--Updating Objects	796
caaipaomwsclient Command--Deleting Objects	797
caaipaomwsclient Command--List Associated Objects	798
caaipaomwsclient Command--Summarize Class Use	799
caaipaomwsclient Command--Summarize Object Associations	799
caaipaomwsclient Command--List Available Associations	800
caaipaomwsclient Command--Summarize Indications	800
caaipaomwsclient Command--Display Class Information	801
caaipaomwsclient Command--Display Propagated State	801
caaipaomwsclient Command--Display Detailed Status of a Web Service	802
Policy Commands	803
dmpolicy createdefaultactions Command--Create Default Actions	803

dmpolicy exportaction Command--Export Actions to a File	804
dmpolicy exportrule Command--Export Rules to a File	805
dmpolicy findbestmachine Command--Find the Best Server	806
dmpolicy findcandidatemachine Command--Find the Candidate Servers	807
dmpolicy getruleexeccount Command--Retrieve rule execution count	809
dmpolicy importaction Command--Import Actions from a File	810
dmpolicy importrule Command--Import Rules from a File	811
dmpolicy resetruleexeccount Command--Reset Retries to Run a Rule	812
dmpolicy runaction Command--Run an Action	814
dmpolicy setruleexeclimit Command--Set Limits for a Rule	816
Policy Configuration	818
caismutility Command -- Add Functionalities to Policy Configuration UI	819
CA Process Automation Commands	821
dpmipamadapter checkstatus Command--Check Status	822
dpmipamadapter getform Command--Get Form	823
dpmipamadapter listforms Command--List Forms	824
dpmipamadapter startprocess Command--Start Process	825
Remote Monitoring Commands	826
rmonwatch add Command--Add Rows to a MIB Table	826
rmonwatch del Command--Delete Rows from a MIB Table	828
rmonwatch getTable Command--View rmonWbem MIB Tables	830
rmonwatch getVersion Command--View Version Information	832
Advanced SNMP Parameters	833
Reservation Manager Commands	835
dpmrm cancel Command--Cancel a Reservation	835
dpmrm checkreservation and updatereservation Commands	836
dpmrm createfromtemplate Command--Create Reservation	837
dpmrm createsoftwaregroup Command	839
dpmrm extend Command--Extend a Reservation	841
dpmrm importvirtualmachines Command	843
dpmrm listsoftwaregroups Command	844
dpmrm listtemplates Command--Display List Reservation Templates	846
Resource Manager Commands	847
dpmresourcemgr addbaremetal Command--Add a Physical Computer	847
dpmresourcemgr addcomputertonetwork Command--Add Server to Network	848
dpmresourcemgr addipaddresspool Command--Add IP Address Pools	849
dpmresourcemgr addmachinestosome Command--Add Servers to a Service	851
dpmresourcemgr addtocontainer Command--Add Content to a Container	852
dpmresourcemgr createcontainer Command--Create a Container	853
dpmresourcemgr createdefaultservice Command--Create Default Service	854
dpmresourcemgr createipnetwork Command--Create Network	855
dpmresourcemgr createresourcegroup Command--Create Resource Group	857

dpmresourcemgr createresourcetemplate Command--Create Server Group Template	858
dpmresourcemgr createservice Command--Create a Service	859
dpmresourcemgr deletebaremetal Command--Delete a Physical Computer	861
dpmresourcemgr deletecontainer Command--Delete a Container	862
dpmresourcemgr deleteipnetworks Command--Delete a Network	863
dpmresourcemgr deletesystem Command--Delete a System	864
dpmresourcemgr getdiscoverednetworks Command--Get Discovered Networks	865
dpmresourcemgr getipaddresspools Command--List IP Address Pools	866
dpmresourcemgr getipnetworks Command--List Networks.....	867
dpmresourcemgr getserverlist Command--Get a List of Servers.....	868
dpmresourcemgr getservicelist Command--Get List of Services	869
dpmresourcemgr getsubnetdefaults Command--List Subnet Defaults	870
dpmresourcemgr removecomputerfromipnetwork Command--Remove Server From Network	871
dpmresourcemgr removefromcontainer Command--Remove Content from a Container.....	872
dpmresourcemgr removeipaddresspool Command--Remove IP Address Pool.....	873
dpmresourcemgr removemachinesfromservice Command--Remove Servers From a Service	874
dpmresourcemgr removeresourcegroup Command--Remove Resource Group.....	875
dpmresourcemgr removeservice Command--Remove a Service.....	876
dpmresourcemgr removesourcetemplate Command--Remove Resource Template.....	877
dpmresourcemgr retrievesresourcegroup Command--Retrieve Resource Group	878
dpmresourcemgr retrievesourcetemplate Command--Retrieve Resource Template.....	879
dpmresourcemgr setcontainername Command--Set Container Name	880
dpmresourcemgr setcontainerservice Command--Set Container to a Service.....	881
dpmresourcemgr setmanagementip Command--Set the Management IP for a Host.....	883
dpmresourcemgr setmanagementstatus Command--Set the Management Status for a Host	884
dpmresourcemgr setserviceproperties Command--Set Service Properties.....	885
dpmresourcemgr updateresourcegroup Command--Update Resource Group	886
dpmresourcemgr updateresourcetemplate Command--Update Resource Template.....	887
dpmresourcemgr updateservice Command--Update a Service	889
dpmresourcemgr updateipnetwork Command--Update a Network	890
dpmresourcemgr updatesubnetdefaults Command--Update Network Defaults	891
dpmresourcemgr updateipaddresspool Command--Update IP Address Pool.....	893
Service Response Monitor CLI Commands.....	894
svcwatch add adir Command--Add an Active Directory Test.....	895
svcwatch add custom Command--Add a Custom Test.....	900
svcwatch add dhcp Command--Add a DHCP Test.....	905
svcwatch add dns Command--Add a DNS Test	910
svcwatch add fileio Command--Add a File IO Test.....	915
svcwatch add ftp Command--Add an FTP Test	920
svcwatch add http https Command--Add an HTTP or HTTPS Test	925
svcwatch add imap Command--Add an IMAP Test	931
svcwatch add ldap Command--Add an LDAP Test	936

svcwatch add mapi Command--Add a MAPI Test	941
svcwatch add nis Command--Add a NIS Test	947
svcwatch add nntp Command--Add an NNTP Test	952
svcwatch add ping Command--Add a PING Test	957
svcwatch add pop3 Command--Add a POP3 Test	962
svcwatch add rtemail Command--Add a Round Trip Email Test	967
svcwatch add smtp Command--Add an SMTP Test	973
svcwatch add snmp Command--Add an SNMP Test	978
svcwatch add sql Command--Add an SQL Test	984
svcwatch add tcpconnect Command--Add a TCP Connect Test.....	990
svcwatch add tftp Command--Add a TFTP Test	995
svcwatch add vuser Command--Add a Virtual User Test	1000
svcwatch delete Command--Delete a Test	1005
svcwatch list Command--View Test Information	1008
svcwatch setstatus Command--Change the Status of a Test	1011
svcwatch version Command--View SRM Version Information	1014
Software Delivery Commands	1016
dpmsd cancelimage Command--Cancel a Pending Image	1017
dpmsd deliver Command--Deliver a Package	1018
dpmsd deploypkggrp Command--Deliver a Package Group	1022
dpmsd deployprocgrp Command--Deliver a Procedure Group	1025
dpmsd image Command--Deploy an Image	1028
dpmsd imgjobcheck Command--Get OS Imaging Job Status	1032
dpmsd installSDAgent--Install Software Delivery Agent	1033
dpmsd list Command--List Software Packages	1036
dpmsd listsd Command--List Installed SD Adapters	1037
dpmsd pkgjobcheck Command--Get Package Deployment Job Status.....	1038
dpmsd setmanagedstatus Command--Set Management Status of Package	1040
Storage Provisioning Manager Command.....	1042
cadpmspm attach Command--Attach Storage to Host	1042
cadpmspm clone storage--Clone Storage	1044
cadpmspm deprovision Command--Deprovision Existing Storage	1045
cadpmspm discover Command--Discover Storage Objects	1047
cadpmspm jobstatus Command--Get Job Status	1049
cadpmspm move Command--Move Storage	1050
cadpmspm provision Command--Provision New Storage.....	1052
cadpmspm resize Command--Resize Existing Storage	1054
Log Files	1055
Log File Settings.....	1056

Chapter 4: Command Line Utilities

1057

dpmkpdb Command--Import or Export Performance Data	1058
dpmutil Syntax	1059
dpmutil -amq Command--Configure the ActiveMQ Message Broker	1060
dpmutil -ccm Command--Configure CA Configuration Automation	1061
dpmutil -dhcp_unix Command--Configure DHCP for UNIX	1062
dpmutil ec2-owner Command--Configure EC2 Owner	1063
dpmutil ec2-private-keypair Command--Configure Private Key Pair for EC2	1065
dpmutil -emc-smi Command--Configure EMC SMI-S Provider Server	1066
dpmutil ecs Command--Configure Windows Scheduler	1066
dpmutil -hd Command--Configure Help Desk	1067
dpmutil -hmc Command--Configure HMC or IVM Server	1068
dpmutil -hyperv Command--Configure Hyper-V Server	1069
dpmutil -itpam-cfg-eem Command--Configure CA Process Automation/CA EEM	1070
dpmutil -itpamserver Command--Configure CA Process Automation	1071
dpmutil -js Command--Configure JumpStart	1072
dpmutil mgmtadb Command--Configure Management Database	1072
dpmutil netapp Command--Configure NetApp DataFabric Manager	1073
dpmutil -nim Command--Configure NIM Server	1074
dpmutil -perfadb Command--Configure the Performance Database	1074
dpmutil -proxy Command--Configure Proxy Server	1075
dpmutil remoteucsaim Command--Configure Remote Cisco Unified Computing System AIM	1076
dpmutil remotevcaim Command--Configure Remote VC AIM	1077
dpmutil -sc Command--Configure the Service Controller	1077
dpmutil -scvmmserver Command--Configure SCVMM Server	1078
dpmutil -sdscale Command--Configure the Software Delivery Scalability Server	1079
dpmutil -sdserver Command--Configure Software Delivery Server	1080
dpmutil -snmp Command--Configure SNMP	1081
dpmutil -superuser Command--Configure SuperUser Credentials	1083
dpmutil -systemcredential Command--Configure External System Credentials	1084
dpmutil -sysuser Command--Configure Local Credentials Used at Service Startup	1085
dpmutil ucs Command--Configure Cisco Unified Computing System Server	1086
dpmutil upm Command--Configure the CA Patch Manager DB	1087
dpmutil vcdefault Command--Configure Default VMware vCenter Administrator User	1087
dpmutil vcserver Command--Configure VMware vCenter	1088
dpmutil -vioserver Command--Configure VIO Server	1089

Index

1091

Chapter 1: Introduction

This section contains the following topics:

[Scope of This Guide](#) (see page 17)

[Related Publications](#) (see page 17)

[Conventions](#) (see page 18)

Scope of This Guide

This guide describes AutoShell, CA Server Automation CLI commands, and performance metrics. A glossary explains specific terminology used in virtualization technologies.

AutoShell is the central command-line interface of CA Server Automation that you can use to automate complex recurring and management tasks. AutoShell provides a standalone JavaScript implementation for object orientation, XML, and regular expression processing. AutoShell uses an out-of-the-box version of the Mozilla Spidermonkey JavaScript interpreter that also provides JavaScript functionality to the Mozilla Firefox web browser. The interpreter allows you to use JavaScript syntax directly in the AutoShell, for example, for displaying a directory listing.

This guide provides you all AutoShell commands, examples, and best practices for this user interface.

Related Publications

The CA Bookshelf provides the following CA Server Automation publications:

Administration Guide

Describes product architecture, troubleshooting, concepts, and configuration tasks for administrators.

Installation Guide

Describes installation prerequisites, best practices, and procedures for CA Server Automation.

Reference Guide

Provides detailed information about AutoShell, CLI scripting commands, and log files.

Performance Metrics Reference

Describes the performance metrics that are available for monitoring the systems performance of the supported platforms.

CA Process Automation Connector Reference Guide

Provides detailed information about CA Process Automation connectors and use cases.

Online Help

Provides information to help you complete tasks using the CA Server Automation user interface.

Reservation Manager Help

Provides information to help users and administrators complete tasks using the Reservation Manager user interface.

Release Notes

Provides information about new and changed features and product implementation information including operating system support, system requirements, and how to contact Technical Support.

Service Response Monitoring User Guide

Provides installation and configuration details of SRM.

SystemEDGE User Guide

Provides end-user information about the SystemEDGE agent.

SystemEDGE Release Notes

Provides information about new and changed features and agent implementation information including operating system support, system requirements, and how to contact Technical Support.

To view PDF guides, download and install Adobe Reader from the Adobe website if it is not already installed on your computer.

Conventions

This guide uses the following conventions:

Case-Sensitivity

All names of classes, commands, directives, environment parameters, functions, and properties mentioned in this guide are case-sensitive and you must spell them exactly as shown. System command and environment variable names *may* be case-sensitive, depending on your operating system's requirements.

Cross-References

References to information in other guides or in other sections in this guide appear in the following format:

Guide Name

Indicates the name of another guide.

"Chapter Name"

Indicates the name of a chapter in this or another guide.

Synonyms

Terms such as attribute, object, object identifier (OID) are synonymous to the term 'variable' in this document.

Syntax

Syntax and user input use the following form:

Italic

Indicates a variable name or placeholder for which you must supply an actual value.

{a|b}

Indicates a choice of mandatory operands, a or b.

[] or [[]]

Indicates optional operands.

Syntax Example

The following example uses these conventions:

```
modify -t ZONE [-m zoneserver] -p psetname {-min mincpu|-max maxcpu} pset -session ssh
```

The operands *-min* and *-max* are mandatory, but you can only use one of them depending on what you want to define, the minimum number of CPUs in the processor set or the maximum number. The operand *-m* is not required for this command to function. All other parts of the command must be entered as shown.

Installation Path

Install_Path used in path statements indicates the directory in which CA Server Automation or components of CA Server Automation are installed.

Defaults:

- Windows x86: C:\Program Files\CA
- Windows x64: C:\CA, C:\Program Files (x86)\CA, or C:\Program Files\CA
- UNIX, Linux: /opt/CA

Chapter 2: AutoShell

AutoShell is a command line and scripting environment that you can use to automate complex recurring and management tasks.

This chapter details AutoShell and provides you many examples and best practices of this user interface.

This section contains the following topics:

[About AutoShell](#) (see page 21)

[Manager Shell and Client Shell](#) (see page 23)

[Valid AutoShell User](#) (see page 23)

[Accessing AutoShell](#) (see page 24)

[Common Information Model \(CIM\) Objects](#) (see page 24)

[Invoking AutoShell](#) (see page 28)

[Stringification](#) (see page 33)

[Writing Data to the Console \(stdout\)](#) (see page 34)

[Reading Data from the Keyboard \(stdin\)](#) (see page 35)

[Accessing the Operating System](#) (see page 36)

[Running Scripts](#) (see page 37)

[AutoShell Core Reference](#) (see page 46)

[AutoShell Loadable Modules' Command Reference](#) (see page 160)

About AutoShell

AutoShell is a combination of a scripting language and a command line shell. It is based on the standardized scripting language ECMA-Script, most commonly referred to as JavaScript. While JavaScript is mostly known as a scripting language that is used on web pages, it does not need to run in a browser. It is a standalone scripting language implementing support for object orientation, XML and regular expression processing. AutoShell uses an out-of-the-box version of the Mozilla Spidermonkey JavaScript interpreter which also provides JavaScript functionality to the Mozilla Firefox web browser. This allows you to use JavaScript syntax directly in the AutoShell, for example, for displaying a directory listing.

Typically, strings must be surrounded by quotes, special characters must be escaped, or function calls must be enclosed in parentheses. Entering instructions in this manner is not efficient, so AutoShell implements a unique command translation layer on top of the JavaScript language that accepts typical shell-type commands with mandatory and optional arguments. These commands do not require strings to be placed in quotes and translated into the underlying JavaScript syntax.

Key AutoShell features include the following:

- Operating system access to run child processes and capture their output.
- Parallel remote script execution to permit efficient processing of tasks that cannot be run centrally.
- Remote file system access for advanced provisioning by copying files directly between computers.
- Support for common Application Programming Interface (API) mechanisms typically used by management APIs, such as library calls, Java, and COM, to permit access from scripts to these APIs.
- Remote registry access (Windows specific).
- Ability to push and install its own client shell and arbitrary install packages to remote Windows systems.

Note: For information about JavaScript language references and examples, see the website <https://developer.mozilla.org/en/JavaScript>. AutoShell uses JavaScript version 1.7.

More Information

[Invoking AutoShell](#) (see page 28)

[Stringification](#) (see page 33)

[AutoShell Core Reference](#) (see page 46)

[Manager Shell and Client Shell](#) (see page 23)

[Running Scripts](#) (see page 37)

Manager Shell and Client Shell

AutoShell consists of two parts: the manager shell and the client shell. To enter commands or start the execution of scripts, work with the manager shell to perform local tasks on the manager system. The client shell is considered to be part of the AutoShell remote execution infrastructure. To enable remote execution and remote file system access, the client shell must be installed on all managed remote nodes. Other than that you do not need to use the client shell. The manager shell invokes the client shell for remote operations through secure SSH connections. The term AutoShell always refers to the manager shell for this reason.

Manager Shell

Lets you interactively enter commands, start scripts and perform local tasks on the manager computer.

Client Shell

Enables remote execution and remote file system access using Secure Shell (SSH) connections, and must be installed on all managed server remote nodes.

AutoShell can be invoked in interactive mode or scripting (batch) mode.

Valid AutoShell User

During the CA Server Automation installation, you define a CA Embedded Entitlements Manager (CA EEM) user identity and password in the Native Security User Information screen of the installation wizard. The credentials are stored in the CA EEM database. The user is assigned to the CA Server Automation administrator group and can be used to log in to the CA Server Automation User Interface and AutoShell manager.

If CA Server Automation components that use CA EEM are installed on a local or remote system, the AutoShell manager always validates the login credentials against the CA EEM data. If not, the AutoShell manager validates the login credentials against Windows authentication.

Accessing AutoShell

You can access AutoShell from the CA Server Automation server.

To access AutoShell

1. Open the Windows Explorer and navigate to the following directory:
C:\Program Files\CA\SC\AutoShellClient
2. Double-click caaipaomautoshellclient.exe.
The AutoShell Command Prompt window opens.
3. Enter user name and password of a valid CA Server Automation user.
The AutoShell command prompt appears.

More Information

[Valid AutoShell User](#) (see page 23)

Common Information Model (CIM) Objects

CA Server Automation uses a CIM-based object model to store information about managed objects. Managed objects can be retrieved, queried, and manipulated as native objects inside the AutoShell scripting environment. For CIM detail, see the CIM specification published by the Distributed Management Task Force (DMTF): <http://www.dmtf.org/standards/cim/>. For DMTF CIM documentation format and feature specifications, see http://www.dmtf.org/standards/published_documents/DSP0202_1.0.0.pdf.

To enable the optional AutoShell CIM feature during installation, select the AutoShell CIM ALM (AutoShell Loadable Module).

Important! AutoShell provides full write access to the CA Server Automation object store. Using improper AutoShell commands or scripts can invalidate the integrity of this store, causing malfunction of other CA Server Automation components. Always perform read-only operations on the object store from within AutoShell.

Retrieve CIM Classes

You can retrieve CIM classes in AutoShell.

To retrieve DMTF CIM classes, use the following command:

```
get-autoShellClasses CIM_*
```


To retrieve CA-defined CIM extension classes, use the following command:
`get-autoShellClasses CA_*`

Retrieve Antecedents and Dependents From CIM Objects

You can use `getAntecedents()` and `getDependents()` methods to retrieve CIM object antecedents and dependents.

To retrieve CIM object antecedents, call the `getAntecedents()` method:

```
// Retrieve all computer systems
arr=CA_ComputerSystem.getInstances()
// Continue working with the first computer system returned
cs=arr[0]
// retrieve the antecedents
arrAnt = cs.getAntecedents()
for(i=0; i<arrAnt.length; i++)
{
    show-cimObject arrAnt[i]
}
```

To retrieve antecedents instances of a CIM class, call `getAntecedents()` method:

```
os = cs.getAntecedents("CIM_OperatingSystem");
show-cimObject os
```

To retrieve CIM object dependents, call the `getDependents()` method:

```
// Retrieve all computer systems,
arr=CA_ComputerSystem.getInstances()
// Continue working with the first computer system returned
cs=arr[0]
// retrieve the dependents
arrAnt = cs. getDependents()
for(i=0; i<arrAnt.length; i++)
{
    show-cimObject arrAnt[i]
}
```

To retrieve dependents instances of a CIM class, call the `getDependents()` method:

```
fs = cs. getDependents("CIM_FileSystem");
show-cimObject fs
```

Retrieve CIM Objects

Each CIM class has a static `getInstances()` method that returns an array of class objects.

To retrieve all systems managed by CA Server Automation, use the following command:

```
aObjs = CA_ComputerSystem.getInstances()  
? "Query returned", oObjs.length
```

To retrieve a subset of systems managed by CA Server Automation, use one of the following filter expressions:

```
aObjs = CA_ComputerSystem.getInstances("ElementName='Everest.local'")
```

or

```
aObjs = CA_ComputerSystem.getInstances("ElementName like 'Ev*')")
```

To retrieve system names and descriptions, use the following command:

```
aObjs = CA_ComputerSystem.getInstances();  
for(i=0; i<aObjs.length; i++)  
    ? aObjs[i].getElementName(),aObjs[i].getDescription()
```

To retrieve CIM object properties and values, use the following command:

```
aObjs = CA_ComputerSystem.getInstances("ElementName='Debian1'")  
show-cimObject aObjs[0]
```

Example: Output for show-cimObject command

```
CA_ComputerSystem
=====
Caption: null
Description: null
ElementName: Debian1
Facets: 44
InstallDate: null
Name: 47332da1-e782-0308-d614-c2e99de989bc
OperationalStatus: undefined
StatusDescriptions: undefined
Status: null
HealthState: 5
PropagatedHealthState: null
IsPropagated: true
PrimaryStatus: null
DetailedStatus: null
OperatingStatus: null
CommunicationStatus: null
ManagementStatus: null
EnabledState: null
OtherEnabledState: null
RequestedState: null
EnabledDefault: null
TimeOfLastStateChange: null
AvailableRequestedStates: undefined
TransitioningToState: null
CreationClassName: CA_ComputerSystem
NameFormat: null
PrimaryOwnerName: null
PrimaryOwnerContact: null
Roles: undefined
OtherIdentifyingInfo: undefined
IdentifyingDescriptions: undefined
Dedicated: undefined
OtherDedicatedDescriptions: undefined
ResetCapability: null
PowerManagementCapabilities: undefined
HostName: 192.168.1.119
IpAddress: null
SerialNumber: null
Version: null
CpuCount: 1
TotalVisibleMemorySize: 256
PatchLevel: null
BiosVersion: null
Fans: null
Chassis: null
```

DrCapable: null
VendorId: null
MacAddress: null
ArchTypeUID: null
Available: null
LastUpdate: null
SnapshotType: null
TotalStorageCapacity: null
Model: null
CpuSpeed: null
SystemImageId: null
HardwareClassId: null

Invoking AutoShell

You can invoke AutoShell from a command-line interface by using the following command:

```
caaiapaautoshell.exe [-U username] [-P password] [-h host] [[-f] scriptfile]  
[scriptargs] [-e expr] [-i]
```

-U *username*

Specifies a valid CA Server Automation user. You can also set the user name by using the CAASUSER environment variable.

-P *password*

Specifies the password for that user. You can also set the password by using the CAASPASS environment variable.

-f *scriptfile* [*scriptargs*]

(Optional) Loads the specified script file with its optional script arguments. This parameter can be used multiple times.

-e *expr*

(Optional) Evaluates the specified expression.

-i

(Optional) Enters interactive mode after loading scripts.

-locale=*locale*

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

Examples

Run the hello.js script:

```
caaipaomautoshell.exe -U vaimuser -P topsecret -f hello.js
```

Calculate the value of 4*3:

```
caaipaomautoshell.exe -U vaimuser -P topsecret -e 4*3
```

Set the environment variables for username and password and calculate the value of $\sin(\pi/2)$:

```
set CAASUSER=vaimuser  
set CAASSPASS=topsecret  
caaipaomautoshell.exe -e Math.sin(Math.PI/2)
```

More Information

[Use Interactive Mode](#) (see page 30)

[Use Remote Interactive Mode](#) (see page 31)

[Batch Mode](#) (see page 31)

[Remote Batch Mode](#) (see page 32)

[Valid AutoShell User](#) (see page 23)

Use Interactive Mode

Use interactive mode to perform simple operations on a local or remote host or for testing small code fragments when you create scripted command sequences.

To use interactive mode

1. Run the following AutoShell executable in one of the following ways:

```
caaipaomautoshell.exe
```

AutoShell prompts you for a user name and password. AutoShell then either performs an authentication check or stores the credentials for future use, such as authenticating to remote clients. This decision depends on the type of product integration.

```
caaipaomautoshell.exe -U bob -P mypassword
```

You specify the user name and password by using the `-U` and `-P` switches.

2. (Optional) Specify credentials by setting the environment variables `CAASUSER` and `CAASSPASS` as follows:

```
set CAASUSER=bob
set CAASSPASS=mypassword
```

The AutoShell prompt appears after a successful logon.

```
asmgr: :->
```

asmgr

Specifies the local host name.

3. Enter any valid JavaScript expression at the prompt because AutoShell uses JavaScript as its underlying scripting language.

Example:

```
2+3 displays 5
```

AutoShell has an automatic result display that eliminates the need to use output commands at the command prompt for simple expressions. However, displaying the result of the expression in a script does require you to use an output command. See also the [run-local command](#) (see page 38) section.

4. Enter one of the following commands to terminate the AutoShell interactive session:

```
quit
exit
```

Use Remote Interactive Mode

You can start a remote interactive session to a managed system that has the AutoShell client installed. AutoShell uses SSH for communication between manager and client shells to connect to any SSH server system with user name and password as login credentials. In this case, the available commands are not AutoShell commands, but the commands of the SSH configured on the remote system.

To use remote interactive mode

1. Specify the remote host name with the `-h` switch:

```
caaipaomautoshell.exe -h ascll
```

The interactive command prompt displays and contains the name of the remote host specified. Commands entered at the prompt are executed in the client shell on the remote system. The result returns to the manager shell and displays in the interactive session.

Note: When scripts are specified for execution in an interactive remote session, they must reside on the remote system.

2. Enter *one* of the following options to terminate the AutoShell interactive session:

```
quit  
exit
```

Batch Mode

AutoShell scripts can be invoked from interactive mode, but they are typically run without any manual intervention.

To use batch mode

Invoke AutoShell with the `-f` switch and the path name of the script to run:

```
caaipaomautoshell.exe -f hello.js
```

AutoShell runs the script and terminates when script execution finishes. Interactive mode starts after the script finishes. Specify `-i` at the end of the command (order matters) to enter interactive mode after processing the script:

```
caaipaomautoshell.exe -f hello.js -i
```

Note: In these examples, add `-U` / `-P` switches with credentials or set the `CAASUSER` / `CAASSPASS` environment variables. If credentials are not defined, AutoShell prompts for them before starting script execution.

You can also pass arguments to the script:

```
caaipaomautoshell.exe -f hello.js 1 abc "x y z"
```

Passes 1, "abc" and "x y z" to the script. These arguments can be accessed through the arguments array which is a standard JavaScript mechanism.

Specify multiple scripts using multiple occurrences of the `-f` switch:

```
caaipaomautoshell.exe -f hello.js -f world.js
```

The scripts are executed in the order they are specified. When specifying arguments, each script receives its own set of arguments, for example;

```
caaipaomautoshell.exe -f hello.js 1 -f world.js 2
```

In this case the hello.js script receives the actual parameter 1 and world.js the parameter 2.

Expression evaluation is another variant. Instead of adding simple expressions to a script and execute them using `-f`, they can be specified on the command line using the `-e` switch:

```
caaipaomautoshell.exe -e Math.sqrt(2)
caaipaomautoshell.exe -e "for(i=1;i<11;i++)qout(i)"
```

Remote Batch Mode

You can run one or multiple scripts on a remote system with remote batch mode.

To use remote batch mode

Specify the host name using the `-h` switch before specifying the script files. For example, enter the following command:

```
caaipaomautoshell.exe -h ascl1 -f hello.js -f world.js
```

The script files reside on the manager system and are transferred to the remote client for execution.

Stringification

Stringification takes a sequence of characters and turns it into a JavaScript literal string. For example, the AutoShell `get-help` command that takes a filter string as an optional parameter:

```
get-help help
```

Because AutoShell uses an out-of-the-box JavaScript interpreter, any command entered at the prompt or in a script file, is translated into valid JavaScript syntax. The AutoShell command translation layer turns the command into a JavaScript function call:

```
help( "help" );
```

Literal strings in JavaScript must always be enclosed in quotes, therefore the command translation layer automatically places quotes around the string specified in the command. If quotes are found around a string, it is not stringified again.

Special characters inside literal JavaScript strings must be escaped. In particular, backslash and quotes inside quote delimited strings. Escaping is done by prefixing these special characters with a backslash. Consider the AutoShell `dir` command (1) that gets translated to a function call (2):

```
(1) dir c:\Program Files\CA\*.*
```

```
(2) ca.aip.direct.directory("c:\\Program Files\\CA\\*.*" , "", false, false, false, false);
```

Quotes may be required when dealing with strings that contain spaces and take multiple input parameters in one option. If a string with spaces inside is specified without quotes, the space is interpreted as an argument delimiter until all input parameters are matched. To prevent this behavior use quotes.

Instead of using a command without quotes (1), escape backslashes, and enclose the path in quotes so that AutoShell correctly identifies arguments (2):

```
(1) copy c:\Program Files\CA\*.* c:\temp
```

```
(2) copy "c:\\Program Files\\CA\\*.*" c:\temp
```

Note: Because the command translation layer does not stringify the path, special characters inside the string must be escaped manually.

To keep a string including spaces together and still get automatic escaping of special characters, you can also use the `_S()` macro:

```
copy _S(c:\Program Files\CA\*.*) c:\temp
```

When using AutoShell commands to automate tasks, parts of the commands must often accept input from JavaScript variables. In this case, prevent stringification. Otherwise, the variable name is turned into a literal string and the command does not produce the expected result. To prevent stringification of an expression, place the expression into parenthesis:

```
var topic = "help";  
get-help (topic)
```

Writing Data to the Console (stdout)

Scripts often display results or diagnostic messages. Use `"?"` and `"??"` commands to write any type of output to stdout, which specifies the console screen when no redirection is specified. To write output from within an expression to stdout, use the `qout()` and `qqout()` functions. The function `qout()` is equivalent to `"?"` and `qqout()` to `"??"`. Both commands and functions accept lists of arguments.

?, qout

Writes the string representation of each argument in a list to stdout. A single space is automatically placed between each displayed argument. The output is terminated with a linefeed character.

??, qqout

Writes the string representation of each argument in a list to stdout. A single space is automatically placed between each displayed argument.

Note: The `"?"` and `"??"` output commands can only be used at the beginning of a line.

Examples

The `"?"` command prints the specified value followed by a line feed:

```
? "Hello World"  
Hello World
```

The `"?"` command also accepts a comma-separated list of arguments. The displayed arguments are separated with a single space in the output:

```
? "Hello", "World"  
Hello World
```

The "??" command works like "?", but does not add the line feed at the end of the output. Several "??" commands can be used in a script to construct a single-line output:

```
?? "Hello"  
?? " "  
?? "World"
```

Both commands accept any data type and combinations thereof:

```
? "Today is", new Date  
? "The square root of 2 is", Math.sqrt(2)
```

Display the numbers from 1 to 10:

```
for(i=1;i<11;i++)qout(i);
```

Reading Data from the Keyboard (stdin)

If a script requires input while it is running, use one of the following commands:

wait

Waits for a key press and optionally stores the typed character in a variable.

accept

Reads a string from the keyboard (stdin) until enter is pressed:

Examples

Wait until an arbitrary key has been pressed:

```
wait "Press any key..."
```

Wait until one key has been pressed and store the resulting character in the variable ch.

```
wait ""Press any key" to ch
```

Read a string until an enter is encountered and store it to the variable str.

```
accept to str
```

Display the prompt message "Enter amount: ", read the input string until an enter is encountered, and store it to the variable str.

```
accept "Enter amount: " to str
```

Accessing the Operating System

Management shells often call operating system commands or other native utilities to accomplish system-related tasks. From the interactive AutoShell prompt or from within scripts, execute OS commands by prefixing them with an exclamation mark. The exclamation mark must always be the first non-whitespace character of the input line.

By default, AutoShell writes the output of OS commands directly to the AutoShell console. However, output that is written to the AutoShell console is not directly accessible by AutoShell for further processing. Applications that process child process output typically redirect the output to a file and read that file after the child process terminates. AutoShell lets you directly assign child process output to a JavaScript object. See the following examples.

You can launch any application from AutoShell using the OS access. When invoking OS commands, AutoShell always waits for the command to complete before returning to the command prompt or continuing script execution.

Examples

List the files in the current directory, print the version of the running OS, and start Notepad. These sample commands run without any AutoShell interaction. The first of the following commands uses the operating system `dir` command but not the built-in AutoShell `dir` command. AutoShell waits until the Notepad application is closed.

```
! dir
! ver
! notepad
```

Concatenate commands by using an ampersand "&" and execute them with a single child process:

```
! dir >t.txt & type t.txt
```

Using the OSRedirect Class

AutoShell uses the predefined `OSRedirect` class for output redirection when invoking child commands. To redirect output and capture the child process return code, create an `OSRedirect` object and pass it in the command invocation using the optional `-output` clause:

```
out = new OSRedirect();
! dir -output out
```

When you interactively execute this command in the AutoShell console, it does not display any information in the console. It redirects the output to the specified object. After completion, you can call a set of methods on the specified object to get information about the command execution:

```
? "Error occurred during command execution:", out.errorOccurred()
? "Did command complete:", out.hasCompleted()
? "Output produced by command:", out.output()
? "Command return code (errorlevel):", out.result()
```

The methods `output()` and `result()` return the required information. The `output()` method returns a string with the complete directory listing. You can use the standard JavaScript string functions to further process this output. For example, parse the lines into an array and display them line-by-line:

```
lines=out.output().split("\n");
for(i=0;i<lines.length;i++)qout(lines[i]);
```

If you want to invoke the `dir` operating system command with the `!` AutoShell command, allocate an `OSRedirect` object before issuing the `!` command. To save this step, you can also invoke OS commands using the `!!` command. The `!!` command creates an `OSRedirect` object and assigns it to the specified variable if it does not exist.

The following command outputs the current directory listing to the variable `out`, even if no `OSRedirect` object is explicitly allocated:

```
!! dir -output out
```

If you issue the `!!` command without the optional `-output` clause, it automatically writes the directory information to variable `$$stdout`:

```
!! dir
? $$stdout.output()
```

Running Scripts

This section covers the execution of scripts on the local host and on multiple remote hosts.

More Information

[Using the run-local Command](#) (see page 38)

[Using the run-remote Command](#) (see page 40)

Using the run-local Command

Use the run-local command to evaluate script expressions or to run script files locally. The run-local command allows passing of parameters to the script code.

To evaluate an expression, specify the expression, for example:

```
run-local 1+1
```

The expression parameter is stringified, so you can specify expressions with or without quotes.

To execute a script, specify the pathname of the script in the optional `-file` clause:

```
run-local -file scripts\primes.js
```

The run-local command with the `-file` option is the only way for a running script to invoke another script. Specifying the pathname of a JavaScript file to invoke the script only works at the interactive prompt. Pathname specification is not supported inside script files. If you want to execute scripts from scripts, consider the following example:

```
// Begin of script file: hello.js
function hello()
{
    ? "Hello World!"
}
// End of script file
```

Use the following command to invoke this script:

```
run-local -file hello.js
```

This command makes the `hello()` function available in the current scripting context, but it does not execute it, so nothing is visible on the console. After the run-local command returns, the calling script can invoke the newly defined function to print the expected greeting:

```
hello();
```

When invoking a script file, only code that is not contained in any surrounding scope (like a function) is executed during the run-local command runs. Changing the script to the following example, prints the greeting before the run-local command returns.

```
// Begin of script file: hello2.js
? "Hello World!"
// End of script file
```

To make the function available to the call and still execute the code, the script file can contain a call to the function itself:

```
// Begin of script file: hello3.js
hello();
function hello()
{
    ? "Hello World!"
}
// End of script file
```

Directly executable code in script files can appear anywhere outside function scopes. Placing executable code in front of the first function declaration, as in the previous example, is not necessary.

You can pass parameters in an expression or in a script being run with the `run-local` command using the optional `-with` clause. The parameters are accessible in the expression or script through the standard JavaScript arguments array:

```
// Begin of script file: args.js
var i, l;
l = arguments.length;
for(i=0; i < l; i++)
{
    ? arguments[i]
}
// End of script file
```

The script prints all the arguments that are passed to it. For example, the following command prints "1", "abc" and the current date and time.

```
run-local -file args.js -with 1, "abc", new Date()
```

More Information

[run-local Command--Execute a Script on the Local System \(Funlet\)](#) (see page 93)

[Accessing the Operating System](#) (see page 36)

[Using the OSRedirect Class](#) (see page 36)

Using the run-remote Command

The run-remote command executes expressions or script files on one or more remote AutoShell client systems. For remote execution to work, the remote target node must have the AutoShell client installed and be configured.

The run-remote command supports specifying expressions, scripts and arguments in the same way as the run-local command. Script files must reside on the manager system and transfer to the target systems for execution. In addition, specify the remote clients with the on clause. Specifying a server name is the simplest way of invoking the run-remote command:

```
run-remote "1+2" on "asc11"
```

Note: Stringification is not supported for the host name.

You can specify multiple targets for remote execution by using a list:

```
run-remote "1+2" on "asc11", "asc12"
```

Using run-remote this way does not get any information about the remote job that was executed. The job runs or fails without any further indication. Remote commands execute asynchronously which means run-remote commands return while the actual remote command is still executing.

To wait until the remote command processes or until an error occurs, invoke run-remote with the `-wait` option:

```
run-remote "1+2" on "asc11" -wait
```

More control over remote execution provides the predefined RemoteTarget AutoShell class.

More Information

[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)
[Using the RemoteTarget Class](#) (see page 41)

Using the RemoteTarget Class

Use the run-remote command with the RemoteTarget class in the following ways:

- Work with RemoteTarget objects returned by the run-remote command.
- Create RemoteTarget objects and pass them to the run-remote command.

The run-remote command always returns an array of RemoteTarget objects. The length of this array is equal to the number of remote targets that have been specified in the on clause. The array elements directly correspond to the remote targets by position:

```
arrRT = run-remote -file hello.js on "ascl1", "ascl2"
```

The run-remote command returns an array with two RemoteTarget objects:

- arrRT[0] contains a RemoteTarget object that corresponds to the execution of the script hello.js on the remote host named ascl1.
- arrRT[1] corresponds to the execution of the same script on ascl2.

RemoteTarget objects make status information of the script available, regarding its execution process, output, or result. The following RemoteTarget methods retrieve this status information:

hasCompleted()

Indicates if the command processing has been completed (true) or not (false). If run-remote is invoked without the optional -wait switch, command execution runs asynchronously. Call this method to verify if a command completes and if a result is available. The method hasCompleted() returns true when the command successfully completes or false when an error occurs that prevents further processing.

Return values: true or false

errorOccurred()

Indicates whether the command finishes successfully and the output and result can be verified (false), or if an error occurs (true).

Return values: true or false

output()

Returns a string that contains the captured output produced by running the specified script on the corresponding remote system.

Return value: *string*

result()

Returns an XML-encoded value of the specified script that has been executed on the corresponding remote system.

Return value: *XML*

See the [AutoShell Classes](#) (see page 135) section for additional methods of the RemoteTarget class.

Examples

Assuming the previously introduced hello.js script produces the "Hello World!" greeting, the output() methods for both RemoteTarget objects return this string. The following script puts this type of processing in a larger context:

```
var t;
var arrRT;
var i, l;

arrRT = run-remote -file hello.js on "ascl1", "ascl2"
l = arrRT.length;

// Wait for completion, could use -wait clause above as well
// but this way we can bail out on our own timeout
t = 0;

while(t<60)
{
  for(i=0; i<l; i++)
  {
    if(!arrRT[i].hasCompleted())
      break;
  }
  // Break if all invocations finished
  if(i==l)
    break;
  sleep(500);
  t++;
}
```

```

for(i=0; i<l; i++)
{
  if(arrRT[i].hasCompleted())
  {
    if(arrRT[i].errorOccurred())
    {
      ? "Error occurred on ", arrRT[i].getHostName()
    }
    else
    {
      ? arrRT[i].getHostName(),"returned output:"
      ? arrRT[i].output()
    }
  }
  else
  {
    arrRT[i].abort();
  }
}

```

Combining remote execution with OS command invocation enables access to OS services or data on remote systems. For example, the following code queries and returns the OS version string of the operating system running on server ascl1.

```

arrRT = run-remote ! ver on "ascl1" -wait
? arrRT[0].output()

```

AutoShell only displays the remote command output in the previous examples. Expressions or scripts processed remotely can also return a value:

```

arrRT = run-remote 2*21 on "ascl1" -wait

```

In this case the expression "2*21" is sent to server ascl1 for remote calculation. When the command completes, the result is available through the result() method of the corresponding RemoteTarget object.

The result() method does not return the value directly, but an XML representation of it:

```

? arrRT[0].result()
<value type="number">42</value>

```

To turn this XML string into a regular JavaScript value, use get-remoteResult():

```

v = get-remoteResult(arrRT[0])
? typeof v, v

```

AutoShell serialization also applies to remote scripts which return complex types like arrays or objects. Serialization enables the transfer of large amounts of structured data between remote servers without losing structural data. The following example shows a remote execution that returns an array consisting of three elements:

```
arrRT = run-remote [1,"abc",new Date()] on "ascl1" -wait
v = get-remoteResult(arrRT[0])
? typeof v, v
```

Another way of working with RemoteTarget objects is to create them before invoking the actual remote command. A RemoteTarget object is constructed by specifying the name of the target host in the object creation:

```
var rt1 = new RemoteTarget("ascl1");
var rt2 = new RemoteTarget("ascl2");
```

Explicitly creating RemoteTarget objects is useful when more than one command or script is executed for the same remote host. In this case, the same RemoteTarget object can be reused for each run-remote. Otherwise, run-remote creates a RemoteTarget object for each invocation.

To use the RemoteTarget objects with run-remote, specify them instead of the host names:

```
run-remote -file hello.js on rt1, rt2 -wait
? rt1.output()
? rt2.output()
run-remote Math.log(Math.E) on rt1, rt2
? get-remoteResult(rt1)
? get-remoteResult(rt2)
```

The usage of RemoteTarget objects does not require to store the array returned by run-remote. The array only contains references to the RemoteTarget objects that were initially passed to the command. So the explicitly created RemoteTarget objects can be directly accessed to query results of the execution.

You can specify remote targets in a list like in the previous example or as an array.

```
arrRT = [new RemoteTarget("ascl1"), new RemoteTarget("ascl2")];
run-remote -file hello.js on arrRT
```

RemoteTarget objects let you reuse objects in subsequent run-remote command invocations, and also provide a way to manage remote client session creation.

When AutoShell executes run-remote, it connects to the specified remote system through SSH. By default, the SSH server on the remote system starts a new instance of the client AutoShell. The newly created AutoShell process evaluates the passed script and the process is destroyed after evaluation finishes. For multiple subsequent command invocation, multiple client AutoShell processes are created and destroyed on the remote system. Additionally, any context information created when running a script is lost and is not available to subsequent remote scripts.

The RemoteTarget class implements an easy to use mechanism to control the client AutoShell session creation on the remote host. Call createRemoteContext() on a RemoteTarget object to create a permanent client AutoShell session on the remote host for this RemoteTarget object:

```
var rt = new RemoteTarget("ascl1");
rt.createRemoteContext()
```

Subsequent run-remote invocations do not allocate a temporary client AutoShell session on the remote host, but rather use the permanent one created by createRemoteContext(). Additionally, any context information that is created when running a script is available to any remote scripts executed later. The permanent client AutoShell session on the remote host is automatically destroyed when the RemoteTarget object is collected. The session can also be explicitly destroyed by calling destroyRemoteContext() on the corresponding object.

The effect of using createRemoteContext() can easily be verified using the following example:

```
var rt = new RemoteTarget("ascl1");
rt.createRemoteContext()
run-remote X=5 on rt -wait
run-remote X on rt -wait
? get-remoteResult(rt)
rt.destroyRemoteContext()
run-remote X=5 on rt -wait
run-remote X on rt -wait
? get-remoteResult(rt)
```

The second command invocation returns 5 because the variable X created in the first call still exists in the permanent client AutoShell session. When the remote client session is destroyed, the two remaining invocations trigger the creation of two temporary client AutoShell sessions. The second command returns an empty result, because the variable X was created in another session was already destroyed.

More Information

[RemoteTarget Class](#) (see page 146)

AutoShell Core Reference

This section describes the commands, functions, and classes which are specific to the AutoShell core. Further commands that belong to platform-specific extensions are described in the [ALM Command Reference](#) (see page 160) section.

AutoShell commands are grouped into cmdlets and funclets:

Cmdlets

Specify command definitions that must start with the first non-whitespace character in a line. Because of this restriction they can only be used standalone and not as part of a broader JavaScript expression. Specifically, they cannot be used as rvalues (right-hand side operand of an assignment operator). ? is an example of an AutoShell cmdlet.

Funclets

Maintain verbose command-like syntax with optional clauses, stringification, and so on, and can also return values. Often funclets, are used like cmdlets, that is, standalone in a single line. They can however return a value that can be processed as part of a broader expression.

Commands, Functions, and Classes According to Categories

AutoShell provides the following CA Server Automation-specific commands (cmdlets, funclets), functions, and classes. Use these commands to create appropriate AutoShell applications to manage your virtual environment.

Cmdlet

!, !!, ?, ??, accept, arrdump, cat, cd, DBG_PROMPT, DBG_PROMPT_LINE, exit, external, get-autoShellClassInfo, get-webServiceInfo, help, mkdir, objdump, pwd, quit, set alternate to, set console, set result disp, start-java, type, wait, weak external

Funclet

add, ASSERT, associate, chdir, clone, copy, cp, create, custom, del, dir, get-help, get-remoteResult, install, ls, modify, move, mv, new-SSHSession, new-ZONESession, new-comObject, new-webService, push-client, push-winRemote, push-winRemote, PW_GET, query-service, reboot, rebuild, rem-client, remove, ren, rename, reset, rm, rmdir, run-SSHCommand, run-SSHShell, run-ZONECommand, run-client, run-local, run-remote, run-winRemote, show, shutdown, start, start-service, status, stop, stop-service, uninstall

Functions

base64Decode, base64Encode, curDir, delete, disable, enable, gete, memoRead, memoWrit, platform, pute, qout, qqout, regCreateKey, regCreateSubkeys, regDeleteKey, regDeleteVal, regGetKeyValues, regGetSubKey, regGetVal, regIsKey, regIsVal, reSetKeyValues, regSetVal, setProcExitCode, shellType, typeName

Classes

OSRedirect, RemoteTarget

AutoShell commands, functions, and classes can be categorized according to the following criteria:

Arrays

arrdump

AutoShell deployment

exit, quit, push-client, rem-client, run-client, setProcExitCode, shellType

Classes and objects

new-comObject, get-autoShellClassInfo, get-remoteResult, objdump, OSRedirect, RemoteTarget, run-local, run-remote, typeName

Command execution

!, !!, get-remoteResult, OSRedirect, push-winRemote, RemoteTarget, run-local, run-remote, run-SSHCommand, run-SSHShell, run-winRemote, run-ZONECommand

Datatypes

typeName

Encryption

base64Decode, base64Encode

Environment variables

gete, pute

Files and directories

cat, cd, chdir, copy, cp, curDir, del, dir, ls, mkdir, mv, pwd, ren, rm, rmdir, type

Help

get-help, help

Input and output

?, ??, accept, DBG_PROMPT, DBG_PROMPT_LINE, PW_GET, qout, qqout, set alternate to, set console, set result disp, wait

Mathematical expressions

ASSERT

Miscellaneous

external, start-java, weak external

Platforms

push-winRemote, run-LPARCommand, run-winRemote, run-ZONECommand,
platform

Registry

regCreateKey, regCreateSubkeys, regDeleteKey, regDeleteVal, regGetKeyValues,
regGetSubKey, regGetVal, regIsKey, regIsVal, reSetKeyValues, regSetVal

Remote access

get-remoteResult, push-winRemote, RemoteTarget, run-remote,
run-SSHCommand, run-SSHShell, run-winRemote

Services

get-webServiceInfo, new-webService, query-service, start-service, stop-service

Sessions

new-SSHSession, new-ZONESession

Solaris Zones

associate, clone, create, custom, delete, disable, enable, install, modify, move,
reboot, rename, show, start, status, stop, uninstall

Strings

accept, PW_GET, memoRead, memoWrit, typeName

General AutoShell Commands

This section details the general AutoShell core commands (cmdlets, funclets) in alphabetic order. These general commands are not platform-specific to LPAR or Solaris Zones. AutoShell defines a series of commands that simplify invoking the corresponding tasks.

The following regular JavaScript function call obtains a detailed directory listing of C:\:

```
ca.aip.direct.directory("C:\\*.*" , "", false, false, false, true)
```

The corresponding AutoShell command looks like the following:

```
dir C:\\*.* -l
```

Examples

Lists the content of the current directory in the console:

```
dir *.*
```

Assign the directory listing to an array:

```
arr = dir *.* -retval
```

!! Command--Invoke Command or Child Process and Auto-capture Output (Cmdlet)

Run an operating system command or start an application as a child process. A user-specified `OSRedirect` object or a default system variable (`$$stdout`) captures child process output automatically.

The command has the following syntax:

```
!! cmd [-output osRedirect]
```

cmd

Command to execute or application to run. Multiple commands can be concatenated using a `&&` sequence. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

osRedirect

(Optional) Existing `OSRedirect` object to receive the output of the child process and capture it in the specified variable.

Default: `$$stdout`

Examples

Get DOS memory information about Windows OS with output redirected to a variable, `out`, that is not initialized before the call:

```
!! mem -output out
// Output line by line
aLines = out.output().split(String.fromCharCode(10));
arrdump aLines
```

Perform a DNS lookup and automatically capture the output to `$$stdout`:

```
!! nslookup ca.com
? $$stdout.output()
```

See also:

[OSRedirect Class](#) (see page 136)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

! Command--Invoke Command or Child Process (Cmdlet)

Run an operating system command or start an application as child process. The output produced by the child process is written to the AutoShell console by default. Optionally, redirect the output to an AutoShell OSRedirect object.

The command has the following syntax:

```
! cmd [-output osRedirect]
```

cmd

Specifies a command to execute or application to run. Multiple commands can be concatenated using a && sequence. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code into parenthesis.

osRedirect

(Optional) Existing OSRedirect object to receive the output of the child process.

Default: ""

Examples

Display the version string of a Windows OS:

```
! ver
```

Display the version string of a Windows OS display current directory:

```
! ver && cd
```

Ping a system on the network and capture output:

```
out=new OSRedirect();  
! ping 192.168.0.100 -output out  
if(out.result()==0)  
{  
    ? out.output();  
}  
else  
{  
    ? "Ping failed"  
}
```

See also:

[OSRedirect Class](#) (see page 136)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

?? Command--Write Output in a List to stdout (Cmdlet)

Writes the string representation of each argument in a list to stdout. A single space is automatically placed between each displayed argument. Subsequent output calls using `?`, `??`, `qout()` or `qqout()` place their output immediately after the original output, so `??` or `qqout()` are typically used to construct output lines using several invocations.

The command has the following syntax:

```
?? [x]
```

x

(Optional) List of arguments to output.

Examples

Print three column headers with separating spaces:

```
?? "Col1"  
?? " "  
?? "Col2"  
?? " "  
? "Col3"
```

The third label is printed using the `?` command to move the cursor to the beginning of the next line.

See also:

[OSRedirect Class](#) (see page 136)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

? Command--Write Output to stdout (Cmdlet)

This command (`cmdlet`) writes the string representation of each argument in a list to stdout. A single space is automatically placed between each argument. The `set console` and `set alternate` commands can redirect or suppress the output. The output terminates with a linefeed character.

The command has the following syntax:

```
? [x]
```

x

(Optional) List of arguments to output. If the list is empty, only a linefeed character prints.

Examples

Print "Hello World!":

```
? "Hello World!"
```

Print a list of numbers:

```
? 1,2,3
```

Print an empty line:

```
?
```

Output the numbers from 1 to 10. The ? command can only be used at the beginning of a line, to output text from within an expression the corresponding `qout()` function must be used:

```
for(i=1;i<11;i++) qout(i);
```

See also:

[qout--Write Output Followed by Linefeed to stdout \(Function\)](#) (see page 117)

[qqout--Write Output to stdout \(Function\)](#) (see page 118)

[?? Command--Write Output in a List to stdout \(Cmdlet\)](#) (see page 52)

[set alternate to Command--Set Alternate Output File \(Cmdlet\)](#) (see page 105)

[set console Command--Suppress Console Output \(Cmdlet\)](#) (see page 106)

accept Command--Read String From stdin and Assign it to a Variable (Cmdlet)

Optionally prompts for input, reads characters until an enter is encountered and assigns the character string without the enter to a variable.

The command has the following syntax:

```
accept [prompt] to var [-hidden]
```

prompt

(Optional) Specifies a character string to display.

var

Name of variable to assign the string holding the read characters to.

-hidden

(Optional) Masks the input string to enter a password.

Examples

Read a string and assign it to the variable name:

```
accept "Enter your name: " to name
```

Mask input to enter a password:

```
accept "Enter password:" to password -hidden
```

See also:

[wait Command--Wait for a Key Press \(Cmdlet\)](#) (see page 110)

[PW_GET Command--Input Hidden Data \(Funclet\)](#) (see page 85)

arrdump Command--Display an Array (Cmdlet)

This command lists all elements of a one-dimensional array.

The command has the following syntax:

```
arrdump arr
```

arr

Defines the array to display.

Example

Display an array:

```
a = [1,2,"a", true, new Date()];  
arrdump a
```

See also:

[objdump Command--Display Enumerable Properties of an Object \(Cmdlet\)](#) (see page 80)

ASSERT Command--Compare Mathematical Expressions (Funclet)

This command (funclet) checks if the statement or assertion that consists of two mathematical expressions and an operation is true or false. The mathematical expressions can be composed of the following elements:

- Functions
- Operations
- Decimal fractions
- Numerical values in decimal, octal, or hexadecimal format

If the statement or assertion is false, AutoShell raises a customizable error message.

To write numbers in octal format, precede the value with a 0, for example, 0123. To write numbers in hexadecimal format, precede the value with a 0x, for example, 0x123.

The command has the following syntax:

```
ASSERT x relation v [-msg msg]
```

x

Defines a mathematical expression that is compared against the expression *v*.

relation

Specifies the relation used for the comparison. Options include the following:

- eq (equal)
- ne (not equal)
- gt (greater than)
- lt (less than)
- ge (greater or equal)
- le (less or equal)

v

Defines a mathematical expression.

-msg *msg*

(Optional) Defines an alternative error message instead of the default error message raised when the comparison returns false.

Default: null

Examples

Show that the statement `6>7` is wrong:

```
ASSERT 6 gt 7 -msg "This is wrong."  
This is wrong.
```

Verifies if the octal value `0123` is equal to `0x9e`:

```
ASSERT 0123 eq 0x9e  
Failed: 0123==158, actual=83
```

Check if `PI` is greater than `4.5*sin(PI/2)`:

```
ASSERT Math.PI gt 4.5*Math.sin(Math.PI/2)  
Failed: Math.PI>4.5, actual=3.141592653589793
```

cat, type Commands--Display Text Files (Cmdlets)

These commands (cmdlets) write the content of one or more text files to the standard output stream. Use with `set alternate` to concatenate multiple files.

The commands have the following syntax:

```
{cat|type} fileList
```

fileList

Defines a comma-separated list of absolute or relative file names display. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Examples

Display Windows boot options:

```
cat c:\boot.ini
```

Concatenate two files:

```
set alternate to output.txt  
cat input1.txt, input2.txt  
set alternate to
```

See also:

[memoRead--Read a Text File Into a String \(Function\)](#) (see page 114)

cd, chdir Commands--Change Directory (Cmdlet, Funclet)

These commands change the current directory on the local system. If *p* is supplied, it defines the new directory. If *p* is a relative path, the command uses the current path as a prefix for the directory.

The command has the following syntax:

```
cd [p] (Cmdlet)
chdir [p] (Funclet)
```

p

(Optional) Defines the directory to change to. The separation character on Windows can be a backslash (\) or a forward slash (/). For UNIX-like systems, it must be a forward slash. For cross-platform script compatibility, only use the forward slash in path specifications.

Default: ""

Examples

Change to root directory:

```
cd /
```

Change to directory c:/test

```
cd c:/test
```

Change to the parent directory:

```
cd ..
```

See also:

[mkdir Command--Create a Directory \(Cmdlet\)](#) (see page 72)

[pwd Command--Print Name of Working Directory \(Cmdlet\)](#) (see page 86)

[dir, ls Commands--Get File and Directory Information \(Funclets\)](#) (see page 64)

[curDir--Retrieves the Current Directory \(Function\)](#) (see page 112)

copy, cp Commands--Copy Files and Directories (Funclets)

These commands (funclets) copy files or directories to another location. Either the source or the target location must be on the local server. Operations between two remote systems are not supported. To specify a remote host, prefix the pathname with the hostname followed by two colons (::). The operation is performed through SFTP. Thus the remote system does not necessarily have the client AutoShell installed, an SSH server with SFTP access is sufficient. You can write the output to a file using the set alternate command.

The command has the following syntax:

```
{copy|cp} src trg [-silent] [-retval] [-R] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

src

Defines a search mask that specifies the directories or files to copy. The separation character on Windows can be a backslash (\) or a forward slash (/). For UNIX-like systems, it must be a forward slash. For cross-platform script compatibility, only use the forward slash in path specifications.

Search masks use a simplified regular expression syntax that is compatible with DOS wildcard matching:

- * matches any sequence of characters, including zero characters.
- ? matches exactly one character.
- [abc] matches exactly one character which is a, b, or c.
- [a-f] matches anything from a through f.
- [^a-f] matches anything *except* a through f.
- [-_] matches - or _; [^_] matches anything else. The dash "-" is not a special character when it occurs immediately after the opening bracket or after ^.
- [a^] matches an a or a ^. (The caret "^" is not a special character when it does *not* occur immediately after the opening bracket.
- *, \?, \[, \], \\ match the single characters *, ?, [,], \.

All other characters are not special characters and match themselves.

trg

Defines the path of the target location.

-silent

(Optional) Suppresses output to the active output stream.

-retval

(Optional) Creates an array representing the return value. The array contains information about new file names. Each element consists of a string representing the fully qualified name of one destination file.

-R

(Optional) Copies directories recursively.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the phrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

Examples

Copy file c:/test/test1.log to directory c:/test1

```
copy c:/test/test1.log c:/test1
```

Copy entire directory c:/test to directory c:/test1

```
cp c:/test c:/test1 -R
```

Copy entire directory c:/test to the remote host remotesys into directory c:/test1

```
cp c:/test1/ remotesys::c:/test1/ -R
```

See also:

[mv, ren Commands--Move Files and Directories \(Funclets\)](#) (see page 74)

DBG_PROMPT Command--Break Script Execution for Inspection (Cmdlet)

The DBG_PROMPT command (cmdlet) interrupts script execution and displays an interactive prompt to evaluate expressions in the current script context. Typically used for inspecting current variable values, the debug prompt reoccurs until an empty input line is entered.

The command has the following syntax:

```
DBG_PROMPT [prompt]
```

prompt

Specifies a string with optional prompt to display. When using several prompts, the prompt can be used to indicate the location that triggered the interruption.

Default: "Dbg> "

Examples

Break during the last iteration of the loop:

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT
        ? i
}
```

Break using a custom prompt:

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT "i==10> "
    ? i
}
```

See also:

[DBG_PROMPT_LINE Command--Break Script Execution for Inspection With Line Numbers \(Cmdlet\)](#) (see page 61)

DBG_PROMPT_LINE Command--Break Script Execution for Inspection With Line Numbers (Cmdlet)

The `DBG_PROMPT_LINE` command (cmdlet) interrupts script execution and displays an interactive prompt to evaluate expressions in the current script context. The prompt automatically includes the current line number. Typically used for inspecting current variable values, the debug prompt reoccurs until an empty input line is entered.

The command has the following syntax:

```
DBG_PROMPT_LINE [prompt]
```

prompt

Specifies a string with optional prompt to display. When using several prompts, the prompt can be used to indicate the location that triggered the interruption. If prompt is specified, it displays "prompt(n)> "

Default: "Dbg(n)> "

n

Displays the line number as input prompt.

Examples

Break during the last iteration of the loop:

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT_LINE
    ? i
}
```

Break using a custom prompt:

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT_LINE "i==10> "
        ? i
}
```

See also:

[DBG_PROMPT Command--Break Script Execution for Inspection \(Cmdlet\)](#) (see page 60)

del, rm Commands--Delete Files (Funclets)

These commands (funclets) delete files or directories. To specify a remote host, prefix the pathname with the hostname followed by two colons (::). The operation is performed through SFTP. Thus the remote system does not necessarily have the client AutoShell installed, an SSH server with SFTP access is sufficient.

The command has the following syntax:

```
{del|rm} src [-silent] [-retval] [-R] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

src

Defines the files or directories to delete.

-silent

(Optional) Suppresses output to the active output stream.

-retval

(Optional) Creates an array representing the return value. The array contains information about new file names. Each element consists of a string representing the fully qualified name of one destination file.

-R

(Optional) Copies directories recursively.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the phrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

Examples

Delete file c:/test/test1.log:

```
del c:/test/test1.log
```

Remove entire directory c:/test:

```
rm c:/test -R
```

Delete entire directory c:/test on the remote host remotesys:

```
del remotesys::c:/test1/ -R
```

dir, ls Commands--Get File and Directory Information (Funclets)

These commands (funclets) list information about the contents in the specified directory on a local or remote system. You can write the output to a file using the set alternate to command.

They have the following syntax:

```
{dir|ls} mask [-silent] [-retval] [-l] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

mask

Defines a string with a search mask optionally containing an absolute or relative directory specification. The separation character on Windows can be a backslash (\) or a forward slash (/). For UNIX-like systems, it must be a forward slash. For cross-platform script compatibility, only use the forward slash in path specifications.

Search masks use a simplified regular expression syntax that is compatible with DOS wildcard matching:

- * matches any sequence of characters, including zero characters.
- ? matches exactly one character.
- [abc] matches exactly one character which is a, b, or c.
- [a-f] matches anything from a through f.
- [^a-f] matches anything *except* a through f.
- [-_] matches - or _; [^-_] matches anything else. The dash "-" is not a special character when it occurs immediately after the opening bracket or after ^.
- [a^] matches an a or a ^. (The caret "^" is not a special character when it does *not* occur immediately after the opening bracket.
- *, \?, \[, \], \\ match the single characters *, ?, [,], \.

All other characters are not special characters and match themselves.

Default: ""

-silent

(Optional) Suppresses output to the active output stream.

-retval

(Optional) Creates an array representing the return value. The array contains the file information. Each element consists of a string representing the file name. If argument `-l` is specified, each element consists of an array containing following items:

- `item[0]` - string representing the file name
- `item[1]` - number representing the file size
- `item[2]` - date representing the last modification time
- `item[3]` - Boolean value representing the directory flag
- `item[4]` - string representing fully qualified file name including path

-l

(Optional) Use long listing format for output.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify `PW_GET()` with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the phrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

Examples

Get file names of the current directory:

```
dir
```

Get file names of directory c:\test:

```
dir c:/test
```

Get file names of directory c:\test and c:\test\more as content of the file test.list:

```
set alternate to test.list
dir c:\test
dir c:\test\more
set alternate to
```

List the directory contents of c:\temp on the remote server ascli1:

```
dir ascli1::c:/temp
```

Get file names of directory `c:/test/more` as an array for further processing and suppress any output to stdout:

```
set result disp off
aRet = dir c:/test/more -silent -retval
```

Get extended information for files of directory `c:/test/more` as an array for further processing:

```
aRet = dir c:/test/more -silent -retval -l
```

See also:

[cd, chdir Commands--Change Directory \(Cmdlet, Funclet\)](#) (see page 57)

[pwd Command--Print Name of Working Directory \(Cmdlet\)](#) (see page 86)

[set alternate to Command--Set Alternate Output File \(Cmdlet\)](#) (see page 105)

exit, quit Commands--Terminate an AutoShell Session (Cmdlets)

These commands (cmdlets) terminate the current AutoShell session. Issue the command either at the interactive command prompt or inside a script.

The command has the following syntax:

```
exit
quit
```

See also:

[setProcExitCode--Set AutoShell Exit Code \(Function\)](#) (see page 132)

external Command--Declare a Native External Function (Cmdlet)

Declares a native external function.

The command has the following syntax:

```
external decl
```

decl

Specifies a native external function.

get-autoShellClassInfo Command--Display Methods and Properties for Autoshell Native Classes (Cmdlet)

This command (cmdlet) displays methods and properties for AutoShell native classes. AutoShell defines a number of classes to extend the standard Spidermonkey/JavaScript scripting environment. Product-specific AutoShell loadable modules (ALMs) can register additional classes. These classes contain type information for their constructors, methods, variables and enumerations. The information can be displayed using the get-autoShellClassInfo cmdlet.

Note: This information is different from the help information. AutoShell generates the information solely based on internal class definitions.

The command has the following syntax:

```
get-autoShellClassInfo class [-super]
```

class

Specifies the AutoShell class for which to display information. Unquoted argument tokens are stringified automatically. Prevent automatic quoting for expressions by placing expression code in parentheses. An object name can be passed instead of a class name. In such a case, the get-autoShellClassInfo command retrieves the class name from the objects and attempts to query the class information. To prevent stringification, the object expression must be passed in parentheses.

Examples

Display details of RemoteTarget class:

```
get-autoShellClassInfo RemoteTarget
```

Display details of OSRedirect class from an object:

```
os = new OSRedirect();  
get-autoShellClassInfo (os)
```

See also:

[help Command--Request Help Information \(Cmdlet\)](#) (see page 71)

[get-help Command--Get Help Information \(Funclet\)](#) (see page 69)

get-help Command--Get Help Information (Funclet)

This command (funclet) displays help information for AutoShell commands, functions and classes and optionally redirects it to a file.

The command has the following syntax:

```
get-help [mask] [-output outFile]
```

mask

(Optional) Specifies the mask that is used to filter out the commands about which help information is needed. You can use the asterisk "*" and the question mark "?" as wildcards.

Default: "help"

outFile

(Optional) Specifies the absolute or relative path of the file to contain the help information. Any existing file with the same name and path is overwritten.

Default: ""

Example

Display help for the run-remote funclet:

```
help run-remote
```

Display help for the RemoteTarget class including all constructors and methods:

```
help RemoteTarget*
```

List all help topics and write them to a file:

```
help * -output help.txt
```

See also:

[get-autoShellClassInfo Command--Display Methods and Properties for AutoShell Native Classes \(Cmdlet\)](#) (see page 68)

[help Command--Request Help Information \(Cmdlet\)](#) (see page 71)

get-remoteResult Command--Get Result from a Remote Target (Funclet)

This funclet retrieves the result of a remote execution from a RemoteTarget object. The return value of a remote execution must be transferred over the network back to the system that originated the request. The result is serialized into an XML representation on the remote system and the RemoteTarget.result() method returns the result value in its serialized representation.

Turn the serialized representation into a regular JavaScript value using the get-remoteResult() funclet with the RemoteTarget object itself as a parameter. Typically there is no need for script to call result() directly. Serialization maintains type information, so when the remote node returns a number, get-remoteResult() returns a value of type number. If the remote system returns a Date object, it also becomes a Date on the originating system.

Return values from remote scripts are not limited to simple data types. Complex arrays or data-only objects can also be returned, allowing remote execution to transfer large amounts of data between servers. The only limitation is that the data structures must not contain any circular references.

The command has the following syntax:

```
get-remoteResult remoteTarget
```

The funclet returns a polymorphic value from the remote execution process. If the execution has not finished, the return value is of type undefined. If another error occurs, an exception is created.

remoteTarget

Specifies the name of the remote target system.

Example

Get serialized representation and actual value of a remote execution:

```
rt = new RemoteTarget("ascli1", "bob", "ca123456");
run-remote "new Date()" on rt -wait
? rt.result()           // XML string
v = get-remoteResult(rt) // Calls result() internally
? v
? typeName(v)          // Date
```

See also:

[RemoteTarget Class](#) (see page 146)

[RemoteTarget.output Method](#) (see page 156)

[RemoteTarget.result Method](#) (see page 158)

get-webServiceInfo Command--Display WSDL Info (Cmdlet)

The get-webServiceInfo displays WSDL information.

```
get-webServiceInfo w [-detail]
```

w

Specifies the name of the web service.

help Command--Request Help Information (Cmdlet)

This command (cmdlet) displays help information for AutoShell commands, functions and classes and optionally redirects it to a file.

The command has the following syntax:

```
help [mask] [-output outFile]
```

mask

(Optional) Specifies the mask that is used to filter out the commands about which help information is needed. You can use the asterisk "*" and the question mark "?" as wildcards.

Default: "help"

outFile

(Optional) Specifies the absolute or relative path of the file to contain the help information. Any existing file with the same name and path is overwritten.

Default: ""

Example

Display help for the run-remote funclet:

```
help run-remote
```

Display help for the RemoteTarget class including all constructors and methods:

```
help RemoteTarget*
```

List all help topics and write them to a file:

```
help * -output help.txt
```

See also:

[get-autoShellClassInfo Command--Display Methods and Properties for AutoShell Native Classes \(Cmdlet\)](#) (see page 68)

[get-help Command--Get Help Information \(Funclet\)](#) (see page 69)

mkdir Command--Create a Directory (Cmdlet)

This command creates a directory on the local or a remote system, if it does not exist.

The function automatically creates path elements leading to the specified directory location. To specify a remote host, prefix the pathname with the hostname followed by two colons (::). The operation is performed through SFTP. Thus the remote system does not necessarily have the client AutoShell installed, an SSH server with SFTP access is sufficient.

The command has the following syntax:

```
mkdir d [-silent] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

d

Defines the name of the new directory.

-silent

(Optional) The command suppresses any output to stdout.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the phrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

Examples

Create directory c:\temp\xyz:

```
mkdir c:\temp\xyz
```

Create a directory on the remote server ascl1:

```
mkdir ascl1:/home/alice/data
```

Create directory temp as subdirectory of current working directory, but suppress any output to stdout:

```
mkdir temp -silent
```

See also:

[cd, chdir Commands--Change Directory \(Cmdlet, Funclet\)](#) (see page 57)

[pwd Command--Print Name of Working Directory \(Cmdlet\)](#) (see page 86)

[dir, ls Commands--Get File and Directory Information \(Funclets\)](#) (see page 64)

[rmdir Command--Remove Directory \(Funclet\)](#) (see page 90)

mv, ren Commands--Move Files and Directories (Funclets)

These commands move files or directories to another location. Either the source or target location must be on the local server. Operations between two remote systems are not supported. To specify a remote host, prefix the pathname with the hostname followed by two colons (::). The operation is performed through SFTP. Thus the remote system does not necessarily have the client AutoShell installed, an SSH server with SFTP access is sufficient. You can write the output to a file using the set alternate command.

Moving or renaming is equivalent in this case.

The commands have the following syntax:

```
{mv|ren} src trg [-silent] [-retval] [-R] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

src

Defines a search mask that specifies the directories or files to move. The separation character on Windows can be a backslash (\) or a forward slash (/). For UNIX-like systems, it must be a forward slash. For cross-platform script compatibility, only use the forward slash in path specifications.

Search masks use a simplified regular expression syntax that is compatible with DOS wildcard matching:

- * matches any sequence of characters, including zero characters.
- ? matches exactly one character.
- [abc] matches exactly one character which is a, b, or c.
- [a-f] matches anything from a through f.
- [^a-f] matches anything *except* a through f.
- [-_] matches - or _; [^_] matches anything else. The dash "-" is not a special character when it occurs immediately after the opening bracket or after ^.
- [a^] matches an a or a ^. (The caret "^" is not a special character when it does *not* occur immediately after the opening bracket.
- *, \?, \[, \], \\ match the single characters *, ?, [,], \.

All other characters are not special characters and match themselves.

trg

Specifies the path of the target location.

-silent

(Optional) Suppresses output to the active output stream.

-retval

(Optional) Creates an array representing the return value. The array contains information about new file names. Each element consists of a string representing the fully qualified name of one destination file.

-R

(Optional) Copies directories recursively.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the phrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

Examples

Move file `c:/test/test1.log` to directory `c:/test1`:

```
mv c:/test/test1.log c:/test1
```

Move entire directory `c:/test` to directory `c:/test1`:

```
mv c:/test c:/test1 -R
```

Move entire directory `c:/test` to the remote host `remotesys` into the directory `c:/test1`:

```
mv c:/test1/ remotesys::c:/test1/ -R
```

See also:

[copy, cp Commands--Copy Files and Directories \(Funclets\)](#) (see page 58)

new-comObject Command--Instantiate a COM Object (Funclet)

COM is a Windows-specific standardized API that allows application to expose services for consumption by COM clients. COM client support allows access to a wide range of COM servers like Internet Explorer, Microsoft Office, or various management-related OS services, in particular WMI. The `new-comObject` command creates proxy client objects that make the services exposed by a COM server available for scripting.

The command has the following syntax:

```
new-comObject progid [-locale lcid]
```

progid

Specifies a string that identifies the COM object to instantiate. The string can either be the human readable program ID (PROGID) or the internal CLSID. When specifying a CLSID, it must be contained in curly brackets.

lcid

Specifies an integer that defines the desired locale for COM servers supporting multiple locales.

Default: 0

Return value

The `new-comObject` returns an object that represents a proxy between the AutoShell and the COM server. The object mirrors the methods and properties exposed by the COM server. For details about the public interface of a COM server, refer to the programmer documentation of that particular server. The exposed methods and properties are accessed using regular JavaScript syntax on the returned `AutoObject`.

Examples

Open a web page in Internet Explorer and query its URL:

```
o = new-comObject InternetExplorer.Application
// Make Internet Explorer visible
o.Visible = true;
o.navigate("www.ca.com");
// Get active document (another COM object)
doc = o.document
? doc.Url      // for example http://www.ca.com/us
```

Get the computer name using the scripting object:

```
o = new-comObject WScript.Network
? o.ComputerName
```

Retrieve some information about the operating system and logical drives using WMI:

```
loc=new-comObject WbemScripting.SWbemLocator
// Security is actually not required for localhost
loc.Security_.AuthenticationLevel=0; // wbemAuthenticationLevelDefault
loc.Security_.ImpersonationLevel=3; // wbemImpersonationLevelImpersonate
wmisrv=loc.connectServer("localhost", "root\cimv2");
items = wmisrv.ExecQuery("Select * from Win32_OperatingSystem");
// Returned collection has only one item,
// since there is just one OS running
os = items[0];
? "OS Type:", os.Caption;
? "Service Pack:", os.CSDVersion;
? "Computer name:", os.CSName;
? "FreePhysicalMemory:", os.FreePhysicalMemory;
? "RegisteredUser:", os.RegisteredUser;
? "TotalVisibleMemorySize:", os.TotalVisibleMemorySize;
? "Version:", os.Version;
? "WindowsDirectory:", os.WindowsDirectory;
// Now for the disks
items = wmisrv.ExecQuery("Select * from Win32_LogicalDisk");
// Iterate over returned disks
for(i in items)
{
    disk = items[i];
    ? "Description:", disk.Description;
    ? "DeviceID:", disk.DeviceID;
    ? "FileSystem:", disk.FileSystem;
    ? "FreeSpace:", disk.FreeSpace;
    ? "Name:", disk.Name;
    ? "Size:", disk.Size;
}
```

new-SSHSession Command--Create a new Secure SSH Session (Funclet)

The new-SSHSession command creates a session on an SSH server and returns a CASSHELL object that represents the session. This object must be saved to perform any future action on the session. This command is typically used with the run-SSHCommand or run-SSHShell commands.

The command has the following syntax:

```
new-SSHSession -host hostname [-user username] [-pass password] [-key key]  
[-prompt prompt][-port portnumber]
```

-host *hostname*

The name of the SSH server.

-user *username*

(Optional) Specifies the user name to use to log in to the remote node. A user name is required when either password or public key authentication is used. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

-prompt *prompt*

(Optional) Specifies the fixed portion at the end of the prompt string displayed by the SSH server to indicate it is ready for input (for example: ":::->" for AutoShell). This information is required to determine the completion of commands executed during the SSH session. Specify as many fixed characters as possible. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ":::->"

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

Return value

Returns a CASSHELL object. This object can be passed to the run-SSHCommand funcllet to execute commands or to run-SSHSession to run an interactive session on the SSH server.

Example

Create an SSH session, assuming the server has AutoShell configured as SSH shell for the specified account. Run a command and disconnect:

```
ssh=new-SSHSession -host client8 -user bob -pass xyz42 -prompt :::->
? run-SSHCommand -session ssh -command "! ver"
ssh.disconnect();
```

new-webService Command--Create a Web Service Object (Funclet)

This command creates a web service object.

The command has the following syntax:

```
new-webService WSDLURI [-endpoint endpoint] [-user username] [-pass password]
[-proxy proxyname] [-proxyUser username] [-proxyPass password]
```

WSDLURI

Specifies the WSDLURI.

-endpointt *endpoint*

(Optional) Specifies the end point.

Default: ""

-user *username*

(Optional) Specifies the user name to use to log in to the remote node. A user name is required when password authentication is used. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are stringified automatically. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. Unquoted argument tokens are stringified automatically. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-proxy *proxyname*

(Optional) Specifies the name of the proxy.

Default: ""

-proxyUser *username*

(Optional) Specifies the user name to use to log in to the proxy.

Default: \$\$User

-proxyPass *password*

(Optional) Specifies password to use to log in to the proxy.

Default: \$\$Pass

objdump Command--Display Enumerable Properties of an Object (Cmdlet)

This command (cmdlet) lists the enumerable properties of the specified object, their types, and optionally the current value.

The command has the following syntax:

```
objdump obj [-detail]
```

obj

Specifies the object to display.

-detail

(Optional) When this option is specified, the command also displays the current value of the listed properties.

Example

Display properties of an object:

```
o = new Object();  
o.num = 42;  
o.str = "Hello";  
objdump o
```

Display current values:

```
objdump o -detail
```

See also:

[arrdump Command--Display an Array \(Cmdlet\)](#) (see page 54)

push-client Command--Install the Autoshell Client on a Remote Windows System (Funclet)

This command installs the AutoShell client on a remote Windows system.

This command copies and executes on one or multiple remote Windows systems.

Available versions are:

- Single-target version (1)
- Multi-target version (2)

The command has the following syntax:

```
(1) push-client -host hostname [-localPath path] [-user username>] [-pass password]  
(2) push-client [-localPath path] [-user username>] [-pass password] on remoteTargets
```

The single-target command returns true on success.

The multi-target command returns an array of WinRemote objects. Each element of this array corresponds by position to the remote Windows node listed in the *remoteTargets* list. The WinRemote objects can verify whether errors occur and query the output and returned result if execution completes. If a remote node had been specified using an WinRemote object already, the array contains a reference to the original RemoteTarget object. If a target system is specified by a string, a new RemoteTarget object is created internally and assigned to the array.

-host *hostname*

(Optional) Specifies the name of the remote Windows host. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

-localPath *path*

(Optional) Specifies the path pointing to the AutoShell client to copy. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

-user *username*

(Optional) Specifies the user name to log in to the remote Windows server. The user must have rights to connect to the \$ADMIN share. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

on *remoteTargets*

Specifies a comma-separated list of remote systems to run the specified script on. The target nodes can be identified in two ways: By hostname/address or by WinRemote objects. Specifying target systems by name or address while using WinRemote objects offers more flexibility to control the execution process, for example, different credentials for different hosts. WinRemote objects and strings with hostname/address can be mixed in the remote target list. Elements of the target list are not automatically stringified, so when using literal strings for hostnames they must be placed in quotes.

Examples

To install the AutoShell client on host1:

```
push-client -host host1
```

To install the AutoShell client on host1 and host2:

```
x = push-client on "host1", "host2";
? x[1].hasCompleted();
? x[1].result();
? x[1].output();
```

See also:

[push-winRemote Command--Copy and Execute on Multiple Remote Windows Systems \(Funclet\)](#) (see page 83)

[rem-client Command--Remove the Autoshell Client from a Remote Windows System \(Funclet\)](#) (see page 88)

[run-client Command--Run the Autoshell Client on a Remote Windows System \(Funclet\)](#) (see page 92)

[run-winRemote Command--Execute a Command on Multiple Remote Windows Systems \(Funclet\)](#) (see page 103)

push-winRemote Command--Copy and Execute on Multiple Remote Windows Systems (Funclet)

This command copies and executes on one or multiple remote Windows systems.

Available versions are:

- Single-target version (1)
- Multi-target version (2)

The command has the following syntax:

```
(1) push-winRemote -host hostname -localPath path [-cmd cmd] [-user username] [-pass password]
```

```
(2) push-winRemote -localPath path [-cmd cmd] [-user username] [-pass password] on remoteTargets
```

The single target version returns the result code of the executed command.

The multi-target version returns an array of WinRemote objects. Each element of this array corresponds by position to the remote Windows node listed in the *remoteTargets* list. The WinRemote objects can verify whether errors occur and query the output and returned result if execution completes. If a remote node is specified using an WinRemote object, the array contains a reference to the original RemoteTarget object. If a target system is specified by a string, a new RemoteTarget object is created internally and assigned to the array.

-host *hostname*

Specifies the name of the remote Windows host. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

-localPath *path*

Specifies the path pointing to the AutoShell client to copy. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

-cmd *cmd*

(Optional) Specifies the command to execute on the remote Windows system. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

-user *username*

(Optional) Specifies the user name to log in to the remote Windows system. The user must have rights to connect to the \$ADMIN share. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions place the expression code into parentheses.

Default: \$\$Pass

on *remoteTargets*

Specifies a comma-separated list of remote systems to run the specified script on. The target nodes can be identified in two ways: By hostname/address or by WinRemote objects. Specifying target systems by name or address while using WinRemote objects offers more flexibility to control the execution process, for example, different credentials for different hosts. WinRemote objects and strings with hostname/address can be mixed in the remote target list. Elements of the target list are not automatically stringified, so when using literal strings for hostnames they must be placed into quotes.

Examples

To install myinstall.exe on host1:

```
push-winRemote -host host1 -localpath c:\installations\myinstall.exe -cmd
myinstall.exe
```

To install myinstall.exe on host1 and host 2 and verify the results on host1:

```
push-winRemote -cmd myinstall.exe -localpath c:\installations\myinstall.exe on
"host1", "host2";
? x[0].hasCompleted();
? x[0].result();
? x[0].output();
```

See also:

[push-client Command--Install the Autosshell Client on a Remote Windows System \(Funclet\)](#) (see page 81)

[rem-client Command--Remove the Autosshell Client from a Remote Windows System \(Funclet\)](#) (see page 88)

[run-client Command--Run the Autosshell Client on a Remote Windows System \(Funclet\)](#) (see page 92)

[run-winRemote Command--Execute a Command on Multiple Remote Windows Systems \(Funclet\)](#) (see page 103)

PW_GET Command--Input Hidden Data (Funclet)

This command (funclet) optionally prompts for input and reads characters until a enter is encountered. For each character typed, the command displays an asterisk (*) to hide the input.

The command has the following syntax:

```
PW_GET(prompt)
```

The command returns a string holding the characters entered in clear text.

prompt

String to be display as input prompt. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code into parenthesis.

Default: ""

Example

Use this command to hide passwords, if the password cannot be typed in clear text as part of a command:

```
run-remote Math.PI on "ascl1" -user alice -pass PW_GET(Enter password:)
```

See also:

[accept Command--Read String From stdin and Assign it to a Variable \(Cmdlet\)](#) (see page 53)

pwd Command--Print Name of Working Directory (Cmdlet)

This command prints out the name of the current working directory.

The command has the following syntax:

```
pwd
```

Example

When you have installed the AutoShell in its default location, the pwd command displays C:\Program Files\CA\SC\AutoShellManager.

query-service Command--Query the Status of a Windows Service (Funclet)

The query-service command queries the status of a Windows service on the local system or a remote system. The query-service command returns the status of the service as a string. Possible return values include:

- stopped
- starting
- stopping
- running
- continue pending
- pause pending
- paused On error
- query failed

The command has the following syntax:

```
query-service -svcName sName [-host hostname] [-user username] [-pass password]
```

-svcName *sName*

Specifies the name of the service.

-host *hostname*

(Optional) The name of the remote Windows system.

Default: ""

-user *username*

(Optional) The user name used to log in to the remote Windows system. Unquoted argument tokens are stringified automatically. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-pass *password*

The password used to log in to the remote Windows system. Unquoted argument tokens are stringified automatically. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

Examples

To query the local Apache service

```
? query-service -svcName Apache2.2
```

To query the Microsoft SQL Server service on a remote server REMOTESYS

```
? query-service -svcName MSSQLSERVER -host REMOTESYS -user admin -pass notsecret
```

See also:

[start-service Command--Start a Windows Service on a Remote or Local System \(Funclet\)](#)
(see page 107)

[stop-service Command--Stop a Windows Service on a Remote or Local System \(Funclet\)](#)
(see page 108)

rem-client Command--Remove the AutoShell Client from a Remote Windows System (Funclet)

The rem-client command removes the AutoShell client from a remote Windows system.

The command has the following syntax:

```
rem-client -host hostname [-user username] [-pass password]
```

The command returns true on success.

-host *hostname*

Specifies the remote Windows host. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

-user *username*

(Optional) Specifies the user name to log in to the remote Windows system. The user must have rights to connect to the \$ADMIN share. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use when log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

Example

To uninstall the AutoShell client on host1.

```
rem-client -host host1
```

See also:

[push-client Command--Install the Autoshell Client on a Remote Windows System \(Funclet\)](#) (see page 81)

[push-winRemote Command--Copy and Execute on Multiple Remote Windows Systems \(Funclet\)](#) (see page 83)

[run-client Command--Run the Autoshell Client on a Remote Windows System \(Funclet\)](#) (see page 92)

[run-winRemote Command--Execute a Command on Multiple Remote Windows Systems \(Funclet\)](#) (see page 103)

rmdir Command--Remove Directory (Funclet)

This command removes a directory. The specified directory must be empty and must not contain files or subdirectories. To specify a remote host, prefix the pathname with the hostname followed by two colons (::). The operation is performed through SFTP. Thus the remote system does not necessarily have the client AutoShell installed, an SSH server with SFTP access is sufficient.

The command has the following syntax:

```
rmdir d [-silent] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

d

Defines the name of the directory to remove. To specify a remote host, prefix the pathname with the hostname followed by two colons (::). The operation is performed through SFTP. Thus the remote system does not necessarily have the client AutoShell installed, an SSH server with SFTP access is sufficient. You can write the output to a file using the set alternate command.

-silent

(Optional) The command suppresses any output to stdout.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the phrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

Examples

Remove the directory c:\temp:

```
rmdir c:/temp
```

Remove the directory temp as subdirectory of current working directory, but suppress any output to stdout:

```
rmdir temp -silent
```

Remove a directory from the remote system ascl1:

```
rmdir ascl1:/home/alice/data
```

run-client Command--Run the AutoShell Client on a Remote Windows System (Funclet)

The run-client command (funclet) runs the AutoShell client on a remote Windows system. Input and output are redirected to the calling AutoShell.

The command has the following syntax:

```
run-client -host hostname [-user username>] [-pass password]
```

The command returns true or false.

-host *hostname*

Specifies the name of the remote Windows host. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

-user *username*

(Optional) Specifies the user name to log in to the remote Windows system. The user must have rights to connect to the \$ADMIN share. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

Example

To run the AutoShell client on host1.

```
run-client -host host1
```

run-local Command--Execute a Script on the Local System (Funclet)

The run-local funclet executes JavaScript code contained in a string or a disk file on the local machine in the current AutoShell session and returns the result. Code that is not contained in a surrounding function declaration is directly executed. Function declarations are processed so that the functions become available for later execution but are not called. While JavaScript file can be executed from the interactive command prompt by simply entering their pathname, run-local is the only way to execute external scripts from script files or to pass parameters to a script file when invoking it from the interactive command prompt.

The command has the following syntax:

```
run-local [script] [-file file] [-with args,...]
```

The return value depends on the specified script.

script

(Optional) Specifies a string containing the JavaScript code to execute. The funclet either executes a script from a string or from a file. Thus the *script* and *file* arguments are mutually exclusive. Unquoted argument tokens are stringified automatically. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-file *file*

(Optional) Specifies the absolute or relative path to the script file to execute. The funclet either executes a script from a string or from a file. Thus the *script* and *file* arguments are mutually exclusive. Unquoted argument tokens are stringified automatically. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-with *args*,...

Specifies a comma-separated list of actual arguments to pass to the script. The script code can access these arguments using the standard JavaScript arguments array.

Examples

Evaluate a simple expression:

```
run-local 1+2
```

Evaluate an expression from memory passing an argument:

```
run-local "arguments[0]*arguments[0]" -with 2
```

Write a script file and execute it:

```
s = "var i;"
s += "for(i=0;i<arguments.length;i++)"
s += "qout(arguments[i]);"
s += "true;"
memoWrit("script.js", s);
run-local -file script.js -with 1, "abc", new Date()
```

Note: The script code is only directly executed when it is not contained in a function definition. The script does not return the result (in this case true) using the return statement, but by specifying it in the last expression executed.

See also:

[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)

run-remote Command--Execute a Script on Remote Systems (Funclet)

The run-remote command executes JavaScript code contained in a string or in a disk file on one or more remote systems. For remote execution to work, the remote target node must have the client AutoShell installed and be properly configured. Script files have to reside on the manager system and transfer to the target systems for execution. By default, script execution is asynchronous, meaning that local processing is continuing before the remote execution completes.

The command has the following syntax:

```
run-remote [script] [-file file] [-wait] [-with args,...] [-user username]
[-pass password] [-key key phrase] [-port portnumber] on remoteTargets
```

The run-remote command returns an array of RemoteTarget objects representing the execution state of the script on the specified servers. Each element of this array corresponds by position to the remote node listed in the *remoteTargets* list. The RemoteTarget objects can verify several results, for example, whether errors occur, whether execution completes, and whether successful. If a remote node had been specified using an RemoteTarget object already, the array contains a reference to the original RemoteTarget object. If a target system is specified by a string, a new RemoteTarget object is created internally and assigned to the array.

script

(Optional) Specifies a string containing the JavaScript code to execute. The funclet either executes a script from a string or from a file. Thus the *script* and *file* arguments are mutually exclusive. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-file file

(Optional) Specifies the absolute or relative path to the script file to execute. The `funclet` either executes a script from a string or from a file. Thus the *script* and *file* arguments are mutually exclusive. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-wait

(Optional) By default `run-remote` will return after kicking off remote execution and remote processing is asynchronous. When the `-wait` option is specified, `run-remote` waits until all remote tasks either finish successfully or abort with an error.

-with args,...

(Optional) Specifies a comma-separated list of actual arguments to pass to the script. The script code can access these arguments using the standard JavaScript arguments array.

-user username

(Optional) Specifies the user name to use to log in to the remote node. A user name is required for password or public key authentication is used. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. The specified user name is only used for remote systems that are identified by *hostname* in the *remoteTargets* list. When specifying a host using `RemoteTarget` objects, each host has its individual user name set through the `RemoteTarget` object.

Default: \$\$User

-pass password

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify `PW_GET()` with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. The specified password is only used for remote systems that are identified by *hostname* in the *remoteTargets* list. When specifying a host using `RemoteTarget` objects, each host has its individual user name set through the `RemoteTarget` object.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

(Optional) Specifies the passphrase for a private key. If the key is not encrypted, the passphrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-port *portnumber*

(Optional) Specifies the port on which to connect to the target system. The specified port is only used for remote systems that are identified by *hostname* in the *remoteTargets* list. When specifying host using RemoteTarget objects, each host can have its individual port set through the RemoteTarget object.

Default: SSH standard port 22.

on *remoteTargets*

Specifies a comma-separated list of remote systems to run the specified script on. The target nodes can be identified in two ways: By hostname/address or by WinRemote objects. Specifying target systems by name or address while using WinRemote objects offers more flexibility to control the execution process, for example, different credentials for different hosts. WinRemote objects and strings with hostname/address can be mixed in the remote target list. Elements of the target list are not automatically stringified, so when using literal strings for hostnames they must be placed into quotes.

Examples

Evaluate a simple expression and wait until execution completed:

```
aRT = run-remote "1+2" on "ascli1" -wait
? "Error occurred:", aRT[0].errorOccurred()
? "Result:", get-remoteResult(aRT[0])
```


Evaluate a script from a file on two remote systems, wait until execution completed and display output ("Hello World!") and result (42):

```
s="? 'Hello World';42;"
// Create script file for illustration purpose only
// could also call:
// aRT=run-remote (s) on "ascli1", "ascli2"
memoWrit("hello.js", s);
aRT=run-remote -file hello.js on "ascli1", "ascli2"
for(i=0; i<aRT.length; i++)
{
    ? "Target:", aRT[i].getHostName()
    ? "Error occurred:", aRT[i].errorOccurred()
    ? "Output:", aRT[i].errorOccurred()
    ? "Result:", get-remoteResult(aRT[i])
}
}
```

Evaluate a script on two remote systems using RemoteTarget objects, specifying different credentials for the target systems:

```
rt1 = new RemoteTarget("ascli1", "alice", "casogood42");
rt2 = new RemoteTarget("ascli2", "bob", "!secret!7");
// Get OS version string from remote systems
aRT = run-remote "! ver" on rt1, rt2
// Perform wait ourselves
while(true)
{
    // If an error occurs hasCompleted() will
    // return true as well
    if(rt1.hasCompleted() && rt2.hasCompleted())
        break;
    sleep(500);
}
// We can either use the original RemoteTarget objects
// (see above) or the references in the returned array
// (below).
for(i=0; i<aRT.length; i++)
{
    ? "Target:", aRT[i].getHostName()
    ? "Error occurred:", aRT[i].errorOccurred()
    ? "Output:", aRT[i].errorOccurred()
}
}
```

See also:

[RemoteTarget Class](#) (see page 146)

[get-remoteResult Command--Get Result from a Remote Target \(Funclet\)](#) (see page 70)

[run-local Command--Execute a Script on the Local System \(Funclet\)](#) (see page 93)

run-SSHCommand Command--Run a Remote Command Through SSH (Funclet)

The run-SSHCommand executes a command within an existing SSH session or creates a session for command execution. The session is destroyed after the specified command is executed. Although AutoShell uses SSH as its primary remote communication conduit, the remote node for this command does not require a running client AutoShell running. The run-remote command is preferable for executing commands on remote nodes running the client AutoShell because it returns more detailed information about the command execution.

Commands can be executed on any system running an SSH server. The available commands and their options depend on the configured shell on the remote node.

The command has the following syntax:

```
run-SSHCommand [-host hostname] [-user username] [-pass password] [-key key phrase]  
[-prompt prompt] -command cmd [-port portnumber] [-session ssh]
```

-host *hostname*

The name of the SSH server.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies the password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the passphrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-prompt *prompt*

(Optional) Specifies the fixed portion at the end of the prompt string displayed by the SSH server to indicate it is ready for input (for example: "::->" for AutoShell). This information is required to determine the completion of commands executed during the SSH session. Specify as many fixed characters as possible. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: "::->"

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

-cmd *cmd*

(Optional) Specifies the command to execute on the remote Windows system. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: cmd

-session *ssh*

A CASSHELL object as returned by `new-SSHSession`. When passing an existing SSH session object, `run-SSHCommand` does not need to create an SSH session to execute the specified command. When executing multiple commands session negotiation and logon are only performed once typically resulting in improved performance over multiple session allocations. If this parameter is specified, values passed for *host*, *user*, *pass*, *key*, *phrase*, and *port* are ignored. This information is already specified when creating the session.

Default: null

Examples

Run a directory listing on a remote node using an SSH session explicitly specifying user name and password:

```
s=run-SSHCommand -host srv14 -user bob -pass xyz42 -prompt :> -command ls
```

Run a directory listing on a remote node using an SSH session with user name and password used for AutoShell login:

```
s = run-SSHCommand -host srv14 -prompt ":>" -command "ls"
```

Execute two commands in an explicitly allocated SSH session:

```
ssh=new-SSHSession -host client8 -user bob -pass xyz42 -prompt ::->  
? run-SSHCommand -session ssh -command "! ver"  
? run-SSHCommand -session ssh -command "ls *.* -l"  
ssh.disconnect();
```

See also:

[new-SSHSession Command--Create a new Secure SSH Session \(Funclet\)](#) (see page 78)
[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)

run-SSHShell Command--Run an Interactive SSH Command Line (Funclet)

The run-SSHCommand command accepts input lines, sending them to a remote system through SSH and displays the output from the remote node. The command line can be run within an existing SSH session or a new SSH session. The session is destroyed when the input ends. Although AutoShell uses SSH as its primary remote communication conduit, the remote node for this command does not require a running client AutoShell. Commands can be executed on any system running an SSH server. The available commands, their options and the way to end the input loop depend on the configured shell on the remote node. Typically the input loop can be terminated by entering "quit" or "exit".

The command has the following syntax:

```
run-SSHShell [-host hostname] [-user username] [-pass password] [-key key phrase]  
[-prompt prompt] [-port portnumber] [-session ssh]
```

-host *hostname*

The name of the SSH server.

-user *username*

(Optional) Specifies the user name to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies the password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-key *key*

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote node. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: ""

phrase

Specifies the passphrase for a private key. If the key is not encrypted, the passphrase is not required. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-prompt *prompt*

(Optional) Specifies the fixed portion at the end of the prompt string displayed by the SSH server to indicate it is ready for input (for example: "::->" for AutoShell). This information is required to determine the completion of commands executed during the SSH session. Specify as many fixed characters as possible. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: "::->"

-port *portnumber*

(Optional) Port on which to connect to the target system.

Default: SSH standard port 22.

-session *ssh*

Specifies a CASSHELL object as returned by new-SSHSession. When passing an existing SSH session object run-SSHShell does not need to create an SSH session for execution of the commands from the command line. If this parameter is specified, values passed for *host*, *user*, *pass*, *key*, *phrase*, and *port* are ignored. This information is specified when creating the session.

Default: null

Examples

Run an interactive command line shell on a host using an SSH session with password authentication:

```
run-SSHShell -host svr14 -user bob -pass xyz42 -prompt :>
```

Explicitly allocate SSH session, run a command first and then drop into the command line input loop:

```
ssh=new-SSHSession -host client8 -user bob -pass xyz42 -prompt ::->
run-SSHCommand -session ssh -command "cd /"
run-SSHShell -session ssh
ssh.disconnect();
```

See also:

[new-SSHSession Command--Create a new Secure SSH Session \(Funclet\)](#) (see page 78)

[run-SSHCommand Command--Run a Remote Command Through SSH \(Funclet\)](#) (see page 98)

run-winRemote Command--Execute a Command on Multiple Remote Windows Systems (Funclet)

The run-winRemote command executes a command on one or multiple remote Windows systems. The command comes in two flavors:

- Single-target version
- Multi-target version

The single-target version redirects input and output to the calling AutoShell.

The multi-target version of run-winRemote is asynchronous. See the sample code on how to retrieve results.

The command has the following syntax:

```
(1) run-winRemote -host hostname [-cmd cmd] [-user username] [-pass password]
[-workDir workDir]
```

```
(2) run-winRemote [-cmd cmd] [-user username] [-pass password] [-workDir workDir] on
remoteTargets
```

The single targeted version of run-winRemote returns the result code of the executed command.

The multi-target version returns an array of WinRemote objects. Each element of this array corresponds by position to the remote Windows node listed in the *remoteTargets* list. The WinRemote objects can verify several results, for example, if the execution completes, if errors occur, or if the execution finishes successfully. If a remote node is specified using an WinRemote object, the array contains a reference to the original RemoteTarget object. If a target system is specified by a string, a new RemoteTarget object is created internally and assigned to the array.

-host *hostname*

(Optional) Specifies the name of the remote Windows host. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

-cmd *cmd*

(Optional) Specifies the command to execute on the remote Windows system. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses. This parameter is only required when creating an SSH session on the fly using public key authentication.

Default: cmd

-user *username*

(Optional) Specifies the user name to log in to the remote Windows system. The user must have rights to connect to the \$ADMIN share. If no user name is specified, the user name entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$User

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. To enter a hidden password after issuing this command, specify PW_GET() with an optional prompt as argument. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: \$\$Pass

-workDir *workdir*

Specifies the actual working directory.

Default: ""

on *remoteTargets*

Specifies a comma-separated list of remote systems to run the specified script on. The target nodes can be identified in two ways: By hostname/address or by WinRemote objects. Specifying target systems by name or address while using WinRemote objects offers more flexibility to control the execution process, for example, different credentials for different hosts. WinRemote objects and strings with hostname/address can be mixed in the remote target list. Elements of the target list are not automatically stringified, so when using literal strings for hostnames they must be placed into quotes.

Examples

To run cmd.exe (the command shell) on host1:

```
run-winRemote -host host1 -cmd cmd.exe
```

To run mybatch.bat on host1 and host 2 and verify the results of host2

```
x = run-winRemote -cmd mybatch.bat -localpath c:\installations\myinstall.exe on
"host1", "host2";
? x[1].hasCompleted();
? x[1].result();
? x[1].output();
```

See also:

[push-client Command--Install the Autoshell Client on a Remote Windows System \(Funclet\)](#) (see page 81)

[push-winRemote Command--Copy and Execute on Multiple Remote Windows Systems \(Funclet\)](#) (see page 83)

[rem-client Command--Remove the Autoshell Client from a Remote Windows System \(Funclet\)](#) (see page 88)

[run-client Command--Run the Autoshell Client on a Remote Windows System \(Funclet\)](#) (see page 92)

set alternate to Command--Set Alternate Output File (Cmdlet)

This command (cmdlet) sets or resets an alternate output file, for example, for `?`, `??`, `qout()`, or `qqout()`. When setting an alternate output file, any output from commands or functions is not only written to stdout, but also to the specified file. To write to only the alternate output file, suppress stdout using the "set console off" command.

The command has the following syntax:

```
set alternate to outFile [-append]
```

outFile

Specifies a string with an absolute or relative path of the alternate output file. If no file name is specified, the command restores the alternate output stream to the previous setting.

-append

(Optional) Appends the output stream to the specified file.

Examples

Write numbers 1 through 10 to output.txt and the console:

```
set alternate to output.txt
for(i=1; i<11; i++)
{
    ? i
}
set alternate to
```

Append numbers 11 through 20 to output.txt and the console:

```
set alternate to output.txt -append
for(i=11; i<21; i++)
{
    ? i
}
set alternate to
```

Write a directory listing to a file:

```
set alternate to dir.txt
dir -l
set alternate to
```

set console Command--Suppress Console Output (Cmdlet)

This command (cmdlet) is used to suppress any output to stdout which is the console in interactive mode generated by ?, ??, qout(), or qqout(). Typically used with the set alternate command when writing to a file.

The command has the following syntax:

```
set console {on|off}
```

on

Writes output to stdout.

off

Suppresses console output.

Example

Suppress console output and write numbers 1 through 10 to output.txt:

```
set console off
set alternate to output.txt
for(i=1; i<11; i++)
{
    ? i
}
set alternate to
set console on
```

See also:

[set alternate to Command--Set Alternate Output File \(Cmdlet\)](#) (see page 105)

set result display on/off Command--Turn Automatic Result Display On or Off (Cmdlet)

This command turns the result display on or off.

The command has the following syntax:

```
set result disp {on|off}
```

See also:

[dir, ls Commands--Get File and Directory Information \(Funclets\)](#) (see page 64)

start-service Command--Start a Windows Service on a Remote or Local System (Funclet)

The start-service command starts a Windows service on the local or remote system.

The command has the following syntax:

```
start-service -svcName sName [-host hostname] [-user username] [-pass password]
```

The command returns 1 on success and 0 on failure.

-svcName *sName*

Specifies the name of the service.

-host *hostname*

(Optional) The name of the remote server.

Default: ""

-user *username*

(Optional) Specifies the user name to use to log in to the remote node. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

Examples

To start the local Apache service:

```
? start-service -svcName Apache2.2
```

To start the Microsoft SQL Server service on a remote server REMOTESRV

```
? start-service -svcName MSSQLSERVER -host REMOTESRV -user admin -pass notsecret
```

stop-service Command--Stop a Windows Service on a Remote or Local System (Funclet)

The stop-service command stops a Windows service on the local or remote system.

The command has the following syntax:

```
stop-service -svcName sName [-host hostname] [-user username] [-pass password]
```

The command returns 1 on success and 0 on failure.

-svcName *sName*

Specifies the name of the service.

-host *hostname*

(Optional) The name of the remote server.

Default: ""

-user *username*

(Optional) Specifies the user name to use to log in to the remote node. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

-pass *password*

(Optional) Specifies password to use to log in to the remote node. If no password is specified, the password entered during AutoShell login is used. Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

Default: ""

Examples

To stop the local Apache service:

```
? stop-service -svcName Apache2.2
```

To stop the Microsoft SQL Server service on a remote server REMOTESRV:

```
? stop-service -svcName MSSQLSERVER -host REMOTESRV -user admin -pass notsecret
```

start-java Command--Start Java Virtual Machine (Cmdlet)

This command starts the Java Virtual Machine.

The command has the following syntax:

```
start-java [-classpath cp]
```

cp

(Optional) Specifies the class path.

Default: ""

wait Command--Wait for a Key Press (Cmdlet)

This command comes in the following versions:

- Optionally displays a prompt, waits for a key press, and assigns the character equivalent of the pressed key to a string variable.
- Optionally displays a prompt and waits for a key press.

The command has the following syntax:

```
wait [prompt] to var  
wait [prompt]
```

prompt

(Optional) String to be displayed as input prompt. If no prompt is specified, the command automatically displays "Press any key...". To display no prompt, pass an empty string (""). Unquoted argument tokens are automatically stringified. Prevent automatic quoting for expressions by placing expression code in parentheses.

var

Name of variable to assign the pressed key character to.

Examples

Wait for a key press and display a custom prompt message:

```
do  
{  
    wait "Enter a number between 1 and 5:" to x  
    x = parseInt(x);  
}  
while(x<1 || x>5)
```

See also:

[accept Command--Read String From stdin and Assign it to a Variable \(Cmdlet\)](#) (see page 53)

weak external Command--Declare Native External Function (Cmdlet)

This command declares a native external function.

The command has the following syntax:

```
weak external decl
```

decl

Specifies the declaration.

AutoShell Functions

This section details the AutoShell core functions. Functions define a set of processing instructions that receive zero or more arguments and return no value or exactly one value. Function arguments are passed in parentheses and must comply with JavaScript language syntax. Literal strings must be placed in quotation marks, and special characters inside the string (for example, backslashes) must be escaped.

Example

```
// Concatenate s n times
function repeat(s,n)
{
    var i;
    var ret="";
    for(i=0; i < n; i++)
        ret += s;
    return(ret);
}
? repeat("*", 10);
? repeat("\\", 25);
```

base64Decode--Decode a base64 Encoded String (Function)

This function performs the decoding of a base64 encoded string. The input string does not require base64Encode(). Any valid base64 encoded string can be passed for decoding.

The function has the following syntax:

```
base64Decode(sEncoded)
```

The function returns a string holding the decoded data. If an error occurs, for example, in case of malformed input data, the function raises an exception.

sEncoded

Specifies the string to decode.

Example

Decode a string with error handling:

```
try
{
    ? base64Decode("SGVsbG8gV29ybGQh");
}
catch(ex)
{
    ? "Error decoding string"
}
```

See also:

[base64Encode--Perform a base64 Encoding of a String \(Function\)](#) (see page 112)

base64Encode--Perform a base64 Encoding of a String (Function)

This function performs base64 encoding of the input data.

The function has the following syntax:

```
base64Encode(sData)
```

The function returns a string holding the base64 encoded data passed in sData. The result string is formatted using a line length of 76 characters.

sData

Defines the string to encode.

Example

Encode "Hello World":

```
? base64Encode("Hello World!")
```

See also:

[base64Decode--Decode a base64 Encoded String \(Function\)](#) (see page 111)

curDir--Retrieves the Current Directory (Function)

This function retrieves the current directory.

The function has the following syntax:

```
curDir()
```

The function returns a string holding the absolute path of the current working directory.

Example

Change to parent directory and retrieve its path:

```
chdir ..  
? curDir()
```


See also:

[cd, chdir Commands--Change Directory \(Cmdlet, Funclet\)](#) (see page 57)

gete--Get Environment Variable (Function)

Retrieves the value of an environment variable. This function an alternative way of getting environment variables apart from the standard JavaScript environment variable.

The function has the following syntax:

```
gete(sVar)
```

This function returns a string holding the current value of the environment variable specified by *sVar*. If the variable does not exist, an empty string is returned.

sVar

Specifies the environment variable.

Example

Retrieve the current path setting:

```
? gete("PATH");
```

See also:

[pute--Set an Environment Variable \(Function\)](#) (see page 116)

memoRead--Read a Text File Into a String (Function)

The memoRead() function reads a text file into a string.

Note: Using memoRead() on files containing binary information produces unpredictable results.

The function has the following syntax:

```
memoRead(sFileName [,nOffset] [,nCount])
```

The function returns the content of the text file as a JavaScript string.

sFilename

Specifies the name of a text file. If the specified file does not exist, the function creates an exception.

nOffset

(Optional) Specifies the number of characters from the beginning of the text file to the starting point of the read process.

nCount

(Optional) Specifies the number of characters that are read into a string.

Examples

Read Windows boot configuration:

```
s=memoRead("c:\boot.ini");  
? s
```

Read a text file from offset 10 to the end of file:

```
s = memoRead("file.txt", 10);
```

Read five characters from a text file starting at offset 10:

```
s = memoRead("file.txt", 10, 5);
```

See also:

[memoWrit--Write a String to a File \(Function\)](#) (see page 115)

[cat, type Commands--Display Text Files \(Cmdlets\)](#) (see page 56)

memoWrit--Write a String to a File (Function)

The memoWrit() function writes the content of a JavaScript string to a file.

Note: If the file exists, it can either replace the existing file or append the string to the existing string content.

The function has the following syntax:

```
memoWrit(sFileName, sString [, bAppend])
```

sString

Defines the string.

sFileName

Defines the name of a text file. If the specified file does not exist, the function raises an exception.

, bAppend

If set to true, the function adds the string to the given file.

The function returns with the number of bytes written to the specified file.

Note: If there are problems accessing the file, the function creates an exception.

Examples

Write a single-line text file:

```
memoWrit("file.txt", "First line\");
```

Append a second line to the existing file:

```
memoWrit("file.txt","Second line\", true);
```

See also:

[memoRead--Read a Text File Into a String \(Function\)](#) (see page 114)

platform--Query Operating System Type (Function)

This function returns the type of operating system the script is currently running on.

The function has the following syntax:

```
platform()
```

The function returns the system type name in a string.

Example

Verify that a script is being run on Windows:

```
if(platform()!="Windows")
{
  ? "This script uses Windows specific features!"
  ? "Please run on Windows only."
  quit
}
start-service -svcName Apache2.2
```

See also:

[shellType--Query AutoShell Type \(Function\)](#) (see page 133)

pute--Set an Environment Variable (Function)

Sets the value of an existing environment variable or creates one with the specified value if the variable does not exist. This function an alternative way of setting environment variables apart from the standard JavaScript environment variable.

The function has the following syntax:

```
pute(sVar, sVal)
```

The function returns true or false indicating success or failure.

sVar

Specifies the name of the environment variable.

sVal

Defines the value of the environment variable.

Example

Append the current work directory to the path:

```
pute(gete("PATH")+";"+curDir());
```

See also:

[gete--Get Environment Variable \(Function\)](#) (see page 113)

qout--Write Output Followed by Linefeed to stdout (Function)

The `qout` function writes output followed by linefeed to `stdout`. The function writes the string representation of each argument in a list to `stdout`. A single space is automatically placed between each argument. The `set console` and `set alternate` commands can suppress or redirect the output. The output terminates with a linefeed character.

The function has the following syntax:

```
qout([x, ...])
```

x

(Optional) Defines a list of arguments to display. If the list is empty, a linefeed character is printed.

This function does not return a value.

Examples

Print "Hello World!":

```
? "Hello World!"
```

Print a list of numbers:

```
? 1,2,3
```

Print an empty line:

```
?
```

Output the numbers from 1 to 10.

Note: The `?` command can only be used at the beginning of a line. To output text from within an expression, use the corresponding `qout()` function:

```
for(i=1;i<11;i++) qout(i);
```

See also:

[qqout--Write Output to stdout \(Function\)](#) (see page 118)

[?? Command--Write Output in a List to stdout \(Cmdlet\)](#) (see page 52)

qqout--Write Output to stdout (Function)

The qqout function writes the string representation of each argument in a list to stdout. A single space is automatically placed between each argument. The set console and set alternate commands can redirect or suppress the output. Subsequent output calls using ?, ??, qout() or qqout() will place their output immediately after the original output, so ?? or qqout() are typically used to construct output lines using several invocations.

The function has the following syntax:

```
qqout( [x, ... ] )
```

x

(Optional) Defines a list of arguments to display. If the list is empty, a linefeed character is printed.

This function does not return a value.

Examples

Print three column headers with separating spaces:

```
?? "Col1"  
?? " "  
?? "Col2"  
?? " "  
? "Col3"
```

The third label is printed using the ? function so the cursor moves to the beginning of the next line.

See also:

[qout--Write Output Followed by Linefeed to stdout \(Function\)](#) (see page 117)

[? Command--Write Output to stdout \(Cmdlet\)](#) (see page 52)

[?? Command--Write Output in a List to stdout \(Cmdlet\)](#) (see page 52)

[set alternate to Command--Set Alternate Output File \(Cmdlet\)](#) (see page 105)

[set console Command--Suppress Console Output \(Cmdlet\)](#) (see page 106)

regCreateKey--Create a Registry Key (Function)

The `regCreateKey` function creates a key or a key hierarchy under the registry key specified by `sKeyParent` on the local system or a remote system.

This function has the following syntax:

```
regCreateKey(sKeyParent, sKey)
```

sKeyParent

Specifies the parent key.

sKey

Defines the new key

If the function fails, it raises an exception. Otherwise it returns true. The function succeeds if the specified key or part of a key hierarchy exists.

Examples

Create key ACEM under the local machine SOFTWARE key with error handling:

```
try {
    regCreateKey("HKLM\\SOFTWARE", "ACME");
    ? "Success"
}
catch(e) {
    ? "Failure", e
}
```

Create a key hierarchy:

```
regCreateKey("HKLM\\SOFTWARE", "ACME\\HOME");
```

Create key ACEM on remote machine client8:

```
regCreateKey("client8:HKLM\\SOFTWARE", "ACME");
```

See also:

[regDeleteKey--Delete a Registry Key or a Key Hierarchy \(Function\)](#) (see page 121)

[regSetVal--Set Registry Value \(Function\)](#) (see page 131)

regCreateSubkeys--Create Subkeys From an Array (Function)

The `regCreateSubKeys` function creates a key subtree under a given key from an array on the local system or a remote system. The array must contain the names of the keys to be created as relative path strings starting at `sKey`. `regGetSubKeys()` returns this type of array.

This function has the following syntax:

```
regCreateSubKeys(sKey, aSubKeys)
```

sKey

Specifies the registry key.

aSubKeys

Specifies the key subtree.

If the function fails, it raises an exception. Otherwise it returns true.

Examples

Create keys 'x', 'y' and 'z' under HKLM\Software\ACME:

```
regCreateSubKeys("HKLM\Software\ACME", ["x", "y", "z"]);
```

Create nested subkey tree:

```
regCreateSubKeys("HKLM\Software\ACME", ["a\b\c", "x\y\z"]);
```

Get subkey tree from remote system client8 and apply it to remote system client9:

```
arr = regGetSubKeys("client8::HKLM\Software\ACME", true);  
regSetSubKeys("client9::HKLM\Software\ACME", arr);
```

Note: The key values must be handled separately.

See also:

[regCreateKey--Create a Registry Key \(Function\)](#) (see page 119)

[regGetSubKey--Retrieve Sub Keys of a Registry Key \(Function\)](#) (see page 125)

regDeleteKey--Delete a Registry Key or a Key Hierarchy (Function)

The `regDeleteKey` function removes a key and all associated values from the registry on the local system or a remote system. If the specified key contains subkeys, the removal fails by default. However by passing true for *fRecurse* the function deletes *sKey* including all nested subkeys.

The function has the following syntax:

```
regDeleteKey(sKey [, fRecurse])
```

sKey

Specifies the registry key

, fRecurse

Deletes nested subkeys.

If the function fails, it raises an exception. Otherwise it returns true.

Examples

Delete key named "Settings" from HKLM\Software\ACME with error handling:

```
try {
    regDeleteKey("HKLM\Software\ACME\Settings");
    ? "Success"
}
catch(e) {
    ? "Failure:", e
}
```

Delete "Settings" key forcing recursive delete of subkeys:

```
regDeleteKey("HKLM\Software\ACME\Settings", true);
```

Delete "Settings" key on remote machine client8:

```
regDeleteKey("client8:HKLM\Software\ACME\Settings");
```

See also

[regCreateKey--Create a Registry Key \(Function\)](#) (see page 119)

[regDeleteVal--Delete a Registry Value \(Function\)](#) (see page 122)

regDeleteVal--Delete a Registry Value (Function)

The `regSetVal` function removes a value from the registry key specified by `sKey` on the local system or a remote system.

The function has the following syntax:

```
regDeleteVal(sKey, sValName)
```

sKey

Specifies the registry key.

sValName

Specifies the value.

If the function fails, it raises an exception. Otherwise, it returns true.

Examples

Delete value named "Label" from HKLM\Software\ACME with error handling:

```
try {
    regDeleteVal("HKLM\Software\ACME", "Label");
    ? "Success"
}
catch(e) {
    ? "Failure:", e
}
```

Delete a value on remote machine client8:

```
regDeleteVal("client8:HKLM\Software\ACME", "Name");
```

See also:

[regDeleteKey--Delete a Registry Key or a Key Hierarchy \(Function\)](#) (see page 121)

[regSetVal--Set Registry Value \(Function\)](#) (see page 131)

regGetKeyValues--Get Registry Key Value Information (Function)

The `regGetKeyValues` function returns a two-dimensional array with name, actual value, and type of registry values for a key on a local or remote system. The returned array can be passed to `regSetKeyValues()` to copy value settings from one system to another.

The function has the following syntax:

```
regGetKeyValues(sKey)
```

sKey

Specifies the registry key.

The function returns a two-dimensional array. Each top-level element holds a three element sub-array containing a string representing the value name, a polymorphic value for the setting, and an integer for the value type. The type of registry value determines the type of the return value.

- `REG_DWORD` values are returned as numeric
- `REG_SZ` and `REG_EXPAND_SZ` as string
- `REG_MULTI_SZ` as an array of strings
- `REG_EXPAND_SZ` values are already expanded using the current environment variables
- `REG_BINARY` values are returned as AutoShell `CharArr` objects because JavaScript string variables cannot properly handle binary data.

If the function fails, it raises an exception.

Examples

Retrieve and display value information for the key HKLM\Software\ACME\Settings:

```
arr = regGetKeyValues("HKLM\\Software\\ACME\\Settings");
l = arr.length;
for(i=0; i < l; i++)
{
    ? "Name: ", arr[i][0]
    ? "Value:", arr[i][1]
    ?? "Type: "
    switch(arr[i][2])
    {
        case REG_SZ:
            ? "REG_SZ"
            break;
        case REG_EXPAND_SZ:
            ? "REG_EXPAND_SZ"
            break;
        case REG_BINARY:
            ? "REG_BINARY"
            break;
        case REG_DWORD:
            ? "REG_DWORD"
            break;
        case REG_MULTI_SZ:
            ? "REG_MULTI_SZ"
            break;
        default:
            ? "Unknown"
    }
}
```

Get values from remote system client8 and apply them to remote system client9:

```
arr = regGetKeyValues("client8::HKLM\\Software\\ACME\\Settings");
regSetKeyValues("client9::HKLM\\Software\\ACME\\Settings", arr);
```

See also:

[regGetValue--Get Registry Value \(Function\)](#) (see page 127)

[regSetKeyValues--Set Registry Key Values From an Array \(Function\)](#) (see page 130)

regGetSubKey--Retrieve Sub Keys of a Registry Key (Function)

The `regGetSubKeys` function returns the subkeys of a specified registry key on the local system or a remote system in an array. By default only the direct subkeys are returned. Optionally all subkeys can be retrieved recursively.

The function has the following syntax:

```
regGetSubKeys(sKey, fRecurse)
```

sKey

Specifies the registry key

, fRecurse

Retrieves nested subkeys.

This function returns an array holding the names of the subkeys of `sKey`. An individual subkey name is returned as a relative path starting at `sKey`. The returned array can be passed `regCreateSubKeys()` to copy a subkey tree from one system to another.

If the function fails, it raises an exception.

Examples

Get subkeys of `HKLM\Software\CA` with error handling:

```
try {
    var arr, i, l;
    arr = regGetSubKeys("HKLM\\Software\\CA");
    l = arr.length;
    for(i=0; i < l; i++)
    {
        ? arr[i]
    }
}
catch(e) {
    ? "Failure:", e
}
```

Recursively get all subkeys of HKLM\Software\CA:

```
arr = regGetSubKeys("HKLM\\Software\\CA", true);
l = arr.length;
for(i=0; i < l; i++)
{
    ? arr[i]
}
```

Get subkey tree from remote system client8 and apply it to remote system client9:

```
arr = regGetSubKeys("client8::HKLM\\Software\\ACME", true);
regSetSubKeys("client9::HKLM\\Software\\ACME", arr);
```

Note: The key values must be handled separately.

See also:

[regCreateKey--Create a Registry Key \(Function\)](#) (see page 119)

[regCreateSubkeys--Create Subkeys From an Array \(Function\)](#) (see page 120)

regGetVal--Get Registry Value (Function)

The `regGetVal` function retrieves the setting of the value *valName* under the registry key specified by *key* on the local system or a remote system.

The function has the following syntax:

```
regGetVal(sKey, sValName)
```

sKey

Specifies the registry key.

sValName

Specifies the value.

If the specified value exists, the function returns a polymorphic value representing the setting of the specified value. The type of the registry value determines the type of the return value.

- REG_DWORD values are returned as numeric
- REG_SZ and REG_EXPAND_SZ as string
- REG_MULTI_SZ as an array of strings.
- REG_EXPAND_SZ values are already expanded using the current environment variables.
- REG_BINARY values are returned as AutoShell CharArr objects because JavaScript string variables cannot properly handle binary data.

If the specified key or the value does not exist, the function raises an exception.

Examples

Get value named "Number" from key HKLM\Software\ACME:

```
regGetVal("HKLM\Software\ACME", "Number");
```

Get value named "Label" from key HKLM\Software\ACME with error handling:

```
try {
    regGetVal("HKLM\Software\ACME", "Label");
    ? "Success"
}
catch(e) {
    ? "Failure:", e
}
```

Get value from remote machine client8:

```
regGetVal("client8:HKLM\Software\ACME", "Name");
```

See also:

[regGetKeyValues--Get Registry Key Value Information \(Function\)](#) (see page 123)

[regSetVal--Set Registry Value \(Function\)](#) (see page 131)

regIsKey--Check the Existence of a Registry Key (Function)

The regIsKey function checks if the specified registry key exists on the local or a remote system.

The function has the following syntax:

```
regIsKey(sKey)
```

sKey

Specifies the registry key.

The function returns a Boolean value indicating the presence of *sKey*:

true

Indicates that the key exists.

false

Indicates that the key does not exist.

Examples

Locate an ACME key under HKLM\SOFTWARE on the local system:

```
if (regIsKey("HKLM\SOFTWARE\ACME"))
? "Key was found!"
else
? "Key does not exist"
```

Perform the same check on the remote system ascl1:

```
if (regIsKey("ascl1::HKLM\SOFTWARE\ACME"))
? "Key was found"
else
? "Key does not exist"
```

See also:

[regCreateKey--Create a Registry Key \(Function\)](#) (see page 119)

[regIsVal--Check the Existence of a Registry Value \(Function\)](#) (see page 129)

regIsVal--Check the Existence of a Registry Value (Function)

The `regIsVal` function checks if the specified registry value exists on the local or a remote system.

The function has the following syntax:

```
regIsVal(sKey, sValName)
```

sKey

Specifies the registry key.

sValName

Specifies the name of the value.

The function returns a Boolean indicating presence of *sValName*:

true

Indicates that the value exists.

false

Indicates that the value does not exist.

Examples

Verify `HKLM\SOFTWARE\ACME` for a value, 'home':

```
if(regIsVal("HKLM\SOFTWARE\ACME", "home"))
{
    ? "Value was found"
}
else
{
    ? "Value does not exist"
}
```

See also:

[regGetVal--Get Registry Value \(Function\)](#) (see page 127)

[regIsKey--Check the Existence of a Registry Key \(Function\)](#) (see page 128)

[regSetVal--Set Registry Value \(Function\)](#) (see page 131)

regSetKeyValues--Set Registry Key Values From an Array (Function)

The `regSetKeyValues()` function creates or sets values for a key from a two-dimensional array on the local or a remote system. Each top-level element in the array must contain a sub-array with a string representing the name of the value, a polymorphic JavaScript value to set, and optionally, an integer specifying the type of the value to create or set. `regGetKeyValues()` returns this type of array.

The function has the following syntax:

```
regSetKeyValues(sKey, aVals)
```

sKey

Specifies the registry key.

sVals

Defines the array that contains the values.

If the function fails, it raises an exception. Otherwise it returns true.

Examples

Set values 'Number' and 'Label' for HKLM\Software\ACME:

```
vals = [{"Number", 42}, {"Label", "Hello"}];  
regSetKeyValues("HKLM\\Software\\ACME\\Settings", vals);
```

Set values explicitly specifying type:

```
vals = new Array(2);  
val[0] = ["MyPath", "%Path%;c:\\MyDir", REG_EXPAND_SZ];  
val[1] = ["Number", 42, REG_DWORD];  
regSetKeyValues("HKLM\\Software\\ACME\\Settings", vals);
```

Get values from remote system client8 and apply them to remote system client9:

```
arr = regGetKeyValues("client8:HKLM\\Software\\ACME\\Settings");  
regSetKeyValues("client9:HKLM\\Software\\ACME\\Settings", arr);
```

See also:

[regGetKeyValues--Get Registry Key Value Information \(Function\)](#) (see page 123)

[regSetVal--Set Registry Value \(Function\)](#) (see page 131)

regSetVal--Set Registry Value (Function)

The `regSetVal` creates or sets a value for the registry key specified by `sKey` on the local system or a remote system. If the value identified by `sValName` does not exist, it is created, otherwise the existing setting is overwritten with the `value`.

The function has the following syntax:

```
regSetVal(sKey, sValName, value, [type])
```

sKey

Specifies the registry key.

sValName

Specifies the name of the value.

value

Defines the value.

type

Specifies the type of the value.

If the function fails, it raises an exception. Otherwise it returns true.

Examples

Set `REG_DWORD` value for key `HKLM\Software\ACME` named "Number" to 42:

```
regSetVal("HKLM\Software\ACME", "Number", 42);
```

Set `REG_SZ` value for key `HKLM\Software\ACME` named "Label" to "Hello" with error handling:

```
try {
    regSetVal("HKLM\Software\ACME", "Label", "Hello");
    ? "Success"
}
catch(e) {
    ? "Failure:", e
}
```

Set a REG_EXPAND_SZ value:

```
regSetVal("HKLM\\Software\\ACME", "MyPath", "%Path%;c:\\MyDir", REG_EXPAND_SZ);
```

Set a REG_MULTI_SZ value:

```
arr=["Alice", "Bob", "Carol"];  
regSetVal("HKLM\\Software\\ACME", "People", arr);
```

Set string value on remote machine client8:

```
regSetVal("client8::HKLM\\Software\\ACME", "Name", "Bob");
```

See also:

[regCreateKey--Create a Registry Key \(Function\)](#) (see page 119)

[regDeleteVal--Delete a Registry Value \(Function\)](#) (see page 122)

[regGetVal--Get Registry Value \(Function\)](#) (see page 127)

[regSetKeyValues--Set Registry Key Values From an Array \(Function\)](#) (see page 130)

setProcExitCode--Set AutoShell Exit Code (Function)

This function sets the AutoShell exit from a user script overriding a possible AutoShell defined exit code. On Windows systems, the exit code can be retrieved through the %errorlevel% variable after AutoShell terminates.

The function has the following syntax:

```
setProcExitCode(iExitCode)
```

iExitCode

Specifies the exit code.

Example

Set the AutoShell exit code to -42 and exit AutoShell:

```
setProcExitCode(-42);  
quit
```

See also:

[exit, quit Commands--Terminate an AutoShell Session \(Cmdlets\)](#) (see page 67)

shellType--Query AutoShell Type (Function)

This function is used to determine if a script is currently being executed inside a manager AutoShell or a client AutoShell. The client AutoShell does not provide all the functionality of the manager AutoShell (for example, no remote execution). shellType() can be used to verify that a script is being executed inside the intended AutoShell version.

The function has the following syntax:

```
shellType()
```

The function returns a string indicating the type of AutoShell which is currently active:

- manager
- client

Example

Verify AutoShell type and quit when running inside a client shell:

```
if(shellType()!="manager")
{
    ? "This script requires AutoShell manager!"
    quit
}
```

See also:

[platform--Query Operating System Type \(Function\)](#) (see page 115)

typeName--Get the Type Name of an Expression (Function)

This function retrieves the type name of an arbitrary expression. In particular, for objects it returns the name of the defining class.

The function has the following syntax:

```
typeName(val)
```

The function returns a string holding the type name of the passed value. For simple types, it returns one of the following strings:

- Boolean
- function
- null
- number
- string
- void
- xml

For objects, it returns the name of the defining class.

val

Defines the expression.

Example

Get the type of some expressions:

```
? typeName(19) // number
d = new Date;
? typeName(d) // Date
a = ["a", "b", "c"];
? typeName(a) // Array
? typeName(a[0])// string
```

See also:

[gete--Get Environment Variable \(Function\)](#) (see page 113)

[platform--Query Operating System Type \(Function\)](#) (see page 115)

AutoShell Classes

This section details the AutoShell class definitions. AutoShell is based on JavaScript, so object orientation is achieved through function objects as in JavaScript. Functions can serve as regular functions, classes, constructors, or methods.

Objects are created as function instances using the new operator, for example:

```
now = new Date();
```

AutoShell also implements its own classes that are instantiated using the new operator. In many cases, there is no need to work directly with the AutoShell classes because they are encapsulated using AutoShell command definitions.

Constructors are methods called when an object is created. There may be several constructors for a class. In this case, it depends on the actual arguments in the object instantiation which constructor is called.

Methods are a set of associated functions called on an object using the '.' operator and access or manipulate object data using the "this" property.

See also a JavaScript language reference.

Example

Calculate the date/time ten hours from the current time. Day wrapping at midnight is handled properly.

```
d = new Date();  
d.setHours(d.getHours()+10)
```

More Information

[OSRedirect Class](#) (see page 136)

[RemoteTarget Class](#) (see page 146)

OSRedirect Class

OSRedirect is a utility class that is used with the run-local command running scripts locally. Its primary use is to redirect the output of child processes into JavaScript strings. The system variable \$\$stdout, used by the !! command, is also an instance of the OSRedirect class.

Constructor

OSRedirect()

Methods

bool clear()

bool errorOccurred()

string errout()

bool hasCompleted()

string output()

int result()

Event Handlers

void onCompleted(fCompleted)

void onError(fError)

void receivedErrOutput(s)

void receivedOutput(s)

void receivedResult(rc)

See also:

[OSRedirect.OSRedirect Constructor](#) (see page 137)

[OSRedirect.clear Method](#) (see page 137)

[OSRedirect.errorOccurred Method](#) (see page 138)

[OSRedirect.errout Method](#) (see page 139)

[OSRedirect.hasCompleted Method](#) (see page 140)

[OSRedirect.onCompleted Method](#) (see page 141)

[OSRedirect.onError Method](#) (see page 141)

[OSRedirect.output Method](#) (see page 142)

[OSRedirect.receivedErrOutput Method](#) (see page 143)

[OSRedirect.receivedOutput Method](#) (see page 143)

[OSRedirect.receivedResult Method](#) (see page 144)

[OSRedirect.result Method](#) (see page 145)

[run-local Command--Execute a Script on the Local System \(Funclet\)](#) (see page 93)

OSRedirect.OSRedirect Constructor

The constructor called when creating an object.

The constructor has the following syntax:

```
OSRedirect()
```

A constructor does not return a value. If an error occurs, it raises an exception.

Example

Create an OSRedirect object and use it in an invocation.

```
out = new OSRedirect();  
! ping asclil -output out  
? out.output();
```

See also:

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

OSRedirect.clear Method

This method resets the output and error output buffers of OSRedirect() objects. The method is called internally by the ! and !! commands, so there is no need to call it explicitly when using the same OSRedirect() instance in multiple OS invocations.

This method has the following syntax:

```
clear()
```

The method does not return a value.

Example

```
out = new OSRedirect();  
! ping asclil -output out  
? out.output();  
out.clear();  
? out.output(); // Empty output
```

See also:

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.errorOccurred Method

This method checks if an error occurs during remote execution. If an error occurs, `errout()` may provide an error description.

This method has the following syntax:

```
errorOccurred()
```

The method returns the following values:

true

Indicates that an error occurred.

false

Indicates that no error occurred.

Example

Execute invalid command with error check:

```
!! asdfghjk
if($$stdout.errorOccurred())
  ? "Error:", $$stdout.errout()
else
  ? $$stdout.output()
```

See also:

[OSRedirect.errout Method](#) (see page 139)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.errout Method

Child processes produce the following types of result information.

- Regular output written to the screen
- Error output written to the screen
- Final exit code ("errorlevel")

This method has the following syntax:

```
errout()
```

This method returns a string with the output written by the child process to stderr.

Example

Invoke a nslookup and display regular and error output:

```
!! nslookup ascli1  
? $$stdout.output()  
? $$stdout.errout()
```

See also:

[OSRedirect.errorOccurred Method](#) (see page 138)

[OSRedirect.output Method](#) (see page 142)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.hasCompleted Method

This method checks if the execution of a remote command has completed. If `run-remote` is invoked without the optional `-wait` switch, the execution runs asynchronously. Call this method to verify if a command completes and a result is available. The method `hasCompleted()` returns `true` when the command successfully completes, or `false` when an error occurs that prevents further processing.

This method has the following syntax:

```
hasCompleted()
```

The method returns the following values:

true

Indicates if the command processing has been completed.

false

Indicates if the command processing has not been completed, for example, host not found.

Example

```
!! dir c:\*. * /s  
? $$stdout.hasCompleted() // true
```

See also:

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.onCompleted Method

Event handlers are optional functions that can be implemented in user scripts and called by AutoShell when certain events occur. If present, the event handler is called after the child process finishes.

This method has the following syntax:

```
onCompleted(fCompleted)
```

Note: The AutoShell ignores event handler return values.

Example

Specify completion event handler:

```
out = new OSRedirect();
out.onCompleted = function(f){if(f)qout("ret=", this.result());};
! ver -output out
```

See also:

[OSRedirect.result Method](#) (see page 145)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.onError Method

Event handlers are optional functions that can be implemented in user scripts and called by AutoShell when certain events occur. If present, the event handler is called when the error state of the OSRedirect object changes.

This method has the following syntax:

```
onError(fError)
```

Note: The AutoShell ignores event handler return values.

Example

Specify completion event handler:

```
out = new OSRedirect();
out.onError = function(f){if(f)qout("An error occurred!");};
! ver -output out
```

See also:

[OSRedirect.errorOccurred Method](#) (see page 138)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.output Method

Child processes produce three types of result information. The regular output typically written to the screen, error output typically written to the screen, and a final exit code ("errorlevel"). This method returns a string with the output written by the child process to stdout.

This method has the following syntax:

```
output()
```

The method returns a string that contains the command output.

Example

Get Windows OS version string:

```
!! ver  
? $$stdout.output()
```

See also:

[OSRedirect.errout Method](#) (see page 139)

[OSRedirect.result Method](#) (see page 145)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.receivedErrOutput Method

Event handlers are optional functions that can be implemented in user scripts and called by AutoShell when certain events occur. If present, the event handler is called when the child process writes to stderr. You can call this handler multiple times.

This method has the following syntax:

```
receivedErrOutput(s)
```

Note: The AutoShell ignores event handler return values.

Example

Display error output while being received:

```
out = new OSRedirect();
out.receivedErrOutput = function(s){qout(s)};
! ver -output out
```

See also:

[OSRedirect.receivedOutput Method](#) (see page 143)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.receivedOutput Method

Event handlers are optional functions that can be implemented in user scripts called by AutoShell when certain events occur. If present, the event handler is called when the child process writes to stdout. You can call this handler multiple times.

This method has the following syntax:

```
receivedOutput(s)
```

Note: The AutoShell ignores event handler return values.

Example

Display output while being received:

```
out = new OSRedirect();
out.receivedOutput = function(s){qout(s)};
! ver -output out
```

See also:

[OSRedirect.receivedErrOutput Method](#) (see page 143)

[OSRedirect.receivedResult Method](#) (see page 144)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.receivedResult Method

Event handlers are optional functions that can be implemented in user scripts and called by AutoShell when certain events occur. If present, the event handler is called when the exit code of the child process is set.

This method has the following syntax:

```
receivedResult(rc)
```

Note: The AutoShell ignores event handler return values.

Example

Display exit code from event handler:

```
out = new OSRedirect();  
out.receivedResult = function(rc){qout("Exit code", rc)};  
! ver -output out
```

See also:

[OSRedirect.receivedOutput Method](#) (see page 143)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

OSRedirect.result Method

Child processes produce three types of result information. The regular output typically written to the screen, error output typically written to the screen, and a final exit code ("errorlevel").

This method has the following syntax:

```
result()
```

This method returns the child process exit code.

Example

On Windows ping sets the exit code to 0 if the specified host responded otherwise to 1. Check and display result:

```
!! ping asclil
if($$stdout.result()==0)
    ? "Ping successful"
else
    ? "Ping failed"
```

Note: An application-specific failure does not cause the `errorOccurred()` flag of `OSRedirect()` to be set.

See also:

[OSRedirect.output Method](#) (see page 142)

[! Command--Invoke Command or Child Process \(Cmdlet\)](#) (see page 51)

[!! Command--Invoke Command or Child Process and Auto-capture Output \(Cmdlet\)](#) (see page 50)

RemoteTarget Class

RemoteTarget is a utility class that is used with the run-remote command in remote script execution. RemoteTarget specifies details of the target system, like host name and address, credentials to use for authentication, or retrieves status and result information, like error condition, output, and final result.

Constructor

```
RemoteTarget(host [,user] [,pass] [,key] [,phrase] [, port])
```

Methods

```
void abort()  
int createRemoteContext()  
int destroyRemoteContext()  
bool errorOccurred()  
bool execute(strScript)  
int getError()  
string getHostName()  
bool hasCompleted()  
string output()  
string result()  
bool wasAborted()
```

Event Handlers

```
int onError(e)  
int receivedOutput(s)  
int receivedResult(s)
```

Variables

```
static readonly REM_ERR_NONE  
static readonly REM_ERR_CON  
static readonly REM_ERR_AUTH  
static readonly REM_ERR_CHN  
static readonly REM_ERR_READ  
static readonly REM_ERR_EXEC  
static readonly REM_ERR_ABORT  
static readonly REM_ERR_DISC
```

RemoteTarget.RemoteTarget Constructor

The constructor called when creating an object. A constructor does not return a value. If an error occurs, it raises an exception.

The constructor has the following syntax:

```
RemoteTarget(host [,user] [,pass] [,key] [,phrase] [, port])
```

host

Specifies the name of the remote host.

user

(Optional) Specifies the user name to use to log in to the remote node. A user name is required for password or public key authentication. If no user name is specified, the user name entered during AutoShell login is used.

Default: \$\$User

pass

(Optional) Specifies password to use to log in to the remote host. If no password is specified, the password entered during AutoShell login is used.

Default: \$\$Pass

key

(Optional) Specifies an absolute or relative path to a file containing a private RSA key to use to log in to the remote host. If a private key and a password are specified, AutoShell attempts a public key logon first, and if that fails, a password logon.

Default: ""

phrase

(Optional) Specifies the passphrase for a private key. If the key is not encrypted, the passphrase is not required.

Default: ""

port

(Optional) Specifies the port on which to connect to the remote host.

Default: SSH standard port 22.

Examples

Create a RemoteTarget object using password authentication:

```
rt = new RemoteTarget("ascli1", "bob", "ca123456");
```

Create a RemoteTarget object attempting public key authentication followed by password if public key authentication fails:

```
rt = new RemoteTarget("ascli1", "bob", "ca123456", "bob_rsa.key");
```

In this case the file bob_rsa.key contains the private RSA key.

See also:

[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)

RemoteTarget.abort Method

Remote script execution by default is performed asynchronously. If an outstanding operation must be aborted, call abort() on the RemoteTarget object associated with the remote operation.

This method has the following syntax:

```
abort()
```

This method does not return a value.

Example

Abort a remote operation:

```
rt = new RemoteTarget("ascli1");
run-remote Math.sin(Math.PI/2) on rt
if(!rt.hasCompleted())
{
    rt.abort()
    ? "Operation aborted:", rt.wasAborted()
}
```

See also:

[RemoteTarget.wasAborted Method](#) (see page 159)

[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)

RemoteTarget.createRemoteContext Method

By default, a new Autoshell session is created for each command run on a RemoteTarget object. When running multiple commands on the same target, it is preferable to use a single persistent session to execute multiple scripts. Additionally a persistent session maintains state. For example, a variable created during one execution is still available during subsequent evaluations.

Calling createRemoteContext() on a RemoteTarget object creates a new Autoshell session on the remote system, and keeps it open until either destroyRemoteContext() is called, or the RemoteTarget object is collected. Calling createRemoteContext() when a context exists destroys the existing context and creates a new one.

This method has the following syntax:

```
createRemoteContext()
```

The method returns one of the following error codes:

RemoteTarget.REM_ERR_NONE

Indicates success.

RemoteTarget.REM_ERR_CON

Indicates a connection error, for example, host unknown.

RemoteTarget.REM_ERR_AUTH

Indicates an authentication error.

Example

Create a persistent session, define a variable in the first evaluation and retrieve it in the second one:

```
rt=new RemoteTarget("ascli1");
if(rt.createRemoteContext()==RemoteTarget.REM_ERR_NONE)
{
    run-remote x=42 on rt
    run-remote x on rt -wait
    ? get-remoteResult(rt) // 42
    rt.destroyRemoteContext();
    ? get-remoteResult(rt) // empty
}
else
{
    ? "Error occurred!";
}
```

See also:

[RemoteTarget.destroyRemoteContext Method](#) (see page 150)

RemoteTarget.destroyRemoteContext Method

This method destroys a context previously created by `createRemoteContext()`. After destroying a persistent context the `RemoteTarget` object can still be used in remote execution. Until another context is created, it creates and destroys a context internally for each script evaluation.

This method has the following syntax:

```
destroyRemoteContext()
```

The method returns one of the following error codes:

RemoteTarget.REM_ERR_NONE

Indicates success.

RemoteTarget.REM_ERR_DISC

Indicates a disconnection from the remote system.

Example

Create a persistent session, define a variable in the first evaluation and retrieve it in the second one:

```
rt=new RemoteTarget("ascli1");
if(rt.createRemoteContext()==RemoteTarget.REM_ERR_NONE)
{
    run-remote x=42 on rt
    run-remote x on rt -wait
    ? get-remoteResult(rt) // 42
    rt.destroyRemoteContext();
    ? get-remoteResult(rt) // empty
}
else
{
    ? "Error occurred!";
}
```

See also:

[RemoteTarget.createRemoteContext Method](#) (see page 149)

RemoteTarget.errorOccurred Method

This method checks if an error occurs during remote execution. If an error occurs, `getError()` provides further information.

This method has the following syntax:

```
errorOccurred()
```

The method returns the following values:

true

Indicates that an error occurred.

false

Indicates that no error occurred.

Example

Check for error after remote execution finishes:

```
rt = new RemoteTarget("ascli1");
run-remote Math.cos(0) on rt -wait
if(rt.errorOccurred())
{
    ? "Error occurred!"
}
else
{
    ? get-remoteResult(rt)
}
```

See also:

[RemoteTarget.getError Method](#) (see page 152)

RemoteTarget.execute Method

This method is the core method invoked by the run-remote command. This method should not be called directly because run-remote performs certain processing on the specified script before passing it to this method.

This method has the following syntax:

```
execute(strScript)
```

The method returns the following values:

true

Indicates that the asynchronous execution kicked off successfully.

false

Indicates that a failure occurs when starting the remote execution.

Examples

Evaluate an expression (recommended):

```
rt = new RemoteTarget("ascli1");  
run-remote Math.sqrt(1) on rt  
? get-remoteResult(rt);
```

Evaluate an expression (not recommended):

```
rt = new RemoteTarget("ascli1");  
rt.execute("Math.sqrt(1)");  
? get-remoteResult(rt);
```

See also:

[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)

RemoteTarget.getError Method

This method returns detailed error information after a remote execution attempt fails.

This method has the following syntax:

```
getError()
```


The method returns one of the following error codes:

RemoteTarget.REM_ERR_NONE

Indicates success.

RemoteTarget.REM_ERR_CON

Indicates a connection error, for example, host unknown.

RemoteTarget.REM_ERR_AUTH

Indicates an authentication error.

RemoteTarget.REM_ERR_CHN

Indicates an error when establishing a channel.

RemoteTarget.REM_ERR_READ

Indicates an error when reading from a remote system.

RemoteTarget.REM_ERR_EXEC

Indicates an error when evaluating expressions.

RemoteTarget.REM_ERR_ABORT

Indicates when a user aborts an operation.

RemoteTarget.REM_ERR_DISC

Indicates an error when disconnecting from a remote system.

Example

Display detailed error information:

```
rt = new RemoteTarget("ascli1");
run-remote Math.sqrt(1) on rt -wait
switch(rt.getError())
{
    case RemoteTarget.REM_ERR_NONE:
        ? "Success"
        break;
    case RemoteTarget.REM_ERR_CON:
        ? "Error connecting"
        break;
    case RemoteTarget.REM_ERR_AUTH:
        ? "Error authenticating"
        break;
    case RemoteTarget.REM_ERR_CHN:
        ? "Error establishing channel"
        break;
```

```
case RemoteTarget.REM_ERR_READ:
    ? "Error reading from remote system"
    break;
case RemoteTarget.REM_ERR_EXEC:
    ? "Error evaluating expression"
    break;
case RemoteTarget.REM_ERR_ABORT:
    ? "Operation aborted by user"
    break;
case RemoteTarget.REM_ERR_DISC:
    ? "Error disconnecting from remote system"
    break;
default:
    ? "Unknown error"
}
```

See also:

[RemoteTarget.errorOccurred Method](#) (see page 151)

[RemoteTarget.onError Method](#) (see page 156)

RemoteTarget.getHostName Method

This method returns the host name or address that was specified when creating the RemoteTarget object.

This method has the following syntax:

```
getHostName()
```

The method returns a string with the hostname or address of the target node.

Example

Implicitly create a RemoteTarget object and retrieve the hostname:

```
aRT = run-remote "! ver" on "ascli1"
? aRT[0].getHostName()
```

See also:

[RemoteTarget.RemoteTarget Constructor](#) (see page 147)

[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)

RemoteTarget.hasCompleted Method

This method checks the execution state of a remote script evaluation. Execution is considered complete if the abort() call ends execution, if an error occurs during processing, or if the script is successfully run on the target node. Find out if execution completes successfully or an error occurs using the errorOccurred() and getError() methods.

This method has the following syntax:

```
hasCompleted()
```

The method returns the following values:

true

Execution complete.

false

Execution is not complete.

Example

Manually implement a wait for completion.

```
aRT = run-remote "! ver" on "ascli1"
while(!aRT[0].hasCompleted())
{
    sleep(500);
}
if(aRT[0].errorOccurred())
{
    ? "Error"
}
else
{
    ? "Result:", get-remoteResult(aRT[0])
}
}
```

Usually scripts only use the -wait option of the run-remote command.

See also:

[RemoteTarget.abort Method](#) (see page 148)

[RemoteTarget.errorOccurred Method](#) (see page 151)

[RemoteTarget.getError Method](#) (see page 152)

[run-remote Command--Execute a Script on Remote Systems \(Funclet\)](#) (see page 94)

RemoteTarget.onError Method

Event handlers are optional functions that can be implemented in user scripts and called by AutoShell when certain events occur. If present, the event handler is called when an error occurs during remote execution.

This method has the following syntax:

```
onError(e)
```

Note: The AutoShell ignores event handler return values.

Example

Specify error handler for a RemoteTarget:

```
rt = new RemoteTarget("ascli1");  
// Assign event handler  
rt.onError = function(e){qout("Error occured, code=", e)};  
run-remote "2*2" on rt
```

See also:

[RemoteTarget.errorOccurred Method](#) (see page 151)

[RemoteTarget.getError Method](#) (see page 152)

RemoteTarget.output Method

Scripts produce two types of result information. The output that is typically written to the screen and a final return value. This method returns a string with the output generated by the script running on the remote system. The output is accumulated while the script is running, so this method can return a non-empty result before script execution completes.

This method has the following syntax:

```
output()
```

The method returns a string with the data written to stdout by the script.

Example

Remotely query list of running services on a Windows host:

```
rt = new RemoteTarget("ascli1", "bob", "casogood42");  
run-remote "! sc query" on rt -wait  
? rt.output()
```

See also:

[RemoteTarget.result Method](#) (see page 158)

RemoteTarget.receivedOutput Method

Event handlers are optional functions that can be implemented in user scripts and called by AutoShell when certain events occur. If present, the event handler is called when output is received from the remote target. This handler can be called multiple times.

This method has the following syntax:

```
receivedOutput(s)
```

Note: The AutoShell ignores event handler return values.

Example

Specify output handler for a RemoteTarget and access associated object in the event handler:

```
rt = new RemoteTarget("ascli1");
// Assign output handler
rt.receivedOutput = function(s){
    qout("Output received=", s);
    qout("Originating host=", this.getHostName());
};
run-remote "! ver" on rt
```

See also:

[RemoteTarget.output Method](#) (see page 156)

RemoteTarget.receivedResult Method

Event handlers are optional functions that can be implemented in user scripts and called by AutoShell when certain events occur. If present, this event handler is called when the final result value is received from the remote target.

This method has the following syntax:

```
receivedResult(s)
```

Note: The AutoShell ignores event handler return values.

Example

Specify result handler for a RemoteTarget and access associated object in the event handler:

```
rt = new RemoteTarget("ascli1");
// Assign output handler
rt.receivedResult = function(s){
    qout("Result received=", s);
    qout("Originating host=", this.getHostName());
};
run-remote "Math.SQRT2" on rt
```

See also:

[RemoteTarget.result Method](#) (see page 158)

RemoteTarget.result Method

Scripts produce two types of result information. The output typically written to the screen and a final return value. This method retrieves a representation of the return value of a remote execution process. Because returning a result is the last step in script evaluation, this method only returns a valid value if script execution successfully completes. `hasCompleted()` returns true and `errorOccurred()` returns false. Because the return value of a remote execution must be transferred over the network to the system that originated the request, it is serialized into an XML representation on the remote system and the `RemoteTarget.result()` method returns the result value in its serialized representation.

Turn the serialized representation into a regular JavaScript value using the `get-remoteResult()` funclet with the `RemoteTarget` object itself as a parameter. Typically there is no need for scripts to call `result()` directly. Serialization maintains type information. For example, if the remote node returns a number, `get-remoteResult()` returns a value of type number. If the remote system returns a Date object, it also becomes a date on the originating system.

Return values from remote scripts are not limited to simple data types. Complex arrays or data-only objects can also be returned, allowing remote execution to transfer large amounts of data between servers. The only limitation is that the data structures must not contain any circular references.

This method has the following syntax:

```
result()
```

The remote script returns a string with a serialized XML representation of the value.

Example

Get serialized representation and actual value of a remote execution:

```
rt = new RemoteTarget("ascli1", "bob", "ca123456");
run-remote "[1,2,3]" on rt -wait
? rt.result()
v = get-remoteResult(rt) // Calls result() internally
? typeof v      // Array
? typeof v[0]   // number
arrdump v
```

RemoteTarget.wasAborted Method

This method checks if remote script execution is aborted programmatically an abort() call.

This method has the following syntax:

```
wasAborted()
```

The method returns the following values:

true

Indicates that the execution has been aborted.

false

Indicates that the execution has not been aborted.

Example

Check if a remote operation is aborted:

```
rt = new RemoteTarget("ascli1");
run-remote Math.sin(3/2*Math.PI) on rt
if(!rt.hasCompleted())
{
    rt.abort()
    ? "Operation aborted:", rt.wasAborted()
}
```

See also:

[RemoteTarget.abort Method](#) (see page 148)

AutoShell Loadable Modules' Command Reference

This section describes the commands which belong to the following optional platform-specific AutoShell Loadable Modules:

- Microsoft Hyper-V
- Oracle Solaris Zones
- VMware vCenter Server

More Information

[CA Oracle Solaris Zones AutoShell Commands](#) (see page 327)

[CA VMware vCenter Server AutoShell Commands](#) (see page 160)

[CA IBM LPAR AutoShell Commands](#) (see page 300)

[CA Microsoft Hyper-V AutoShell Commands](#) (see page 200)

CA VMware vCenter Server AutoShell Commands

You can use the AutoShell to script and automate CA VMware vCenter Server commands and run actions based on the command results. Corresponding commands are also available in the CLI.

dpmvc-addesxhost Command--Add an ESX Host (Funclet)

The `dpmvc-addesxhost` command adds an ESX host to a vCenter Server.

This command has the following format:

```
dpmvc-addesxhost
-esx_host_name esxhostname
-vc_server vcservername
-esx_host_user esxhostuser
-esx_host_password esxhostpassword
[-management_ip managementip]
[-host_connect_port hostconnectport]
[-folder_name foldername]
[-datacenter_name datacentername]
[-cluster_name clustername]
[-vim_account_user vimaccountuser]
[-vim_account_password vimaccountpassword]

-esx_host_name esxhostname
```

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-esx_host_user *esxhostuser*

(Optional) Specifies the user name to access the ESX host.

-esx_host_password *esxhostpassword*

Specifies the password used to access the ESX host.

-management_ip *managementip*

(Optional) Specifies the management IP.

-host_connect_port *hostconnectport*

(Optional) Specifies the port used by ESX host to communicate with vCenter Server.

-folder_name *foldername*

(Optional) Specifies the name of the folder where you can add the ESX host.

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-cluster_name *clustername*

(Optional) Specifies the name of the cluster where you can add the ESX host.

-vim_account_user *vimaccountuser*

(Optional) Specifies the username of the VIM account.

-vim_account_password *vimaccountpassword*

(Optional) Specifies the password to access the VIM account.

Example: Add an ESX Host

This example adds an ESX host, "esx1" to the vCenter Server "myvcenter."

```
dpmvc-addesxhost -esx_host_name esx1 -vc_server myvcenter -esx_host_user admin  
-esx_host_password topsecret
```

dpmvc-addvmvdisk Command--Add Virtual Disk (Funclet)

The dpmvc-addvmvdisk command adds a virtual disk to a VM.

This command has the following format:

```
dpmvc-addvmvdisk
-datastore_name datastorename
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
-capacity_in_mb disk_capacity
[-controller_key controllerkey]
[-disk_mode
{append|independent_nonpersistent|independent_persistent|nonpersistent|persistent
|undoable}]
-thin_provisioning {no|yes}
[-unit_number unitnumber]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-capacity_in_mb *disk_capacity*

Specifies the capacity of the virtual disk in MB.

-controller_key *controllerkey*

(Optional) Specifies the controller key of the disk.

-disk_mode {append | independent_nonpersistent | independent_persistent | nonpersistent | persistent | undoable}

(Optional) Specifies the mode of the disk.

-thin_provisioning {no | yes}

(Optional) Indicates whether the disk must be thin provisioned.

-unit *disk_unitnumber*

(Optional) Specifies the unit number of the disk.

Example: Add a Virtual Disk

This example adds a virtual disk to the VM, "myvm."

```
dpmvc-addvmvdisk -datastore_name disk1 -vm_name myvm -vc_server myvcenterserver
-capacity_in_mb 20000 -thin_provisioning no
```

dpmvc-addvmvnic Command--Add Virtual NIC (Funclet)

The dpmvc-addvmvnic command adds a virtual NIC to a VM.

This command has the following format:

```
dpmvc-addvmvnic
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
-devicetype {e1000 | vmxnet}
-network_name networkname
[-mac_address macaddress]
-wake_on_lan_enabled {no | yes}
```

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-devicetype {e1000 | vmxnet}

Indicates the type of network device.

-network_name *networkname*

Specifies the virtual network the NIC must be connected to. You can distinguish the names of Standard Switches and Distributed Virtual Switches based on the following naming convention:

- For Standard Switches, the name is the network name.
- For Distributed Virtual Switches, the name is a concatenation of the dvPort group name followed by the Distributed Virtual Switch name enclosed in parentheses: dvPortGroupName (dvSwitchName)

-mac_address *macaddress*

(Optional) Specifies the MAC (Media Access Control) address of the network device.

-wake_on_lan_enabled {no | yes}

Indicates whether to start the VM when LAN is enabled.

Example: Add a vNIC

This example adds a virtual NIC to a VM.

```
dpmvc-addvmvnic -vm_name myvm -vc_server myvcenterserver -devicetype vmxnet  
-network_name net1 -wake_on_lan_enabled no
```

dpmvc-clone Command--Clone a VM (Funclet)

The dpmvc-clone command lets you clone a VM.

This command has the following format:

```
dpmvc-clone  
{-template_name templatename | -cloned_from vmname}  
{-datacenter_name datacentername | -vc vcenterservername}  
-vm_name vmname  
-esx_host_name esxhostname  
-datacenter_name datacentername  
-data_store_name datastorename  
-compute_resource_name name  
-resource_pool_name resourcepool  
-spec_name specificationname  
[-auto_deploy value]  
[-deploy_template templatename]  
[-network_connection  
"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_  
gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask[,win_dns_server=wins_dns_  
server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_sec  
ondary"]]  
[-global_dns_search_suffix suffix1[,suffix2,suffix3,...]]  
[-linux_domain_name domain_name]  
[-linux_dns_servers primary_dns=value,secondary_dns=value,tertiary_dns=value]  
[-create_disk MB,datastorename,controller]  
[-modify_disk key, datastorename]  
[-set_memory number]  
[-set_cpu number]  
-vm_os_username username  
[-vm_os_password password]  
[-auth_file authorizationfilename]  
[-auth_comp componentID]  
[-scalability_server scalabilityservername]
```

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-compute_resource_name *name*

Specifies the cluster or VMware ESX host where the VM is created.

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-vm_name *vmname*

Specifies the VM.

-resource_pool_name *resourcepool*

Specifies the name of the resource pool from which you want to select the VM for cloning.

-spec_name *specificationname*

(Optional) Specifies the name of the specification you want to use for the cloned virtual machine.

-auto_deploy {*yes* | *no*}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask,win_dns_server=wins_dns_server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_secondary"

Windows:

(Optional) Creates network connections for a VM. This parameter can be specified multiple times. Sequence number is the integer value starting with one (1) that identifies the network connection. The full set of parameters must be enclosed in parentheses if it includes any spaces. All values except nic, network_name, ip_address, def_gateway and subnet_mask; can be reset by entering the name without a value (for example, "nic=123,alt_gateway=").

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask"

Linux:

(Optional) Creates one or more network connections. This parameter can be specified multiple times. The sequence number is the integer value starting with one (1) that identifies the network connection. The full set of parameters must be enclosed in parentheses if it contains any spaces. Only alt_gateway can be reset by entering the name without a value (for example, "nic=123,alt_gateway=").

-global_dns_search_suffix suffix1{,suffix2,suffix3,...}

Linux:

Specifies one or more DNS search suffixes.

Windows:

(Optional) Specifies one or more DNS search suffixes.

-linux_domain_name domain_name

Linux:

(Optional) Specifies the domain name for a Linux VM.

-linux_dns_servers primary_dns=value{,secondary_dns=value,tertiary_dns=value}

Linux:

(Optional) Specifies the IP addresses of DNS servers for a Linux VM. One or more of the values can be omitted (for example, primary_dns=123.123.123.13,secondary_dns=,tertiary_dns=123.123.123.15) to unset the value.

-create_disk *MB,datastore_name,controller*

(Optional) Creates one or more additional hard drives. This parameter can be specified multiple times.

MB

Defines the size in megabytes of the hard drive.

Datastore_name

Specifies the name of the data store for this hard drive. Currently only one data store is supported per VM.

Controller

Specifies the controller key for this hard drive.

-modify_disk {*key,datastore_name*}

(Optional) Specifies the data store where the hard drive resides. Can be specified multiple times.

key

Specifies an existing disk.

datastore_name

Specifies the name of the data store that this disk moves to.

-set_memory *number*

(Optional) Defines the amount of memory in megabytes (MB) for the cloned VM. This value overrides the value specified in the template.

-set_cpu *number*

(Optional) Define the number of CPUs for the cloned VM. This value overrides the value specified in the template.

-vm_os_username *user name*

Specifies the user for the cloned VM. This user name is also used for authentication when you auto-deploy the image.

Windows: Must be the user name defined in the customization specification.

Linux: Must be the user name defined in the template.

-vm_os_password *password*

(Optional) Specifies the password for the user for the cloned VM. This password is also used for authentication when you auto-deploy the image.

Windows: Must be the same password defined in the customization specification.

Linux: Must be the same password defined in the template.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-template_name *templatename*

Specifies the name of the template you want to use for the cloned virtual machine. This parameter replaces `-vm_name_cloned_from` and you cannot specify both together.

-vm_name_cloned_from *vmname*

Specifies the name of the virtual machine to use as a template for the cloned virtual machine. This parameter replaces `-template_name` and you cannot specify both together.

-resource_pool_name *resourcepool*

Specifies the name of the resource pool from which you want to select the VM for cloning.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-spec_name *specificationname*

(Optional) Specifies the name of the specification you want to use for the cloned virtual machine.

-compute_resource_name *name*

Specifies the cluster or VMware ESX host where the VM is created.

Example: Clone a VM using Global Credentials

This example creates a VM, "testvm01" using the template "BaseW2k3" on the data center, Specifies VAS/MyCity. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword
```


Example: Clone a VM using Authorization File and Component

This example creates a VM, "testvm01" using the template "BaseW2k3" on the data center, VAS/MyCity. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name Specifies VAS/MyCity -datastore_name
storage1
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -auth_file c:\localauth.dat -auth_comp Imaging
```

Example: Clone a VM using Global Credentials and Create 2 CPUs

This example creates a VM, "testvm01" using the template "BaseW2k3" on the data center, VAS/MyCity and creates 2 CPUs for the VM. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -set_cpu 2
```

Example: Clone a VM using Default Authorization File and Set Memory to 4096 MB

This example creates a VM, "testvm01" using the template "BaseW2k3" on the data center, VAS/MyCity and sets the memory to 4096. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -set_memory 4096
```

Example: Clone a VM using Default Authorization File and Create Two Hard Disks

This example creates a VM, "testvm01" using the template "BaseW2k3" on the data center, VAS/MyCity and creates two hard disks for data store storage1. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VASCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000
-create_disk 10000,storage1,2001
```

Example: Clone a VM and Create a Network Connection

This example creates a VM, "testvm01", a network connection using the template "BaseW2k3" on the data center, VAS/MyCity, and sets the initial values. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VASCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -network_connection "nic=1,network_name=VM
Network,ip_address=123.321.5.22,def_gatewat=123.321.1.108,subnet_mask=255.255.255
.0,win_dns_server=123.321.42.1,win_alt_dns=123.321.3.101,wins_primary=123.321.3.3
"
```

Example: Clone a VM using Default Authorization File from a Powered off VM

This example creates a VM, "testvm01" using VM "testvm02" in place of a template on the data center, VAS/MyCity. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -datacenter_name "VAS/MyCity" -vm_name testvm01 -datastore_name storage1
-esx_host_name vc1.ca.com -resource_pool_name Resources/DPMTTest
-compute_resource_name DPMCluster/ussdCluster -vm_name_cloned_from testvm02
-auto_deploy yes -spec_name w2k3_spec_01 -vm_os_username administrator
-vm_os_password mypassword
```

Example: Clone a VM using Default Authorization File and Modify an Existing Hard Disk

This example creates a VM, "testvm01" using the template "BaseW2k3" on the data center, VAS/MyCity and creates two hard disks for data store storage1. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000 -modify_disk
2000,storage2
```

dpmvc-createdatastore Command--Create a Datastore (Funclet)

The dpmvc-createdatastore command creates a datastore.

This command has the following format:

```
dpmvc-createdatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
-datastore_type {local|nas|vmfs}
[-vmfs_device_path path]
[-vmfs_block_size_mb blocksize]
[-local_path lpath]
[-nas_user_name nasuser]
[-nas_password naspasswd]
[-nas_remote_hostname nasremhost]
[-nas_remote_path nsarempath]
[-nas_access_mode {read-only|read-write}]
[-sc sc_url]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-datastore_type {vmfs|nas|local}

Specifies the type of datastore to create, VMFS, NAS, or local.

-vmfs_device_path *path*

(Optional) Specifies the device path of the VMFS disk to use when creating a datastore. You can obtain the path using the "getavailablescsidisks" command.

-vmfs_block_size_mb *blocksize*

(Optional) Specifies the block size to use when creating a datastore, in MB. Block sizes available are 1, 2, 4 and 8. Block sizes enable VM disk files to reach a maximum of 256GB, 512GB, 1024GB, and 2048GB respectively.

-local_path *lpath*

(Optional) When creating a local datastore, this path specifies the ESX host local file system path that is used to create the datastore.

-nas_user_name *nasuser*

(Optional) Specifies the remote host user name for NAS datastore creation.

-nas_password *naspasswd*

(Optional) Specifies the remote host password for NAS datastore creation.

-nas_remote_hostname *nasremhost*

(Optional) Specifies the hostname of the server hosting the network-based storage.

-nas_remote_path *nasrempath*

(Optional) Specifies the file system path on the remote server used for the NAS datastore.

-nas_access_mode *nasaccmode*

(Optional) Specifies the desired access to the network-based storage used for the NAS datastore. Available modes are "read-only" and "read-write".

Default: "read-write"

dpmvc-cycle Command--Cycle a VM (Funclet)

The dpmvc-cycle command powers on, powers off, resets, or suspends a VM.

This command has the following format:

```
dpmvc-cycle  
-powerop {poweron|poweroff|reset|suspend|shutdown guest}  
{-datacenter_name datacentername | -vc vcservername}  
-vm_name vmname
```

-powerop {poweron|poweroff|reset|suspend|shutdown guest}

Specifies the power operation to perform on the VM. Options include the following:

poweron

Powers on the VM.

poweroff

Powers off the VM.

reset

Resets the VM.

suspend

Temporarily suspends the VM.

shutdown guest

Shuts down the guest OS gracefully.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-vm_name *vmname*

Specifies the VM.

Example: Turn off a VM when there are Multiple vCenter Servers in the Data Center

This example powers off a VM from a specific vCenter Server.

```
dpmvc-cycle -powerop poweroff -vc my_server1 -vm_name my_vm
```

Example: Turn on a VM in Single vCenter Server Environment

This example powers off the VM, "testvm" that belongs to the data center "VAS/MyCity."

```
dpmvc-cycle -vm_name testvm -powerop poweron -datacenter_name VAS/MyCity
```

dpmvc-datastore Command--Get Data Store Properties (Funclet)

The dpmvc-datastore command retrieves the free space and capacity settings for a specific VMware vCenter data store.

This command has the following format:

```
dpmvc-datastore
{-datacenter_name datacentername | -vc vcenterservername}
-datastore_name datastorename
[-getproperty {all|capacity|freespace}]
```

-getproperty {all|capacity|freespace}

(Optional) Specifies which property to retrieve. Options include the following:

all

Retrieves capacity and free space in the data store.

capacity

Retrieves the capacity in the data store.

freespace

Retrieves the free space in the data store.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

Example: Get all Properties for the Data Store

This example retrieves both free space and capacity for the data store.

```
dpmvc-datastore -getproperty all -datacenter_name VAS/MyCity -datastore_name storage1
```

dpmvc-delete Command--Destroy a VM (Funclet)

The dpmvc-delete command deletes a VM that is in a powered off state. Use this command to clean up and free unused resources.

Important! Verify that you back up any important data *before* you issue this command. This command deletes the VM data store and data files for the VM, including the disk image.

This command has the following format:

```
dpmvc-delete  
{-datacenter_name datacentername | -vc vcenterservername}  
-vm_name vmname
```

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-vm_name *vmname*

Specifies the VM.

Example: Delete a VM in a Single vCenter Server Environment

This example deletes the VM "vm11."

```
dpmvc-delete -vm_name vm11 -datacenter_name lab444
```

Example: Delete a VM in a Multiple vCenter Server Environment

This example deletes the VM "vm11."

```
dpmvc-delete -vm_name vm11 -datacenter_name lab444 -vc vc_server_1
```

dpmvc-distributedswitch Command--Manage Virtual Distributed Switches (Funclet)

The dpmvc distributedswitch command lets you manage virtual distributed switches.

- Add a new virtual distributed switch to the datacenter
- Update the properties of a virtual distributed switch
- Delete a virtual distributed switch
- Create a new distributed port group to a virtual distributed switch
- Update the port group properties of a virtual distributed switch
- Remove a distributed port group from a virtual distributed switch
- Rename a distributed port group of a virtual distributed switch

The command has the following formats:

```
dpmvc-distributedswitch {-vds_add | -vds_update}
-vc_server vcservername
-datacenter_name datacentername
-vds_folder vdsfolder
-switch_name switchname
[-hostnics hostname1:nic1,nic2,...nicn [;hostname2:nic2,...nicn >]*]
[-uplink_port_names uplink1[,uplink2,...,uplinkn]]
[-maxports maxports]
```

```
dpmvc-distributedswitch -vds_remove
-vc_server vcservername
-switch_name switchname
```

```
dpmvc-distributedswitch -add_portgroup
-vc_server vcservername
-switch_name switchname
-portgroup_name portgroupname
[-bindtype earlyBinding | ephemeral | lateBinding]
[-vlan vlanID]
[-numports numberofports]
```

```
dpmvc-distributedswitch -update_portgroup
-vc_server vcservername
-portgroup_name portgroupname
[-portgroup_newname portgroupnewname]
[-bindtype earlyBinding | ephemeral | lateBinding]
[-vlan vlanID]
[-numports numberofports]
```

```
dpmvc-distributedswitch -remove_portgroup
-vc_server vcservername
-portgroup_name portgroupname
```

```
dpmvc-distributedswitch -rename_portgroup
-vc_server vcservername
-portgroup_name portgroupname
-portgroup_newname portgroupnewname
```

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-datacenter_name *datacentername*

Specifies the datacenter to which the virtual distributed switch belongs.

-vds_folder *vdsfolder*

Specifies the folder of the virtual distributed switch in the CA Server Automation Explorer pane.

-switch_name *switchname*

Specifies the switch name to perform the operation on.

-hostnics *hostname1:nic1,nic2,...nicn* [*;hostname2:nic2,...nicn* >]

(Optional) Specifies lists of NICs associated with the ESX host members.

-uplink_port_names *uplink1[,uplink2,...,uplinkn]*

(Optional) Specifies a comma-separated list of uplink port names to use.

-maxports *maxports*

(Optional) Specifies the maximum number of ports for the switch. This number limits the total number of ports (numports) from all of the portgroups on the switch.

-bindtype *earlyBinding* | *ephemeral* | *lateBinding*

(Optional) Specifies the bind type of the port group. Valid values are:

earlyBinding

Assigns the ports when the VM binds to the portgroup. This type of binding ensures connectivity at all times, but permanently reserves the port. This binding type is the default.

lateBinding

Assigns a port to a VM if the VM is powered on and its NIC is in connected state. This binding type reassigns the port when the VM is powered off or its NIC is disconnected. LateBinding is configurable through vCenter.

ephemeral

Assigns a port to a VM if the VM is powered on and its NIC is in connected state. This binding type reassigns the port when the VM is powered off or its NIC is disconnected. Ephemeral binding is configurable through the ESX Host and vCenter.

-numports *numberofports*

(Optional) Specifies the number of ports of the port group.

-portgroup_name *portgroupname*

Specifies the port group name.

-portgroup_newname *portgroupnewname*

Specifies the new port group name.

-vlan *vlanID*

(Optional) Specifies an Integer value (vlan ID) used for the virtual portgroup operations.

Example: Create a New Virtual Distributed Switch

This example creates a new virtual distributed switch.

```
dpmvc-distributedswitch -vds_add -vc_server vc5master -datacenter_name dc3  
-vds_folder vds -switch_name vds1
```

Example: Updates an Existing Virtual Distributed Switch

This example updates a virtual distributed switch. It specifies NICs for MYSERVER1, and removes MYSERVER2 from the virtual distributed switch. When you want to delete a host from a virtual distributed switch, specify the servername without NICs assigned to it (server name followed by colon, for example, MYSERVER2:).

```
dpmvc-distributedswitch -vds_update -vc_server VAS-VC5 -datacenter_name VC5  
-switch_name vdistSwitch -hostnics MYSERVER1:nxmg2,tmp2;MYSERVER2:
```

The command updates vdistSwitch:

- Host MYSERVER1 is using NICs nxmg2 and tmp2
- Host MYSERVER2 is removed from the switch

dpmvc-entermaintenancemode Command--Set to the Maintenance Mode (Funclet)

The dpmvc-entermaintenancemode command sets an ESX host to the maintenance mode

This command has the following format:

```
dpmvc-entermaintenancemode  
-esx_host_name esxhostname  
-vc_server vcservername  
-timeout timeout  
[-evacuate_powered_off_vm {no | yes}]
```

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-timeout *timeout*

Specifies the timeout time to enter or exit the maintenance mode in seconds.

-evacuate_powered_off_vm {no | yes}

(Optional) Specifies whether to exit after the VM stops.

Example: Enter Maintenance Mode

This example sets an ESX host into maintenance mode.

```
dpmvc-entermaintenancemode -esx_host_name esx1 -vc_server myvcserver -timeout 60
```

dpmvc-exitmaintenancemode Command--Exit the Maintenance Mode (Funclet)

The dpmvc-exitmaintenancemode command sets the ESX host out of the maintenance mode.

This command has the following format:

```
dpmvc-exitmaintenancemode  
-esx_host_name esxhostname  
-vc_server vcservername  
-timeout timeout
```

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-timeout *timeout*

Specifies the timeout time to enter or exit the maintenance mode in seconds.

Example: Exit the Maintenance Mode

This example sets the esx1 ESX host out of maintenance mode.

```
dpmvc-exitmaintenancemode -esx_host_name esx1 -vc_server myvcserver  
-timeout 60
```

dpmvc-faulttolerance Command--Specify Fault Tolerant Operations (Funclet)

The dpmvc-faulttolerance command turns on, turns off, enables, disables, tests fault tolerance, or migrates the secondary VM.

This command has the following format:

```
dpmvc-faulttolerance
-turn_on -vm_name <value> [-esx_host_name <value>]|
-turn_off -vm_name <value>|
-enable -vm_name <value>|
-disable -vm_name <value>|
-test_ft -vm_name <value>|
-migrate_secondary -vm_name <value> -esx_host_name <value>
```

-turn_on

Turns on fault tolerance for the specified VM.

(Optional) Specifies the ESX host name of the secondary VM.

-turn_off

Turns off fault tolerance for the specified VM.

Note: This operation deletes the secondary VM.

-enable

Enables fault tolerance for the specified VM.

-disable

Disables fault tolerance for the specified VM.

-test_ft

Tests fault tolerance on the specified VM.

-migrate_secondary

Migrates the fault tolerance secondary VM to another ESX server. The -esx_host_name option is required.

-vm_name

Defines the name of computer resource.

-esx_host_name

Specifies the ESX server where the VM resides.

-vc_server

Specifies virtual center server host name.

dpmvc-getavailablescsidisks Command--Get Available SCSI Disks (Funclet)

The dpmvc-getavailablescsidisks command lists the available SCSI disks.

This command has the following format:

```
dpmvc-getavailablescsidisks  
[-datastore_name datastorename]  
-esx_host_name esxhostname  
-vc_server vcservername  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

dpmvc-gethosthba Command--Get Host Bus Adapters (Funclet)

The dpmvc-gethosthba command retrieves the list of host bus adapters configured on the specified ESX host.

This command has the following format:

```
dpmvc-gethosthba
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
```

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

dpmvc-getresources Command--Get VM Resources (Funclet)

The dpmvc-getresources command retrieves CPU or memory share limits or reservations.

This command has the following format:

```
dpmvc-getresources
-vm_name vmname
{-datacenter_name datacentername | -vc vcservername}
[-resource {all|vm_cpulimit|vm_cpureserv|vm_memlimit|vm_memresrv}]
```

-vm_name *vmname*

Specifies the VM.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-resource {*all*|*vm_cpulimit*|*vm_cpureserv*|*vm_memlimit*|*vm_memresrv*}

Specifies whether a specific resource is retrieved or all resources. Options include the following:

all

Retrieves all CPU and memory share limits and reservations for the VM.

vm_cpulimit

Retrieves the limit for the number of shares of CPU for the VM.

vm_cpureserv

Retrieves the number of shares of CPU reserved for the VM.

vm_memlimit

Retrieves the limit for the number of shares of memory for the VM.

vm_memresrv

Retrieves the number of shares of memory reserved for the VM.

Example: Get all Resources for a VM in a Single vCenter Server Environment

This example retrieves all resources for usa-vm2.

```
dpmvc-getresources -resource all -datacenter_name MyCity -vm_name usa-vm2
```

dpmvc-getshares Command--Get VM Shares (Funclet)

The dpmvc-getshares command lets you view how many shares of CPU or memory are allocated to a VM.

This command has the following format:

```
dpmvc-getshares  
{-datacenter_name datacentername | -vc vcenterservername}  
-vm_name vmname  
[-share {cpu | memory}]
```

-share {cpu | memory}

Displays how many shares of CPU or memory are allocated to a VM. Shares are used to determine which VMs are given more of the available physical resources proportionally to other running VMs. If you allocate more shares to a VM, more physical resources are given to that VM. If you allocate fewer shares to a VM, fewer physical resources are given to that VM.

cpu

Specifies that CPU values are being retrieved.

memory

Specifies that memory values are being retrieved.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-vm_name *vmname*

Specifies the VM.

Example: Get the CPU Values for a VM

This example retrieves CPU values for usa-vm2.

```
dpmvc-getshares -resource cpu -datacenter_name MyCity -vm_name usa-vm2
```

dpmvc-getsnapshots Command--List Snapshots of a VM (Funclet)

The dpmvc-getsnapshots command lists the snapshots of a VM.

This command has the following format:

```
dpmvc-getsnapshots  
{-datacenter_name datacentername | -vc vcenterservername}  
-vm_name vmname  
[-name_only]
```

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-vm_name *vmname*

Specifies the VM.

-name_only

(Optional) Provides snapshot names only.

Example: List Snapshots

This example lists the snapshots of a VM.

```
dpmvc-getsnapshots -vc my_server1 -vm_name my_vm
```

dpmvc-imgjobcheck Command--Get Clone Job Status (Funclet)

The dpmvc-imgjobcheck command obtains the cloning job status for a specific CA Server Automation job ID or a specific vCenter task ID. The job ID is provided for certain dpmvc commands. Use this command to check the status on vCenter.

This command has the following format:

```
dpmvc-imgjobcheck
-status job ID
[-vc vccenterservername]
```

-status *jobID*

Specifies the job ID used to obtain the job status.

-vc *vccenterservername*

Specifies the vCenter Server where the VM is located.

Example: Get the Cloning Job Status Using the Job ID in a Single vCenter Server Environment

This example obtains the cloning job status using a CA Server Automation job ID.

```
dpmvc-imgjobcheck -status 42
```

dpmvc-migrate Command--Migrate a VM (Funclet)

The dpmvc-migrate command migrates a VM from one host server to another host server.

This command has the following format:

```
dpmvc-migrate
{-datacenter_name datacentername | -vc vccenterservername}
-vm_name vm
-migrate_to_target_host name
-migrate_to_target_resource_pool_name name
-migrate_to_compute_resource_name name
```

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcservername*

Specifies the vCenter Server where the VM is located.

-vm_name *vmname*

Specifies the VM.

-migrate_to_target_host *name*

Specifies the name of the host to which you are migrating the VM.

-migrate_to_target_resource_pool_name *name*

Specifies the target resource pool.

-migrate_to_compute_resource_name *name*

Specifies the name of the cluster or VMware ESX host to which you are migrating the VM.

Example: Migrate a VM

This example migrates the VM, "MyVM1", from the data center VPM/MyCity to ServerC-dcavc.

```
dpmvc-migrate -vm_name MyVM1 -datacenter_name VAS/MyCity
-migrate_to_compute_resource_name VASCluster/ClusterA
-migrate_to_target_resource_pool_name Resources/VASTest
-migrate_to_target_host ServerC-dcavc.MyCompany.com
```

dpmvc-removedatastore Command--Remove a Datastore (Funclet)

The dpmvc-removedatastore command deletes a datastore.

This command has the following format:

```
dpmvc-removedatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

dpmvc-removevmdisk Command--Remove Virtual Disk (Funclet)

The dpmvc-removevmdisk command removes a virtual disk from a VM.

This command has the following format:

```
dpmvc -removevmdisk  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-disk_device_key diskdevicekey  
-delete_data {yes | no}
```

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-disk_device_key *diskdevicekey*

Integer that specifies the device key of the disk, for example, 2000.

-delete_data {yes | no}

Indicates whether to delete data while deleting a disk.

Example: Remove a Virtual Disk

This example removes virtual disk with the device key of 2000 from the myvm VM.

```
dpmvc-removevmdisk -vm_name myvm -vc_server myvcsrvr -disk_device_key 2000  
-delete_data yes
```

dpmvc-removevmnic Command--Remove Virtual NIC (Funclet)

The dpmvc-removevmnic command removes a virtual NIC from a VM.

This command has the following format:

```
dpmvc removevmnic  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-device_key devicekey
```

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-device_key *devicekey*

Integer that specifies the device key of the network interface, for example, 3000.

Example: Remove a vNIC

This example removes a vNIC from a VM.

```
dpmvc-removevmnic -vm_name myvm -vc_server myvcsrvr -device_key 3000
```

dpmvc-setresources Command--Set VM Resources (Funclet)

The dpmvc-setresources command lets you adjust CPU or memory share limits or reservations.

This command has the following format:

```
dpmvc-setresources  
-resource {setcpulimit|setcpureserv|setmemlimit|setmemresrv}  
-value value  
{-datacenter_name datacentername | -vc vcservername}  
-vm_name vmname
```

setcpulimit

Defines the limit for the number of shares of CPU for the VM.

setcpureserv

Defines the number of shares of CPU reserved for the VM.

setmemlimit

Defines the limit for the number of shares of memory for the VM.

setmemresrv

Defines the number of shares of memory reserved for the VM.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-vm_name *vmname*

Specifies the VM.

-value *value*

Specifies the value for the resource parameter.

Example: Set Resource CPU Limit to 1000 Shares in a Single vCenter Server Environment

This example adjusts the CPU share limit for the virtual machine "MyVM1" to 1000.

```
dpmvc-setresources -resource setcpulimit -value 1000 -vm_name MyVM1 -datacenter_name VAS/MyCity
```

dpmvc-setshares Command--Set VM Shares (Funclet)

The dpmvc-setshares command lets you set how many shares of CPU or memory are allocated to a VM.

This command has the following format:

```
dpmvc-setshares
{-datacenter_name datacentername | -vc vcenterservername}
-vm_name vmname
-setshare {cpu_add_prop_value | cpu_subtract_prop_value |
cpu_overwrite_prop_value | mem_add_prop_value | mem_subtract_prop_value |
-mem_overwrite_prop_value}
-value value
```

**-setshare {cpu_add_prop_value | cpu_subtract_prop_value |
cpu_overwrite_prop_value | mem_add_prop_value | mem_subtract_prop_value |
mem_overwrite_prop_value}**

Defines how many shares of CPU or memory to add, subtract, or change. Shares are used to determine which VMs are given more of the available physical resources proportionally to other running VMs. If you allocate more shares to a VM, more physical resources are given to that VM. If you allocate fewer shares to a VM, fewer physical resources are given to that VM.

cpu_add_prop_value

Defines the number of shares of CPU to add to the VM.

-cpu_subtract_prop_value value

Defines the number of shares of CPU to subtract from the VM.

-cpu_overwrite_prop_value value

Defines the number of shares of CPU to use for the VM.

-mem_add_prop_value value

Defines the number of shares of memory to add to the VM.

-mem_subtract_prop_value value

Defines the number of shares of memory to subtract from the VM.

-mem_overwrite_prop_value value

Defines the number of shares of CPU to use for the VM.

-datacenter_name datacentername

Specifies the data center where the VM is located.

-vc vcenterservername

Specifies the vCenter Server where the VM is located.

-vm_name vmname

Specifies the VM.

-value value

Specifies the value for the resource parameter.

Example: Increase CPU Shares for a VM in a Single vCenter Server Environment

This example increases the CPU shares for the virtual machine "MyVM01" by 1000 in the data center "VAS/MyCity."

```
dpmvc-setshares -vm_name MyVM01 -datacenter_name VAS/MyCity -setshare  
cpu_add_prop_value -value 1000
```

dpmvc-snapshot Command--Manage Snapshots (Funclet)

The dpmvc-snapshot command lets you manage snapshots for a VM. You can create snapshots, delete one or all snapshots, or revert to a snapshot using this command.

This command has the following format:

```
dpmvc-snapshot
-create
{-datacenter_name datacentername | -vc vcenterservername}
-vm_name vmname
-vm_snapshot_name vmsnapshotname
[-vm_snapshot_desc description]
[-withmemory {true|false}]
[-poweron]
```

```
dpmvc-snapshot
-remove
{-datacenter_name datacentername | -vc vcenterservername}
-vm_name vmname
[-vm_snapshot_name vmsnapshotname [-withchildren]]
```

```
dpmvc-snapshot
-revert
{-datacenter_name datacentername | -vc vcenterservername}
-vm_name vmname
-vm_snapshot_name vmsnapshotname
```

-operation {create|remove|revert}

Specifies to create, remove, or revert a VM snapshot.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vm_snapshot_name *vmsnapshotname*

Defines a name for the VM snapshot.

-vm_snapshot_desc *description*

(Optional) Defines a description for the VM snapshot.

-withmemory {true | false}

(Optional) Specifies that the snapshot includes memory. This parameter is invalid for -remove or -revert.

-withchildren

(Optional) Specifies that you want to remove all children of the snapshot. This parameter is invalid for -create or -revert.

-poweron

(Optional) Specifies that you want to power on the VM. This parameter is invalid for -remove or -revert.

Example: Create a VM Snapshot in a Single vCenter Server Environment

This example creates a snapshot named "Hello world" with a description of "This is my first snapshot" for the VM system named MyVM01.

```
dpmvc-snapshot -create -vm_snapshot_name "Hello world" -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity
```

Example: Revert to a VM Snapshot in a Single vCenter Server Environment

This example reverts MyVM01 to the snapshot named "Hello world".

```
dpmvc-snapshot -revert -vm_snapshot_name "Hello world" -vm_name MyVM01
-datacenter_name VAS/MyCity
```

Example: Remove a VM Snapshot in a Single vCenter Server Environment

This example deletes the snapshot named "Hello world" from a VM system named MyVM01.

```
dpmvc-snapshot -remove -vm_snapshot_name "Hello world" -vm_name MyVM01
-datacenter_name VAS/MyCity
```

dpmvc-templatetovm Command--Convert Template to VM (Funclet)

The dpmvc-templatetovm command converts virtual machine templates back to VMs. This capability allows you to apply patches or software updates to the converted VM and then convert it back to a template, for example.

This command has the following format:

```
dpmvc-templatetovm
-template_name templatename
{-datacenter_name datacentername | -vc vccenterservername}
-compute_resource_name hostname
-resource_pool_name resourcepool
-esx_host_name hostname
```

-template_name *templatename*

Defines the name of the template.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

-compute_resource_name *name*

Specifies the cluster or VMware ESX host where the VM is created.

-resource_pool_name *resourcepool*

Specifies the name of the resource pool from which you want to select the VM for cloning.

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

Example: Convert a Template into a VM in a Single vCenter Server Environment

This example converts the template named convertTest into a VM.

```
dpmvc-templatetovm -datacenter_name VAS/MyCity -template_name convertTest  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA
```

dpmvc vapp Command--Manage vApp

The dpmvc vapp command supports the following operations on vApps:

- Create New vApp
- Clone
- Power On
- Power Off
- Suspend
- Delete vApp from VMware vCenter
- Unregister from VMware vCenter
- Add VMs to vApp
- Add resource pool to vApp
- Add vApps to a vApp
- Update vApp configuration

The command has the following formats:

```
dpmvc-vapp -create
```

```
-vc_server vcservername
```

```
-vapp_path vapppath
```

```
[-vapp_name vappname]
```

```
[-cpu_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
```

```
[-mem_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
```

```
dpmvc-vapp -clone
```

```
-vc_server vcservername
```

```
-vapp_path vapppath
```

```
[-target_vapp_path targetvapppath]
```

```
[-vapp_name vappname]
```

```
[-target_vapp_host targetvapphost]
```

```
[-target_vapp_datastore targetvappdatastore]
```

```
dpmvc-vapp [-power_on | -power_off | -suspend | -delete | -unregister ]
```

```
-vc_server vcservername
```

```
-vapp_path vapppath
```

```
dpmvc-vapp -add_vms_to_vapp
```

```
-vc_server vcservername
```

```
-vapp_path vapppath
```

```
[-vapp_name vappname]
```

```
[-vms vms]
```

dpmvc-vapp -add_rps_to_vapp

-vc_server *vcservername*
-vapp_path *vapppath*
[-vapp_name *vappname*]
[-rps *resourcepool*]

dpmvc-vapp -add_vapps_to_vapp

-vc_server *vcservername*
-vapp_path *vapppath*
[-vapp_name *vappname*]
[-vapps *vapps*]
[-locale *iso639value*]

dpmvc-vapp -update_vapp_config

-vc_server *vcservername*
-vapp_path *vapppath*
[-vapp_name *vappname*]
[-config_settings
vappName, objectType, startAction, startDelay, startOrder, stopAction, stopDelay, waitForGuest]
[-cpu_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*]
[-mem_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*]
[-locale *iso639value*]

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-vapp_path *vapppath*

Specifies the vApp path.

-vapp_name *vappname*

(Optional) Specifies the vApp name.

-cpu_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

(Optional) Specifies the CPU allocation settings when updating a vApp configuration.

isExpandableReservation

Specifies if the allocation is expandable. Possible values: true or false.

limit

Specifies the allocation limit.

reservation

Specifies the reservation allocation.

sharesLevel

Specifies the shares level that is 0 for low, 1 for normal, 2 for high and 3 for custom. e.g. true,-1,1000,1,4000

shares

Specifies the allocation shares.

-mem_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

Specifies the memory allocation settings when updating a vApp configuration.

isExpandableReservation

Specifies if the allocation is expandable. Possible values: true or false.

limit

Specifies the allocation limit.

reservation

Specifies the reservation allocation.

sharesLevel

Specifies the shares level that is 0 for low, 1 for normal, 2 for high and 3 for custom. e.g. true,-1,1000,1,4000

shares

Specifies the allocation shares.

-vms *vms*

(Optional) Specifies the VMs you want to add to vApp.

-rps *resourcepool*

(Optional) Specifies the name of the resource pool which you want to add to a vApp.

-vapps *vapps*

(Optional) Specifies the vApps you want to add to a vApp.

-config_settings

vappName, objectType, startAction, startDelay, startOrder, stopAction, stopDelay, waitForGuest

(Optional) Specifies the configuration settings of the vApp. Proper format is "

startAction none | powerOn

Specifies the start action. The available options are none or powerOn

stopAction none | powerOff | guestShutdown | suspend

Specifies the start action. The available options are none or powerOff or guestShutdown or suspend.

-target_vapp_path targetvapppath

Specifies the target vApp path of the vApp you want to create.

-target_vapp_host targetvapphost

(Optional) Specifies the host where the new vApp is to reside on.

-target_vapp_datastore targetvappdatastore

(Optional) Specifies the name of the data store where the vApp is to reside on.

dpmvc-virtualswitch Command--Manage Virtual Switches (Funclet)

The dpmvc-virtualswitch command lets you manage virtual switches.

- Create a virtual switch
- Update the properties of a virtual switch
- Delete a virtual switch
- Create a port group for a virtual switch
- Update the port group properties of a virtual switch
- Remove a port group from a virtual switch
- Rename a port group of a virtual switch

The command has the following formats:

```
dpmvc-virtualswitch {-vs_add | -vs_update}
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-nic_names nicname1 [,nicname2, ...]
```

```
dpmvc-virtualswitch -vs_remove
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
```

```
dpmvc-virtualswitch -add_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-portgroup_name portgroupname  
[-vlan vlanID]
```

```
dpmvc-virtualswitch -update_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-portgroup_name portgroupname  
[-portgroup_newname portgroupnewname]  
[-vlan vlanID]
```

```
dpmvc-virtualswitch -remove_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-portgroup_name portgroupname  
[-vlan vlanID]
```

```
dpmvc-virtualswitch -rename_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-portgroup_name portgroupname  
-portgroup_newname portgroupnewname
```

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-switch_name *switchname*

Specifies the switch name to perform the operation on.

-nic_names *nicname1* [*nicname2*, ...]

Specifies a list of physical NICs separated by a comma. If you specify multiple NICs, use double quotes to escape the argument, for example: "sc1,nfs1,mnic1,mnic2".

-portgroup_name *portgroupname*

Specifies the port group name.

-portgroup_newname *portgroupnewname*

Specifies the new port group name.

-vlan *vlanID*

(Optional) Specifies an Integer value (vlan ID) used for the virtual portgroup operations.

Example

This example creates a new virtual switch.

```
dpmvc-virtualswitch -vs_add -vc_server vc5master -esx_host_name esx5
-switch_name switch1 -nic_names "sc1,nfs1,mnic1,mnic2"
```

dpmvc-vmtemplate Command--Convert VM to Template (Funclet)

The dpmvc-vmtemplate command converts virtual machines to templates. This capability helps you to enforce consistency across multiple servers in the data center.

This command has the following format:

```
dpmvc-vmtemplate
-vm_name vmname
{-datacenter_name datacentername | -vc vcenterservername}
```

-vm_name *vmname*

Specifies the VM.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc *vcenterservername*

Specifies the vCenter Server where the VM is located.

Example: Convert a VM into a Template in a Single vCenter Server Environment

This example converts the VM named convertTest test into a template.

```
dpmvc-vmtemplate -datacenter_name VAS/MyCity -vm_name convertTest
```

dpmvc-wait Command--Time to Wait for a Job (Funclet)

The dpmvc-wait command specifies the time to wait until a job is performed. This command does not return the status until the job is complete or timeout occurs.

This command has the following format:

```
dpmvc-wait
-job <jobid>
[-timeout timeout]
```

-job *jobid*

Specifies the Job Id.

-timeout *timeout*

Specifies the timeout value to specify the job status.

Example: Status of Job

This example specifies that job 125635 waits 20 seconds until it is performed.

```
dpmvc-wait -job 125635 -timeout 20
```

CA Microsoft Hyper-V AutoShell Commands

You can use the AutoShell to script and automate CA Hyper-V AutoShell Commands and run actions based on the command results. Corresponding commands are also available in the CLI.

Note: The *vmname* option is not unique in the Hyper-V server environment. You can specify a VM by *vmname* for convenience. However, if the *vmname* is not unique, the command fails. In this case you must use the *vmid* option instead. You cannot use the following parameter pair in an AutoShell command: *vmname*, *vmid* and *template*, *templateid*. For example, if you use *vmname*, you cannot use *vmid*.

When DCOM/WMI Port is dynamically assigned during RPC Endpoint negotiation, see The default dynamic port range for TCP/IP <http://support.microsoft.com/kb/929851> for more information.

dpmhv-addVMNic (Cmdlet) Command--Add a Network Interface Controller to a VM

The dpmhv-addVMNic command lets you add network adapter to a VM and provides an option to connect it to a virtual switch.

This command has the following format:

```
dpmhv-addVMNic  
-host hostname  
-vm vmname | -vmid vmguid  
[-mac_address macaddress]  
[-switch switchname]  
[-legacy]  
[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-mac_address *macaddress*

(Optional) Defines the MAC address of the new network adapter. Valid entry: 12 digit hexadecimal values (for example, A-F, 0-9, a-f) in the format: 0013724C2140 or 00:13:72:4C:21:40.

Note: If the macaddress is not specified, Hyper-V automatically generates it.

-switch *switchname*

Specifies the virtual network switch name to connect the network adapter of the VM. If you do not specify this option, the network adapter of the VM is not connected to the virtual network.

Default: None

-legacy

(Optional) Creates legacy network adapter instead of Hyper-V synthetic adapter.

Note: Use this option when installing operating systems that do not have Hyper-V integration services.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Add a Network Interface Controller to a VM

This example adds NIC to a VM, "TestVM."

```
dpmhv -addVMNic
-host hvserver
-vm TestVM
-switch "Local Area Connection
- Virtual Network"
```

dpmhv-addVMSCSIController (Cmdlet) Command--Add SCSI Controllers to a VM

The `dpmhv-addVMSCSIController` command lets you add up to four SCSI Controllers to a VM. You can add up to 64 disks to a SCSI controller.

This command has the following format:

```
dpmhv-addVMSCSIController
-host hostname
-vm vmname|-vmid vmguid
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Add SCSI Controllers to a VM

This example adds a SCSI Controller to the VM, " TestVM."

```
dpmhv-addVMSCSIController -host hvserver -vm TestVM
```

dpmhv-changeVMState (Cmdlet) Command--Change the State of a VM

The `dpmhv-changeVMState` command changes the state of the VM. The various states are: start, suspend, stop, pause, shutdown, and reboot.

This command has the following format:

```
dpmhv-changeVMState
-host hostname
-vm vmname|-vmid vmguid
-state {start,suspend,stop,pause,shutdown,reboot}
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-state {start,suspend,stop,pause,shutdown,reboot}

Specifies the state of the VM. State options are the following:

start

Starts or resumes the VM.

suspend

Suspends the VM temporarily.

stop

Stops the VM.

pauses

Stops the VM temporarily.

shutdown

Shuts down the VM.

reboot

Restarts the VM.

Note:

If you specify an invalid state change, the system reports an exception.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Change the state of a VM

This example starts the VM, "TestVM."

```
dpmhv-changeVMState -host hvserver -vm TestVM -state start
```

dpmhv-createTemplateFromVM (Funclet) Command--Create a Template from a VM

The `dpmhv-createTemplateFromVM` command creates the template using an existing VM. You can use the template to create multiple VMs.

The template copies the VM specifications and virtual hard disks attached to the VM. If you create a template using the VM with snapshots, only the current state is exported without snapshots. This command creates a template catalog for each Hyper-V server. The template catalog is maintained and can be queried from the following location:
%ALLUSERSPROFILE%\ca\vpvm\Hyper-V_Templates\TemplateCatalog.xml

This command has the following format:

```
dpmhv -createTemplateFromVM
-host hostname
-vm vmname|-vmid vmguid
-template templatename
[-async]
[-path pathname]
[-description description]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-template *templatename*

Specifies the name of the generic template created in the template catalog.

Note: The template name must be unique in the local template catalog.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-dest *pathname*

(Optional) Specifies the path of the VM that you want to create (template is stored). If this option is not specified, VM is created in the Hyper-V Server default location. The name of the template created is stored in the following location:
%ALLUSERSPROFILE%\ca\vpms\Hyper-V_Templates.

Note: We recommend not creating VMs in the Hyper-V default location. It conflicts with virtual disk image file names that do not change when creating VMs using templates.

-description *description*

(Optional) Specifies the template description.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

When the `dpmhv-createTemplateFromVM` command is executed synchronously, it returns an empty string. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. If an error occurs during execution, the command reports an exception.

Example: Create a Template from a VM

This example creates the template, "MyTemplate" from the VM, "TestVM."

```
dpmhv-createTemplateFromVM -vm TestVM -template MyTemplate -host hvserver
```

dpmhv-createVirtDisk (Funclet) Command--Create a Virtual Disk

The `dpmhv-createVirtDisk` command lets you create virtual disk.

This command has the following format:

```
dpmhv-createVirtDisk
-host hostname
-path pathname
-type{fixed,dynamic}
-sizeGB size
[-async]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-path *pathname*

Specifies the path of the virtual disk that you want to create.

-type {fixed, dynamic}

Specifies the type of the virtual disk that you want to create. Options include the following:

fixed

Specifies the virtual disk type as fixed.

dynamic

Specifies the virtual disk type as dynamic.

-sizeGB *size*

Specifies the size of the virtual disk in gigabytes.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Create a Virtual Disk

This example creates the virtual disk of size 1 GB.

```
dpmhv-createVirtDisk -host hvserver -path C:\data\Disks\01.vhd -type fixed -sizeGB 1
```

dpmhv-createVirtFloppy (Cmdlet) Command--Create a Virtual Floppy

The `dpmhv-createVirtFloppy` command lets you create virtual floppy disk image file. The virtual floppy can be attached to a VM using the `dpmhv-setVMVirtFloppy` command.

This command has the following format:

```
dpmhv-createVirtFloppy  
-host hostname  
-path pathname[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-path *pathname*

Specifies the path of the virtual floppy disk image file that you want to create. If not specified, the image file is created in the default directory of Hyper-V virtual hard disk.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Create a Virtual Floppy

This example creates a virtual floppy disk image file, "fdd.vfd."

```
dpmhv-createVirtFloppy -host hvserver -path C:\VMDisks\fdd.vfd
```

dpmhv-createVM (Funclet) Command--Create a VM

The dpmhv-createVM command lets you create a VM. This command creates VM without system components such as network adapter or hard disks. Upon creating VM, you can create system components using respective commands.

This command has the following format:

```
dpmhv-createVM
-host hostname
-name name
[-path pathname]
[-template templatename]
[-templateid vmguid]
[-mem mem]
[-cpus cpus]
[-cpuidlimit limit]
[-cpureserve reserve]
[-cpulimit limit]
[-cpuweight weight]
[-startAction {none|auto|always}]
[-startDelay delay]
[-stopAction {save,off,shutdown}]
[-recoveryAction {none,restart,revert}]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-name *name*

(Optional) Specifies the name of the VM.

Note: The name of the VM must be unique.

-dest *pathname*

(Optional) Specifies the path of the VM that you want to create (template is stored). If this option is not specified, VM is created in the Hyper-V Server default location. The name of the template created is stored in the following location:
%ALLUSERSPROFILE%\ca\vpms\Hyper-V_Templates.

Note: We recommend not creating VMs in the Hyper-V default location. It conflicts with virtual disk image file names that do not change when creating VMs using templates.

-template *templatename*

(Optional) Specifies the name of the existing VM. This option is used to copy the configuration settings of existing VMs to create new VMs. We recommend not creating VMs from existing VMs. Two VMs share resources such as virtual disk and cannot be run simultaneously.

Note: The template name must be unique in the local template catalog.

-templateid *vmguid*

(Optional) Specifies the unique ID (GUID) of the existing VM. This option is used to copy the configuration settings of the existing VMs to create new VMs. We recommend not creating VMs from existing VMs. Two VMs can share resources such as virtual disks and cannot be run simultaneously.

Default: -1

Note: This option does not represent the template ID from VM catalog, but it represents the existing VM in Hyper-V environment.

-mem *mem*

(Optional) Specifies the RAM memory in megabytes (MB) for the VM that you want to create.

Default: -1

-cpus *cpus*

(Optional) Specifies the number of CPU cores to assign to the VM.

-cpuidlimit *limit*

(Optional) Specifies the limit of CPU ID functionality of VM. This option improves the compatibility with legacy operating systems such as Windows NT.

Default: -1

-cpureserve *reserve*

(Optional) Specifies the percentage of the CPU that you want to reserve for the VM. If this option is not specified, Hyper-V server assigns CPU cycles based on the overall system usage.

Default: -1

-cpulimit *limit*

(Optional) Specifies the limit for the number of CPU resources used by a VM. This option improves the compatibility while moving VMs between physical hosts with different CPU capabilities.

Default: -1

-cpuweight *weight*

(Optional) Specifies the relative weight of the virtual machine. This option controls resource allocation when more than one VM is running. Valid entry: integer, 1-10000.

Default: -1

-startAction {*none,auto,always*}

(Optional) Specifies the action that you want to perform on the VM when the Hyper-V host starts up. Options include the following:

none

Performs no action.

auto

Starts the VM automatically.

Note: Use this option if the VM was running before the Hyper-V host is shut down.

always

Starts the VM always.

-startDelay *delay*

(Optional) Specifies the delay in seconds to start the VM after the Hyper-V host is up and running.

Default: -1

-stopAction {save,off,shutdown}

(Optional) Specifies the action that you want to perform on the VM when the physical Hyper-V shuts down. Options include the following:

save

Suspends the VM.

off

Power offs the VM.

shutdown

Shuts down the system.

-recoveryAction {none,restart,revert}

(Optional) Specifies the action to perform on the VM after the Hyper-V host restarts from unexpected shutdown. Options include the following:

none

Performs no action.

restart

Restart the VM.

revert

Reverts to the last snapshot of the VM.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

This command returns a string with the unique identifier (GUID) of newly created VM.

Example: Create a VM with default settings

This example creates VM, "TestVM" with default settings.

```
dpmhv-createVM -host hvserver -name TestVM
```

Example: Create a VM with default settings in user-defined directory

This example creates VM,"TestVM" with default settings in user-defined directory.

```
dpmhv-createVM -host hvserver -name TestVM -path c:\VMs\TestVM
```

Example: Create a VM with custom memory and CPU core values

This example creates VM, "TestVM" with custom memory and CPU core values.

```
dpmhv-createVM -host hvserver -name TestVM -mem 1024 -cpus 2
```

dpmhv-createVMFromTemplate (Funclet) Command--Create a VM Using a Template

The `dpmhv-createVMFromTemplate` command creates a VM from a previously created Microsoft System Center Virtual Machine Manager (SCVMM) template and optionally customizes settings for Windows images.

The template is left untouched, so that a single template can be used to create any number of virtual machines. The command supports (SCVMM) hosted templates only. See `dpmhv-createVMFromTemplateEx` for local template support without SCVMM.

This command has the following format:

```
dpmhv -createVMFromTemplate
-host hostname
-vm vmname
-template templatename
-dest pathname
-scvmmHost scvmmHostname
-adminPass adminPW
[-computerName computerName]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-productKey key]
[-userName UserName]
[-organization orgname]
[-domain domainname]
[-domainAdmin domad]
[-domainAdminPass dompw]
[-workgroup wgname]
[-adminUser adminUser]
[-adminUserPass adminUserPW]
[-mem mem]
[-cpus cpus]
[-hardwareProf hwprof]
[-guestOSProf osprof]
[-startVM]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-template *templatename*

Specifies the name of the source template that you want to use to create the VM.

Note: The template name must be unique in the local template catalog.

-dest *pathname*

(Optional) Specifies the path of the VM that you want to create (template is stored). If this option is not specified, VM is created in the Hyper-V Server default location.

The name of the template created is stored in the following location:

%ALLUSERSPROFILE%\ca\vpms\Hyper-V_Templates.

Note: We recommend not creating VMs in the Hyper-V default location. It conflicts with virtual disk image file names that do not change when creating VMs using templates.

-scvmmHost *scvmmHostname*

(Optional) Specifies the host name of the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter is valid when you use SCVMM integration to provision VMs.

-adminPass *adminPW*

(Optional) This option is used to set the default administrator account password for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This parameter is ignored in asynchronous execution.

Note: To set this option successfully, set the Windows Server administrator password which is created using Sysprep tool as empty.

-computerName *computerName*

(Optional) Specifies the computer name of the VM. Support for this parameter requires an image using Sysprep tool. This option is invalid for asynchronous execution of the command.

-ip4addr *ip4addr*

(Optional) Specifies the static IPv4 address that you want to assign to an interface of the VM. To set an IP address for a specific interface, the IP address is prefixed with known interface name and '#' as separator. For example, -ip4addr "Local Area Connection#192.168.1.200". If the template image has more than one network adaptor, the IP address is assigned to the first interface. This option is invalid for asynchronous execution of the command.

-ip4dhcp *ip4dhcplnt*

(Optional) Specifies an option to turn on DHCP of a particular interface of the VM. You can also specify the interface name for this option. For example, -ip4dhcp "Local Area Connection." If the template image has more than one network adaptor, DHCP is turned on for the first interface. This option is invalid for asynchronous execution of the command.

Default: local

-ip4mask *ip4mask*

(Optional) Specifies the subnet mask that you want to assign for the VM. This option is used with -ip4addr option. If an interface name is specified in the -ip4addr option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of the command.

-ip4gw *ip4gw*

(Optional) Specifies the option to set the gateway for VM. This option is used with -ip4addr option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Note: If an interface name is specified in the -ip4addr option, same interface name must be used in this option.

-ip4metric *ip4metric*

(Optional) Specifies the interface metric that you want to set for the VM. This option is used with -ip4addr option. If an interface name is specified in the -ip4addr option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: 1

-ip4dns *ip4dns*

(Optional) Specifies the DNS server that you want to set for the VM. This option is used with -ip4addr option. If an interface name is specified in the -ip4addr option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-adminPass *adminPW*

(Optional) This option is used to set the default administrator account password for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This parameter is ignored in asynchronous execution.

Note: To set this option successfully, set the Windows Server 2003 administrator password which is created using Sysprep tool as empty.

-productKey *key*

(Optional) Specifies the Windows product activation key for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-userName *UserName*

(Optional) Specifies the user name of the Windows image (created using sysprep tool) which is installed on the new VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-organization *orgname*

(Optional) Specifies the organization name of the Windows image (created using sysprep tool) which is installed on the new VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-domain *domainname*

(Optional) Specifies the domain name for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command. Domain and workgroup specifications are mutually exclusive.

-domainAdmin *domad*

(Optional) Specifies the user account as Domain Administrator that you want to create in the domain. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-domainAdminPass *dompw*

(Optional) Specifies the password for Domain Administrator user account that you want to create in the domain. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-workgroup *wgname*

(Optional) Specifies the workgroup that you want to create for the VM. Support for this parameter requires a Windows image created using Sysprep tool. Domain and workgroup specifications are mutually exclusive. This option is invalid for asynchronous execution of command.

-adminUser *adminUser*

(Optional) Specifies the administrator user name that you want to create as a part of the default Administrators group. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-adminUserPass *adminUserPW*

(Optional) Specifies the password of the administrator user account that you want to create as a part of the default Administrators group. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-mem *mem*

(Optional) Specifies the RAM memory in megabytes (MB) for the VM that you want to create.

Default: -1

-cpus *cpus*

(Optional) Specifies the number of CPU cores to assign to the VM.

Default: -1

-scvmmHost *scvmmHostname*

(Optional) Specifies the host name of the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter is valid when you use SCVMM integration to provision VMs.

-hardwareProf *hwprof*

(Optional) Specifies the name of the hardware profile defined by the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter overwrites the hardware configuration settings stored in the SCVMM template. This parameter is valid when you use SCVMM integration to provision VMs.

-guestOSProf *osprof*

(Optional) Specifies the name of the guest operating system profile defined by the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter overwrites the operating system configuration settings stored in the SCVMM library server. This parameter is valid when you use SCVMM integration to provision VMs.

-startVM

(Optional) Specifies the option to start the VM automatically after it is created. By default, the new VM is in powered-off state. You must customize the IP configuration options required to start the VM.

Return Value

The `dpmhv-createVMFromTemplate` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. If an error occurs during execution, the command reports an exception.

Example: Create a VM from an SCVMM Template

This example lets you create VM from an SCVMM template with options to specify the hardware and guest OS profiles:

```
dpmhv-createVMFromTemplate -host hvserver -vm NewVM -dest c:\VMs\NewBox -template Win2k3Clean -scvmmHost SCVMMLibSrv -hardwareProf hw_def -guestOSProf os_def -async -adminPass #admin#
```

See also:

[dpmhv-createVMFromTemplateEx \(Funclet\) Command--Create a VM Using a Template](#)
(see page 217)

dpmhv-createVMFromTemplateEx (Funclet) Command--Create a VM Using a Template

The dpmhv-createVMFromTemplate command lets you create VM using existing templates. If the original template is created in Sysprep state, you can customize the (Windows) settings of the VM created (image) using the command parameters. Upon the creation of the VM, you can create system components using respective commands. You can use one template to create multiple virtual machines.

Note: You can customize the Windows Settings image when the command is run synchronously. For asynchronous execution, you can use dpmhv-setSysprepProperties command to customize the image after the execution is complete.

This command has the following format:

```
dpmhv -createVMFromTemplateEx
-host hostname
-vm vmname
-template templatename
-dest pathname
[-computerName computerName]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-disableAdmin]
[-adminPass adminPW]
[-autoLogon alcnt]
[-duplicatorString dupstr]
[-timeZone timezone]
[-productKey key]
[-userName UserName]
[-organization orgname]
[-domain domainname]
[-domainAdmin domad]
[-domainAdminPass dompw]
[-workgroup wgname]
[-adminUser adminUser]
[-adminUserPass adminUserPW]
[-custom custom]
[-async]
[-mem mem]
[-cpus cpus]
[-cpuidlimit limit]
[-cpufeatlimit limit]
[-cpureserve reserve]
[-cpulimit limit]
[-cpuweight weight]
[-startAction {none,auto,always}]
[-startDelay delay]
[-stopAction {save,off,shutdown}]
[-recoveryAction {none,restart,revert}]
[-scvmmHost scvmmHostname]
[-hardwareProf hwprof]
[-guestOSProf osprof]
[-startVM]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-template *templatename*

Specifies the name of the source template that you want to use to create the VM.

Note: The template name must be unique in the local template catalog.

-dest *pathname*

(Optional) Specifies the path of the VM that you want to create (template is stored). If this option is not specified, VM is created in the Hyper-V Server default location. The name of the template created is stored in the following location:
%ALLUSERSPROFILE%\ca\vpms\Hyper-V_Templates.

Note: We recommend not creating VMs in the Hyper-V default location. It conflicts with virtual disk image file names that do not change when creating VMs using templates.

-computerName *computerName*

(Optional) Specifies the computer name of the VM. Support for this parameter requires an image using Sysprep tool. This option is invalid for asynchronous execution of the command.

-ip4addr *ip4addr*

(Optional) Specifies the static IPv4 address that you want to assign to an interface of the VM. To set an IP address for a specific interface, the IP address is prefixed with known interface name and '#' as separator. For example, -ip4addr "Local Area Connection#192.168.1.200". If the template image has more than one network adaptor, the IP address is assigned to the first interface. This option is invalid for asynchronous execution of the command.

-ip4dhcp *ip4dhcplnt*

(Optional) Specifies an option to turn on DHCP of a particular interface of the VM. You can also specify the interface name for this option. For example, -ip4dhcp "Local Area Connection." If the template image has more than one network adaptor, DHCP is turned on for the first interface. This option is invalid for asynchronous execution of the command.

Default: local

-ip4mask *ip4mask*

(Optional) Specifies the subnet mask that you want to assign for the VM. This option is used with -ip4addr option. If an interface name is specified in the -ip4addr option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of the command.

-ip4gw *ip4gw*

(Optional) Specifies the option to set the gateway for VM. This option is used with *-ip4addr* option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Note: If an interface name is specified in the *-ip4addr* option, same interface name must be used in this option.

-ip4metric *ip4metric*

(Optional) Specifies the interface metric that you want to set for the VM. This option is used with *-ip4addr* option. If an interface name is specified in the *-ip4addr* option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: 1

-ip4dns *ip4dns*

(Optional) Specifies the DNS server that you want to set for the VM. This option is used with *-ip4addr* option. If an interface name is specified in the *-ip4addr* option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-disableAdmin

(Optional) Specifies an option to disable default administrator account for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-adminPass *adminPW*

(Optional) This option is used to set the default administrator account password for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This parameter is ignored in asynchronous execution.

Note: To set this option successfully, set the Windows Server administrator password which is created using Sysprep tool as empty.

-autoLogon *alcnt*

(Optional) Specifies an option to set the number of accounts that you want to create to log in automatically with default administrator account. The accounts are created after the completion of Sysprep process. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default:

-1

-duplicatorString *dupstr*

(Optional) Specifies the name of the system duplicator that you want to set in the registry of the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-timeZone *timezone*

(Optional) Specifies the time zone used by the VM that are created using the template. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default:

-1

-productKey *key*

(Optional) Specifies the Windows product activation key for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-userName *UserName*

(Optional) Specifies the user name of the Windows image (created using sysprep tool) which is installed on the new VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-organization *orgname*

(Optional) Specifies the organization name of the Windows image (created using sysprep tool) which is installed on the new VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-domain *domainname*

(Optional) Specifies the domain name for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command. Domain and workgroup specifications are mutually exclusive.

-domainAdmin *domad*

(Optional) Specifies the user account as Domain Administrator that you want to create in the domain. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-domainAdminPass *dompw*

(Optional) Specifies the password for Domain Administrator user account that you want to create in the domain. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-workgroup *wgname*

(Optional) Specifies the workgroup that you want to create for the VM. Support for this parameter requires a Windows image created using Sysprep tool. Domain and workgroup specifications are mutually exclusive. This option is invalid for asynchronous execution of command.

-adminUser *adminUser*

(Optional) Specifies the administrator user name that you want to create as a part of the default Administrators group. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-adminUserPass *adminUserPW*

(Optional) Specifies the password of the administrator user account that you want to create as a part of the default Administrators group. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-custom *custom*

(Optional) Specifies the list of comma-separated custom commands that you want to execute at the end of Sysprep process. This parameter requires Windows image of Sysprep. This parameter is ignored in asynchronous execution.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-mem *mem*

(Optional) Specifies the RAM memory in megabytes (MB) for the VM that you want to create.

Default: -1

-cpus *cpus*

(Optional) Specifies the number of CPU cores to assign to the VM.

Default: -1

-cpuidlimit *limit*

(Optional) Specifies the limit of CPU ID functionality of VM. This option improves the compatibility with legacy operating systems such as Windows NT.

Default: -1

-cpufeatlimit *limit*

(Optional) Specifies the limit for the functionality of the CPU resources used by a VM. This option improves the compatibility while moving VMs between physical hosts with different CPU capabilities.

Default: -1

-cpureserve *reserve*

(Optional) Specifies the percentage of the CPU that you want to reserve for the VM. If this option is not specified, Hyper-V server assigns CPU cycles based on the overall system usage.

Default: -1

-cpulimit *limit*

(Optional) Specifies the limit for the number of CPU resources used by a VM. This option improves the compatibility while moving VMs between physical hosts with different CPU capabilities.

Default: -1

-cpuweight *weight*

(Optional) Specifies the relative weight of the virtual machine. This option controls resource allocation when more than one VM is running. Valid entry: integer, 1-10000.

Default: -1

-startAction {*none,auto,always*}

(Optional) Specifies the action that you want to perform on the VM when the Hyper-V host starts up. Options include the following:

none

Performs no action.

auto

Starts the VM automatically.

Note: Use this option if the VM was running before the Hyper-V host is shut down.

always

Starts the VM always.

-startDelay *delay*

(Optional) Specifies the delay in seconds to start the VM after the Hyper-V host is up and running.

Default: -1

-stopAction {save,off,shutdown}

(Optional) Specifies the action that you want to perform on the VM when the physical Hyper-V shuts down. Options include the following:

save

Suspends the VM.

off

Power offs the VM.

shutdown

Shuts down the system.

-recoveryAction {none,restart,revert}

(Optional) Specifies the action to perform on the VM after the Hyper-V host restarts from unexpected shutdown. Options include the following:

none

Performs no action.

restart

Restart the VM.

revert

Reverts to the last snapshot of the VM.

-scvmmHost *scvmmHostname*

(Optional) Specifies the host name of the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter is valid when you use SCVMM integration to provision VMs.

-hardwareProf *hwprof*

(Optional) Specifies the name of the hardware profile defined by the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter overwrites the hardware configuration settings stored in the SCVMM template. This parameter is valid when you use SCVMM integration to provision VMs.

-guestOSProf *osprof*

(Optional) Specifies the name of the guest operating system profile defined by the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter overwrites the operating system configuration settings stored in the SCVMM library server. This parameter is valid when you use SCVMM integration to provision VMs.

-startVM

(Optional) Specifies the option to start the VM automatically after it is created. By default, the new VM is in powered-off state. You must customize the IP configuration options required to start the VM.

Return Value

The `dpmhv-createVMFromTemplateEX` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. If an error occurs during execution, the command reports an exception.

Example: Create a VM from a Template

This example lets you create VM, "TestVM" using a template setting as complete static IPv4 configuration:

```
dpmhv-createVMFromTemplateEx -host hvserver -vm NewVM -dest c:\VMs\NewBox -template Win2k3SysPrepped -ip4addr 192.168.1.25 -ip4mask 255.255.255.0 -ip4gw 192.168.1.1 -ip4dns 192.168.1.1 -computerName NewBox
```

See also:

[dpmhv-createVMFromTemplate \(Funclet\) Command--Create a VM Using a Template](#) (see page 211)

dpmhv-destroyTemplate (Cmdlet) Command--Delete a Template

The `dpmhv-destroyTemplate` command lets you delete existing templates from the template catalog.

This command has the following format:

```
dpmhv-destroyTemplate  
-template templatename  
-host hostname
```

-template *templatename*

Specifies the name of the generic template created in the template catalog.

Note: The template name must be unique in the local template catalog.

-host *hostname*

Specifies the name of the Hyper-V server host.

Example: Delete a Template

This example deletes a Template from the template catalog.

```
dpmhv-destroyTemplate -host hvserver -template MyTemplate
```

dpmhv-destroyVM (Cmdlet) Command--Delete VM

The `dpmhv-destroyVM` command lets you delete existing VM specifications, virtual disks, floppy images, and its related snapshots.

This command has the following format:

```
dpmhv-destroyVM
-host hostname
-vm vmname|-vmid vmguid
[-delvhd]
[-delvfd]
[-deliso]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-delvhd

(Optional) Deletes the virtual hard disk assigned to the VM.

-delvfd

(Optional) Deletes the virtual floppy disk assigned to the VM.

-deliso

(Optional) Deletes the ISO image assigned to the VM.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Delete a VM

This example deletes VM with the virtual hard disk and floppy disk attached to the VM.

```
dpmhv-destroyVM -host hvserver -vm TestVM -delvhd -delvfd
```

dpmhv-expandVirtDisk (Funclet) Command--Expand Virtual Disk

The `dpmhv-expandVirtDisk` command lets you increase the virtual disk size. To use the expanded virtual disk space, you must partition the virtual disk in the guest operating system running on a virtual machine.

This command has the following format:

```
dpmhv-expandVirtDisk
-host hostname
-path pathname
-sizeGB size
[-async]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-path *pathname*

Specifies the path name of the virtual disk that you want to expand.

-sizeGB *size*

Specifies the size of the virtual disk in gigabytes.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-expandVirtDisk` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. In synchronous mode, if an error occurs during execution, the command returns the error code and reports the error.

Example: Expand Virtual Disk

This example expands the size of the virtual disk to 4 GB.

```
dpmhv-expandVirtDisk -host hvserver -path C:\VMDisks\hdd.vhd -sizeGB 4
```

dpmhv-exportVM (Funclet) Command--Export a VM

The `dpmhv-exportVM` command lets you export VM specifications, including their related virtual disks and floppy disks. This command is deprecated, use `dpmhv-exportVMEx` instead.

This command has the following format:

```
dpmhv-exportVM  
-host hostname  
-vm vmname|-vmid vmguid  
-dest dest  
[-nostate]  
[-async]  
[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-dest *dest*

Specifies the destination path of the VM that you want to export.

-nostate

(Optional) Specifies that only the VM specification is exported. This option prevents the exported VM from referring to the original VM disk images.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-exportVM` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. In synchronous mode, if an error occurs during execution, the command returns the error code and reports the error.

Example: Export a VM

This example you export VM specifications, including its related virtual disks and floppy disks.

```
dpmhv-exportVM -host hvserver -vm VM_0000001 -dest c:\DATA\Export\
```

dpmhv-exportVMEx (Funclet) Command--Export a VM

The `dpmhv-exportVMEx` command lets you export VM specifications, including its related virtual disks, floppy disks and snapshots.

This command has the following format:

```
dpmhv-exportVMEx  
-host hostname  
-vm vmname|-vmid vmguid  
-dest dest  
[-noruntimeInfo]  
[-nosnapshots]  
[-nostorage]  
[-nosubdir]  
[-snapshot ss]  
[-snapshotid ssguid]  
[-async]  
[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-dest *dest*

Specifies the destination path of the VM that you want to export.

-noruntimeInfo

(Optional) Specifies not to export runtime information of the VM.

-nosnapshots

(Optional) Specifies not to export snapshots of the VM. This option exports only the current state of the VM.

-nostorage

(Optional) Specifies not to export the virtual disk of the VM.

Note: Do not use this option when you export the snapshot of the VM.

-nosubdir

(Optional) Specifies not to create a sub directory with the VM display name underneath the directory specified as the export path.

-snapshot ss

(Optional) Specifies the display name of the snapshot of the VM. The exported VM is in the state (snapshot) similar to the source VM. The exported VM does not include snapshot.

Note: Use

-ssid option as snapshot display name is not unique in the Hyper-V environment.

-snapshotid ssguid

(Optional) Specifies the unique ID (GUID) of the snapshot of the VM. The exported VM is in the state (snapshot) similar to the source VM. The exported VM does not include snapshot.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-exportVMEx` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. In synchronous mode, if an error occurs during execution, the command returns the error code and reports the error.

Example: Export a VM Without Snapshots

This example exports a VM without snapshots.

```
dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:\Ex\TestVM1 -nosubdir -nosnapshots
```

dpmhv-getHostSwitches (Funclet) Command--Show Host Switches on a Hyper-V Host

The `dpmhv-getHostSwitches` command displays host switches (virtual networks) defined on a Hyper-V host.

This command has the following format:

```
dpmhv-getHostSwitches  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the virtual networks (host networks).

Return Value

The `dpmhv-getHostSwitches` command returns an array of objects with each object representing one virtual switch (virtual host) on the specified host.

The following properties are returned in regular mode:

- BytesReceivedPersec
- BytesSentPersec
- Caption
- ElementName
- Name
- StatusDescriptions

The following properties are returned in detail mode:

- BroadcastPacketsReceivedPersec
- BroadcastPacketsSentPersec
- BytesPersec
- BytesReceivedPersec
- BytesSentPersec
- Caption
- CreationClassName
- Dedicated
- Description
- DirectedPacketsReceivedPersec
- DirectedPacketsSentPersec
- ElementName
- EnabledDefault
- EnabledState
- HealthState
- IdentifyingDescriptions
- InstallDate
- LearnedMacAddresses
- LearnedMacAddressesPersec
- MaxChimneyOffloads
- MaxVMQOffloads

- MulticastPacketsReceivedPersec
- MulticastPacketsSentPersec
- Name
- Name1
- NameFormat
- NumLearnableAddresses
- OperationalStatus
- OtherDedicatedDescriptions
- OtherEnabledState
- OtherIdentifyingInfo
- PacketsFlooded
- PacketsFloodedPersec
- PacketsPersec
- PacketsReceivedPersec
- PacketsSentPersec
- Path
- PowerManagementCapabilities
- PrimaryOwnerContact
- PrimaryOwnerName
- PurgedMacAddresses
- PurgedMacAddressesPersec
- RequestedState
- ResetCapability
- Roles
- ScopeOfResidence
- Status
- StatusDescriptions
- TimeOfLastStateChange

Example: Show Host Switches

This example displays the list of host switches on a Hyper-V host.
`dpmhv-getHostSwitches -host hvserver`

dpmhv-getJobInfo (Funclet) Command--Show Job Information

The dpmhv-getJobInfo command retrieves all the information pertaining to an asynchronous job such as completion percentage, error information, and start time.

This command has the following format:

```
dpmhv-getJobInfo  
-jobref job  
[-silent]  
[-retval]  
[-detail]
```

-jobref *job*

Specifies the job ID of an asynchronous job.

Note: Upon the complete execution of asynchronous job, Hyper-V holds the job information for few minutes (five) only, beyond which it becomes invalid.

-silent

Specifies not to direct the output to the screen.

-retval

Returns a value for further processing.

-detail

(Optional) Displays the information details of the asynchronous job.

Return Value

The `dpmhv-getJobInfo` command returns objects where each object represents the current job state.

Hyper-V Jobs:

The following properties are returned for local Hyper-V in regular mode:

- Caption
- ElementName
- ErrorCode
- ErrorDescription
- JobStatus
- PercentComplete
- ResultVMName
- Status
- StatusDescriptions

The following properties are returned for local Hyper-V in detail mode:

- Cancellable
- Caption
- DeleteOnCompletion
- Description
- ElapsedTime
- ElementName
- ErrorCode
- ErrorDescription
- ErrorSummaryDescription
- HealthState
- InstallDate
- InstanceID
- JobRunTimes
- JobState
- JobStatus
- LocalOrUtcTime
- Name

- Notify
- OII- OperationalStatus
- OtherRecoveryAction
- Owner
- Path
- PercentComplete
- Priority
- RecoveryAction
- ResultVMName
- RunDay
- RunDayOfWeek
- RunMonth
- RunStartInterval
- ScheduledStartTime
- StartTime
- Status
- StatusDescriptions
- TimeBeforeRemoval
- TimeOfLastStateChange
- TimeSubmitted
- UntilTime

Microsoft System Center Virtual Machine Manager (SCVMM) Jobs:

Properties returned for SCVMM in regular mode:

- CurrentStep
- Description
- ErrorInfo
- IsCompleted
- OII
- ProgressValue
- ResultVMName
- Status
- StatusString

Properties returned for SCVMM in detail mode:

- AdditionalMessages[]
- AreAuditRecordsAvailable
- AuditRecords[]
- CmdletName
- CurrentStep
- Description
- EndTime
- ErrorInfo
- ID
- IsCompleted
- IsFullyCached
- IsRestartable
- IsStoppable
- IsVisible
- MarkedForDeletion
- Name
- OII
- Owner
- OwnerSID
- PROTipID
- Progress
- ProgressValue
- ResultName
- ResultObjectID
- ResultObjectType
- ResultObjectTypeName
- ResultVMName
- ServerConnection
- Source
- StartTime
- Status

- StatusString
- Steps[]
- Target
- TargetObjectID
- TargetObjectType
- WasNotifiedOfCancel

Example: Show Job Information

This example displays status of completion of an export job.

```
jref = dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:\Ex-async
do
{
    sleep(5000)
    job = dpmhv-getJobInfo -jobref (jref) -silent -retval
    ? "Completion:", job.PercentComplete+"%"
} while(job.ErrorCode==0 && job.PercentComplete<100)
? job.StatusDescriptions
if(job.ErrorCode!=0)
{
    ? job.ErrorDescription
}
```

Example: Show SCVMM Job Information

This example displays the status of the completion of new VM created using an SCVMM template.

```
jref=dpmhv-createVMFromTemplate -host hvserver -vm NewVM -dest c:\VMs\NewBox
-template Win2k3Clean -scvmmHost SCVMMLibSrv -async
do
{
    sleep(5000)
    job = dpmhv-getJobInfo -jobref (jref) -silent -retval
    ? "Completion:", job.ProgressValue+"%"
} while(!job.IsCompleted)
? job.StatusString
s = job.ErrorInfo;
errorCode = parseInt(s.substr(s.lastIndexOf('(')+1).left(-1));
if(errorCode!=0)
{
    ? job.ErrorInfo
}
```

dpmhv-getJobStatus (Funclet) Command--Show Job Status

The dpmhv-getJobStatus command retrieves the status of an asynchronous job.

This command has the following format:

```
dpmhv-getJobStatus  
-jobref job
```

-jobref *job*

Specifies the job ID of an asynchronous job.

Note: Upon the complete execution of asynchronous job, Hyper-V holds the job information for few minutes (five) only, beyond which it becomes invalid.

Return Value

The dpmhv-getJobStatus command returns the following status information of the specified job:

- Canceled
- Created
- Failed
- Progressing
- Succeeded
- Unknown
- Waiting

Example: Show Job Status

This example displays the status of an export job.

```
jref = dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:\Ex -async  
dpmhv-getJobState -jobref (jref)
```

dpmhv-getPhysDisks (Funclet) Command--Show all Physical Disks

The dpmhv-getPhysDisks command lists the physical disks to attach to virtual machines on a Hyper-V server. The physical disks should be displayed in offline status in Windows Disk Manager.

This command has the following format:

```
dpmhv-getPhysDisks  
-host hostname  
[-silent]  
[-retval]  
[-detail]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the physical disks attached to a virtual machine.

Return Value

The `dpmhv-getPhysDisks` command returns an array of objects representing the SCSI controllers of the specified virtual machine.

The properties returned in regular mode are:

- Caption
- DriveNumber
- ElementName
- Name
- Path
- StatusDescriptions

The properties returned in detail mode are:

- AdditionalAvailability
- Availability
- Capabilities
- CapabilityDescriptions
- Caption
- CompressionMethod
- CreationClassName
- DefaultBlockSize
- Description
- DeviceID
- DriveNumber
- ElementName
- EnabledDefault
- EnabledState
- ErrorCleared
- ErrorDescription
- ErrorMethodology
- HealthState
- IdentifyingDescriptions
- InstallDate
- LastCleaned

- LastErrorCode
- LoadTime
- LocationIndicator
- MaxAccessTime
- MaxBlockSize
- MaxMediaSize
- MaxQuiesceTime
- MaxUnitsBeforeCleaning
- MedialsLocked
- MinBlockSize
- MountCount
- Name
- NeedsCleaning
- NumberOfMediaSupported
- OperationalStatus
- OtherEnabledState
- OtherIdentifyingInfo
- Path
- PowerManagementCapabilities
- PowerManagementSupported
- PowerOnHours
- RequestedState
- Security
- Status
- StatusDescriptions
- StatusInfo
- SystemCreationClassName
- SystemName
- TimeOfLastMount
- TimeOfLastStateChange
- TotalMountTime
- TotalPowerOnHours
- UncompressedDataRate

- UnitsDescription
- UnitsUsed
- UnloadTime

Example: Show all Physical Disks

This example displays the basic information about the physical disk attached to a virtual machine.

```
dpmhv-getPhysDisks -host hvserver
```

dpmhv-getVersion (Funclet) Command--Show the Web Service Version

The `dpmhv-getVersion` command retrieves the version of the Web Service of the CA Server Automation.

This command has the following format:

```
dpmhv-getVersion  
[-retval]  
[-silent]
```

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-getVersion` command returns a string indicating the web service version of the CA Server Automation.

Example: Show the Web Service Version

This example displays the version of the Web Service of the CA Server Automation.

```
ver = dpmhv-getVersion -retval -silent  
? ver
```

dpmhv-getVMInfo (Funclet) Command--Show all VMs

The dpmhv-getVMInfo command retrieves the information and settings of a virtual machine.

This command has the following format:

```
dpmhv-getVMInfo  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-getVMInfo` command returns objects with information of virtual machine.

The following properties are returned in regular mode:

- `vm`
- `Caption`
- `ElementName`
- `Name`
- `OperationalStatus`
- `StatusDescriptions`
- `vminfo`
- `MemoryUsage`
- `ProcessorLoad`
- `ProcessorLoadHistory`
- `vmsettings`
- `AutoActivate`
- `AutomaticRecoveryAction`
- `AutomaticShutdownAction`
- `AutomaticStartupAction`
- `AutomaticStartupActionDelay`
- `Description`
- `ElementName`
- `ExternalDataRoot`
- `SnapshotDataRoot`
- `SystemName`
- `vmcursettings`
- `BaseBoardSerialNumber`
- `BIOSGUID`
- `BIOSSerialNumber`
- `ChassisAssetTag`
- `ChassisSerialNumber`
- `SystemName`

The following properties are returned in detail mode:

- vm
- AssignedNumaNodeList
- Caption
- CreationClassName
- Dedicated
- Description
- ElementName
- HealthState
- IdentifyingDescriptions
- InstallDate
- Name
- NameFormat
- OnTimeInMilliseconds
- OperationalStatus
- OtherDedicatedDescriptions
- OtherEnabledState
- OtherIdentifyingInfo
- PowerManagementCapabilities
- PrimaryOwnerContact
- PrimaryOwnerName
- ProcessID
- RequestedState
- ResetCapability
- Roles
- Status
- StatusDescriptions
- TimeOfLastConfigurationChange
- TimeOfLastStateChange
- vminfo
- AsynchronousTasks
- CreationTime
- ElementName
- EnabledState

- GuestOperatingSystem
- HealthState
- Heartbeat
- MemoryUsage
- Name
- Notes
- NumberOfProcessors
- OperationalStatus
- Path
- ProcessorLoad
- ProcessorLoadHistory
- Snapshots
- StatusDescriptions
- ThumbnailImage
- UpTime
- vmsettings
- AdditionalRecoveryInformation
- AllowFullSCSICommandSet
- AutoActivate
- AutomaticRecoveryAction
- AutomaticShutdownAction
- AutomaticStartupAction
- AutomaticStartupActionDelay
- Caption
- CreationTime
- DebugChannelId
- Description
- ElementName
- ExternalDataRoot
- InstanceID
- OtherVirtualSystemType
- Path
- ScopeOfResidence

- SettingType
- SnapshotDataRoot
- SystemName
- Version
- VirtualSystemType
- vmcursettings
- BIOSGUID
- BIOSSerialNumber
- BaseBoardSerialNumber
- ChassisAssetTag
- ChassisSerialNumber
- SystemName

Example: Show all VMs

This example displays the information of a virtual machine.

```
dpmhv-getVMInfo -host hvserver -vm TestVM
```

dpmhv-getVMProperties (Funclet) Command--Show VM Properties

The dpmhv-getVMProperties command retrieves the properties of the VM such as CPU, memory, and actions settings.

This command has the following format:

```
dpmhv-getVMProperties  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]
```

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-getVMProperties` command returns objects where each object represents the VM properties as follows:

- memory
- cpu
- cpuSocketCount
- cpuidLimit
- cpuReserve
- cpuLimit
- cpuWeight
- actions
- startAction
- startDelay
- stopAction
- recoveryAction

Example: Show VM Properties

This example displays the following VM properties: VM CPU and memory allocation.

```
dpmhv-getVMProperties -host hvserver -vm TestVM
```

Example: Show CPU and Memory Allocation of VM

This example displays the CPU and the memory allocation of the VM, "TestVM."

```
props=dpmhv-getVMProperties -host hvserver -vm TestVM -retval -silent
? "Memory:", props.memory+"MB"
? "CPU cores:", props.cpu.cpuSocketCount
```

dpmhv-getVMState (Funclet) Command--Show the VM State

The `dpmhv-getVMState` command retrieves the power state of the VM.

This command has the following format:

```
dpmhv-getVMState
-vm vmname|-vmid vmguid
-host hostname
[-retval]
[-silent]
```

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-host *hostname*

Specifies the name of the Hyper-V server host.

-silent

Specifies not to direct the output to the screen.

-retval

Returns a value for further processing.

Return Value

The `dpmhv-getVMState` command returns a string indicating the power state of the VM. The values returned are:

- disabled: VM is turned off
- enabled: VM is running
- paused: VM is paused
- suspended: VM is suspended

Example: Show the VM State

This example checks whether the VM is running.

```
s = dpmhv-getVMState -host hvserver -vm TestVM -retval -silent
if(s=="enabled")
? "VM is running"
else
? "VM is not running"
```

dpmhv-importVM (Funclet) Command--Import a VM

The `dpmhv-importVM` command imports an exported VM, which was exported using `dpmhv-exportVM` command. You can import VM for only one time as it is imported to a location similar to the exported VM.

This command has the following format:

```
dpmhv-importVM
-host hostname
-path pathname
[-vm vmname]
[-nonewid]
[-async]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-path *pathname*

Specifies the path of the VM that you want to import.

-vm *vmname*

Specifies the name of the VM.

-nonewid

(Optional) Specifies not to generate a unique GUID for the VM that you want to import. The imported VM has the GUID similar to the source VM (exported VM). Two VMs with same GUID creates conflicts and is not recommended.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-importVM` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. In synchronous mode, if an error occurs during execution, the command returns the error code and reports the error.

Example: Import a VM

This example imports a VM which is exported using `dpmhv-exportVM` command.

```
dpmhv-exportVM -host hvserver -vm TestVM -dest c:\Ex\MyVM dpmhv-importVM  
-host hvserver -path c:\Ex\MyVM -vm TestVM1
```

Additionally, you can issue `dpmhv-importVM` command to create multiple VMs from the originally exported VM.

dpmhv-importVMEx (Funclet) Command--Copy and Import VM

The `dpmhv-importVMEx` command lets you import and copy an exported VM, which was exported using `dpmhv-exportVMEx` command. You can import VM for multiple times as it is imported to a location different to the exported VM.

This command has the following format:

```
dpmhv-importVMEx  
-host hostname  
-path pathname  
[-dest dest]  
[-vm vmname]  
[-nonewid]  
[-async]  
[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-path *pathname*

Specifies the path of the VM that you want to import and copy.

-dest *dest*

Specifies the destination path to import and copy the VM to.

-vm *vmname*

Specifies the name of the VM.

-nonewid

(Optional) Specifies not to generate a unique GUID for the VM that you want to import. The imported VM has the GUID similar to the source VM (exported VM). Two VMs with same GUID creates conflicts and is not recommended.

-async

(Optional) Specifies the option to execute the command asynchronously. By default, this command is executed synchronously.

Note: If this command completes the execution before executing this option, it returns a job ID that is used to view the operation status.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-importVMEx` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. In synchronous mode, if an error occurs during execution, the command returns the error code and reports the error.

Example: Import VM

This example imports a VM which is exported using `dpmhv-exportVMEx` command.

```
dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:\Ex\MyVM -nosubdir  
dpmhv-importVMEx -host hvserver -path c:\Ex\MyVM -dest C:\VMs\TestVM1 -vm TestVM1
```

Additionally, you can issue `dpmhv-importVMEx` command to create multiple VMs from the originally exported VM.

dpmhv-setSysprepProperties (Cmdlet) Command--Create a Virtual Disk

The `dpmhv-setSysprepProperties` command lets you create a virtual disk image.

This command has the following format:

```
dpmhv-setSysprepProperties
-host hostname
-vm vmname|-vmid vmguid|-jobref job
[-computerName computerName]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-disableAdmin]
[-adminPass adminPW]
[-autoLogon alcnt]
[-duplicatorString dupstr]
[-timeZone timezone]
[-productKey key]
[-userName UserName]
[-organization orgname>]
[-domain domname]
[-domainAdmin domad]
[-domainAdminPass dompw]
[-workgroup wgname]
[-adminUser adminUser]
[-adminUserPass adminUserPW]
[-custom custom]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

Default: \$\$HVHost

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-jobref *job*

Specifies the job ID of an asynchronous job.

Note: Upon the complete execution of asynchronous job, Hyper-V holds the job information for few minutes (five) only, beyond which it becomes invalid.

-computerName *computerName*

(Optional) Specifies the name of the computer. Support for this parameter requires an image using Sysprep tool. This option is invalid for asynchronous execution of the command.

Default: Null

-ip4addr *ip4addr*

(Optional) Specifies the static IPv4 address that you want to assign to an interface of the VM. To set an IP address for a specific interface, the IP address is prefixed with known interface name and '#' as separator. For example, -ip4addr "Local Area Connection#192.168.1.200". If the template image has more than one network adaptor, the IP address is assigned to the first interface. This option is invalid for asynchronous execution of the command.

Default: ""

-ip4dhcp *ip4dhcpInt*

(Optional) Specifies an option to turn on DHCP of a particular interface of the VM. You can also specify the interface name for this option. For example, -ip4dhcp "Local Area Connection." If the template image has more than one network adaptor, DHCP is turned on for the first interface. This option is invalid for asynchronous execution of the command.

Default: local

-ip4mask *ip4mask*

(Optional) Specifies the subnet mask that you want to assign for the VM. This option is used with -ip4addr option. If an interface name is specified in the -ip4addr option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of the command.

Default: ""

-ip4gw *ip4gw*

(Optional) Specifies the option to set the gateway for VM. This option is used with -ip4addr option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Note: If an interface name is specified in the -ip4addr option, same interface name must be used in this option.

Default: ""

-ip4metric *ip4metric*

(Optional) Specifies the interface metric that you want to set for the VM. This option is used with *-ip4addr* option. If an interface name is specified in the *-ip4addr* option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: 1

-ip4dns *ip4dns*

(Optional) Specifies the DNS server that you want to set for the VM. This option is used with *-ip4addr* option. If an interface name is specified in the *-ip4addr* option, same interface name must be used in this option. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: ""

-disableAdmin

(Optional) Specifies an option to disable default administrator account for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

-adminPass *adminPW*

(Optional) This option is used to set the default administrator account password for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This parameter is ignored in asynchronous execution.

Note: To set this option successfully, set the Windows Server 2003 administrator password which is created using Sysprep tool as empty.

Default: Null

-autoLogon *alcnt*

(Optional) Specifies an option to set the number of accounts that you want to create to log in automatically with default administrator account. The accounts are created after the completion of Sysprep process. *hyperv_sp_dupstr*. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default:

-1

-duplicatorString *dupstr*

(Optional) Specifies the name of the system duplicator that you want to set in the registry of the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null

-timeZone *timezone*

(Optional) Specifies the time zone used by the VM that are created using the template. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default:

-1

-productKey *key*

(Optional) Specifies the Windows product activation key for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null**-userName *UserName***

(Optional) Specifies the user name of the Windows inside the VM is licensed to. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null**-organization *orgname***

(Optional) Specifies the organization the Windows copy inside the VM is licensed to. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null**-domain *domainname***

(Optional) Specifies the domain name for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command. Domain and workgroup specifications are mutually exclusive.

Default: Null**-domainAdmin *domad***

(Optional) Specifies the user account as Domain Administrator that you want to create in the domain. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null

-domainAdminPass *dompw*

(Optional) Specifies the password for Domain Administrator user account that you want to create in the domain. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null

-workgroup *wgname*

(Optional) Specifies the workgroup that you want to create for the VM. Support for this parameter requires a Windows image created using Sysprep tool. Domain and workgroup specifications are mutually exclusive. This option is invalid for asynchronous execution of command.

Default: Null

-adminUser *adminUser*

(Optional) Specifies the administrator user name that you want to create as a part of the default Administrators group. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null

-adminUserPass *adminUserPW*

(Optional) Specifies the password of the administrator user account that you want to create as a part of the default Administrators group. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid for asynchronous execution of command.

Default: Null

-custom *custom*

(Optional) Specifies the list of comma-separated custom commands that you want to execute at the end of Sysprep process. This parameter requires Windows image of Sysprep. This parameter is ignored in asynchronous execution.

Default: Null

Return Value

The `dpmhv-setSysprepProperties` command returns an empty string when it is executed synchronously. If executed asynchronously, a string representing job ID is passed to `dpmhv-getJobErrorInfo`, `dpmhv-getJobInfo`, or `dpmhv-getJobStatus`. If an error occurs during execution, an exception is reported.

Example: Create a Virtual Disk

This example creates a dynamic virtual 2GB disk.

```
dpmhv-createVirtDisk -host hvserver -path C:\VMDisks\hdd.vhd -type dynamic -sizeGB 2
```

dpmhv-setVMClustered (Cmdlet) Command--Set a VM in a Cluster

The `dpmhv-setVMClustered` command lets you add or remove a VM from the shared cluster resources. In Hyper-V server environment, the VMs in a cluster can either be a shared cluster resource or private resource of the Hyper-V server. If the VM is a cluster shared resource, you cannot enable it for Quick or Live Migration between cluster nodes. Also, you cannot migrate the VM resources located on private storage such as virtual disk on a local hard drive, instead of cluster shared storage.

This command has the following format:

```
dpmhv-setVMClustered
-host hostname
-vm vmname|-vmid vmguid
-clustered {on,off}[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-clustered {on, off}

Specifies the sharing mode of the cluster.

on

VM is a shared cluster resource.

off

VM is a private resource of the Hyper-V server.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Set a VM in a Cluster

This example adds a VM to the shared cluster resources.

```
dpmhv-setVMClustered -host hvserver -vm TestVM -clustered on
```

dpmhv-setVMDisk (Funclet) Command--Create a Drive or Disk on a Drive Controller

The dpmhv-setVMDisk command lets you create a drive or disk on a drive controller. You can use the option to connect/disconnect an image.

This command has the following format:

```
dpmhv-setVMDisk
-host hostname
-vm vmname| -vmid vmguid
[-path pathname]
-drive{HDD,DVD}
-type{scsi,ide}[-id id]
[-lun lun]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-path *pathname*

(Optional) Specifies the path where the drive or disk is created on a drive controller.

-drive {*HDD, DVD*}

Specifies the drive type that you want to assign to the VM. You can assign the following drive types:

HDD

Represents the hard drive

DVD

Represents the drive of either Compact Disk (CD) or Digital Video Disk (DVD).

-type {scsi, ide}

Specifies the controller type that you want to attach to the virtual hard disk of the VM. Hyper-V supports the following two types:

SCSI

Specify the controller type as SCSI.

IDE

Specify the controller type as IDE.

-id *id*

(Optional) Specifies the ID of the controller mentioned in the controller type. For controller type IDE, the value is 0 or 1. For SCSI, the value is 0,1,2, or 3 based on the number of SCSI controllers added to the VM. If no id (or -1) is specified the command chooses the first controller of the specified type that has a channel available.

Default:

-1

-lun *lun*

(Optional) Specifies the logical unit of the controller in the form of <type> and <id>. For IDE, value is 0 or 1, for SCSI it is including the limit [0, 63]. If no unit number (or -1) is specified, the command chooses the available channel on the specified controller.

Default:

-1

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Create a Virtual Drive on a SCSI Controller

This example attaches a virtual hard drive image file to the first channel on the first SCSI controller.

```
dpmhv-setVMDisk -host hvserver -vm TestVM -type scsi -id 0 -lun 0 -drive HDD  
-path c:\VHDS\disk.vhd
```

Example: Attach a DVD to a VM

This example attaches a physical DVD-ROM drive to the VM, "Test VM."

```
dpmhv-setVMDisk -host hvserver -vm TestVM -type ide -id 0 -lun 1 -drive DVD  
-path D:
```

Example: Attach a Physical Host Disk to a SCSI Controller

This example gets the physical host disks and attaches them to the first available SCSI channel.

```
pd=dpmhv-getPhysDisks -host hvserver -retval -silent
```

```
dpmhv-setVMDisk -host hvserver -vm TestVM -type scsi -drive HDD -path (pd[0].Path)
```

dpmhv-setVMProperties (Cmdlet) Command--Set VM Properties

The dpmhv-setVMProperties command lets you set the properties of the VM.

This command has the following format:

```
dpmhv-setVMProperties  
-host hostname  
-vm vmname|-vmid vmguid  
[-name name]  
[-mem mem]  
[-cpus cpus]  
[-cpuidlimit limit]  
[-cpureserve reserve]  
[-cpulimit limit]  
[-cpuweight weight]  
[-startAction {none,auto,always}]  
[-startDelay delay]  
[-stopAction {save,off,shutdown}]  
[-recoveryAction {none,restart,revert}]  
[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-name *name*

Displays the name of the VM.

-mem *mem*

(Optional) Specifies the RAM memory in megabytes (MB) for the VM that you want to create.

Default: -1

-cpus *cpus*

(Optional) Specifies the number of CPU cores to assign to the VM.

-cpuidlimit *limit*

(Optional) Specifies the limit of CPU ID functionality of VM. This option improves the compatibility with legacy operating systems such as Windows NT.

Default: -1

-cpureserve *reserve*

(Optional) Specifies the percentage of the CPU that you want to reserve for the VM. If this option is not specified, Hyper-V server assigns CPU cycles based on the overall system usage.

Default: -1

-cpulimit *limit*

(Optional) Specifies the limit for the number of CPU resources used by a VM. This option improves the compatibility while moving VMs between physical hosts with different CPU capabilities.

Default: -1

-cpuweight *weight*

(Optional) Specifies the relative weight of the virtual machine. This option controls resource allocation when more than one VM is running. Valid entry: integer, 1-10000.

Default: -1

-startAction {*none,auto,always*}

(Optional) Specifies the action that you want to perform on the VM when the Hyper-V host starts up. Options include the following:

none

Performs no action.

auto

Starts the VM automatically.

Note: Use this option if the VM was running before the Hyper-V host is shut down.

always

Starts the VM always.

-startDelay *delay*

(Optional) Specifies the delay in seconds to start the VM after the Hyper-V host is up and running.

Default: -1

-stopAction {save,off,shutdown}

(Optional) Specifies the action that you want to perform on the VM when the physical Hyper-V shuts down. Options include the following:

save

Suspends the VM.

off

Power offs the VM.

shutdown

Shuts down the system.

-startDelay *delay*

(Optional) Specifies the delay in seconds to start the VM after the Hyper-V host is up and running.

Default: -1

-recoveryAction {none,restart,revert}

(Optional) Specifies the action to perform on the VM after the Hyper-V host restarts from unexpected shutdown. Options include the following:

none

Performs no action.

restart

Restart the VM.

revert

Reverts to the last snapshot of the VM.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Change number of Core CPUs

This example changes the number of CPU cores assigned to the VM to two. To run the command successfully, turn off the VM.

```
dpmhv-setVMProperties -host hvserver -vm TestVM -cpus 2
```


dpmhv-SetVMVirtFloppy (Cmdlet) Command--Connect a Virtual Floppy Image to the VM Floppy Drive

The `dpmhv-SetVMVirtFloppy` command lets you connect a virtual floppy image to the specified VM floppy drive on the Hyper-V host.

This command has the following format:

```
dpmhv-setVMVirtFloppy
-host hostname
-vm vmname|-vmid vmguid
[-path pathname]
[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-path *pathname*

(Optional) Specifies the path of the virtual floppy image (*.vfd) file that you want to create.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Example: Attach a Floppy Image to VM Floppy Drive on Hyper-V Host

This example attaches a floppy image, "floppy.vfd" to the VM floppy drive on the Hyper-V host, "TestVM."

```
dpmhv-setVMVirtFloppy
-host hvserver
-vm TestVM
-path c:\img\floppy.vfd
```

Example: Detaches a Floppy Image from VM Floppy Drive on Hyper-V Host

This example detaches a floppy image from the VM floppy drive on the Hyper-V host.

```
dpmhv-setVMVirtFloppy -host hvserver -vm TestVM
```

dpmhv-ShowClusterSharedVolumes (Funclet) Command--Show all Cluster Shared Volumes

The dpmhv-ShowClusterSharedVolumes command displays all cluster shared volumes on a Hyper-V server.

This command has the following format:

```
dpmhv-showClusterSharedVolumes  
-host hostname  
[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The dpmhv-ShowClusterSharedValumes command returns the current key and value pair from an array of objects.

Example: Show all Cluster Shared Volumes

This example returns a list of shared volumes on the host, "hvserver."

```
dpmhv-showClusterSharedVolumes -host hvserver
```

dpmhv-ShowDirectories (Funclet) Command--Show all Directories

The dpmhv-ShowDirectories command displays all subdirectories under a specified directory within a system. The specified system can be Hyper-V server or a Windows system with remote WMI enabled.

This command has the following format:

```
dpmhv-showDirectories  
-host hostname  
-dir dir  
[-retval]  
[-silent]  
[-detail]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-dir *dir*

Specifies the name of the directory.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the list of directories.

Return Value

The dpmhv-ShowDirectories command returns an array of objects which represents a subdirectory under the specified parent directory.

The following properties are returned in regular mode:

- Caption
- Drive
- FileName
- Name
- Path
- Writeable \$<<prop_detail
- AccessMask
- Archive
- CSCreationClassName
- CSName
- Caption
- Compressed
- CompressionMethod
- CreationClassName
- CreationDate
- Description
- Drive
- EightDotThreeFileName
- Encrypted
- EncryptionMethod
- Extension
- FSCreationClassName
- FSName
- FileName
- FileSize
- FileType
- Hidden
- InUseCount
- InstallDate

- LastAccessed
- LastModified
- Name
- Path
- Readable
- Status
- System
- Writeable

Example: Show all Directories

This example displays the subdirectories under the root directory (c: drive) of the host hvserver.

```
dpmhv-showDirectories
-host hvserver
-dir c:\
```

Example: Show all Directories under Program Files

This example displays the subdirectories of "C:\Program Files".

```
function walkDir(sPath)
{
  dirs = dpmhv-showDirectories -host hvserver -dir (sPath) -retval -silent
  for each (subdir in dirs)
  {
    ? subdir.Name
    walkDir(subdir.Name);
  }
}
walkDir("C:\Program Files");
```

dpmhv-ShowHosts (Funclet) Command--Show all Hyper-V Servers

The dpmhv-ShowHosts command displays all Hyper-V Servers.

This command has the following format:

```
dpmhv-showHosts
[-detail]
[-retval]
[-silent]
```

-detail

(Optional) Displays the details of the virtual networks (host networks).

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-ShowHosts` command returns an array of objects representing Hyper-V Servers. The property, "Name" is returned in the regular mode.

The following properties are returned in detail mode:

- AssignedNumaNodeList
- CIMV2_AdminPasswordStatus
- CIMV2_AutomaticManagedPagefile
- CIMV2_AutomaticResetBootOption
- CIMV2_AutomaticResetCapability
- CIMV2_BootOptionOnLimit
- CIMV2_BootOptionOnWatchDog
- CIMV2_BootROMSupported
- CIMV2_BootupState
- CIMV2_Caption
- CIMV2_ChassisBootupState
- CIMV2_CreationClassName
- CIMV2_CurrentTimeZone
- CIMV2_DNSHostName
- CIMV2_DaylightInEffect
- CIMV2_Description
- CIMV2_Domain
- CIMV2_DomainRole
- CIMV2_EnableDaylightSavingsTime
- CIMV2_FrontPanelResetStatus
- CIMV2_InfraredSupported
- CIMV2_InitialLoadInfo
- CIMV2_InstallDate
- CIMV2_KeyboardPasswordStatus
- CIMV2_LastLoadInfo
- CIMV2_Manufacturer
- CIMV2_Model
- CIMV2_Name
- CIMV2_NameFormat

- CIMV2_NetworkServerModeEnabled
- CIMV2_NumberOfLogicalProcessors
- CIMV2_NumberOfProcessors
- CIMV2_OEMLogoBitmap
- CIMV2_OEMStringArray
- CIMV2_PCSystemType
- CIMV2_PartOfDomain
- CIMV2_PauseAfterReset
- CIMV2_PowerManagementCapabilities
- CIMV2_PowerManagementSupported
- CIMV2_PowerOnPasswordStatus
- CIMV2_PowerState
- CIMV2_PowerSupplyState
- CIMV2_PrimaryOwnerContact
- CIMV2_PrimaryOwnerName
- CIMV2_ResetCapability
- CIMV2_ResetCount
- CIMV2_ResetLimit
- CIMV2_Roles
- CIMV2_Status
- CIMV2_SupportContactDescription
- CIMV2_SystemStartupDelay
- CIMV2_SystemStartupOptions
- CIMV2_SystemStartupSetting
- CIMV2_SystemType
- CIMV2_ThermalState
- CIMV2_TotalPhysicalMemory
- CIMV2_UserName
- CIMV2_WakeUpType
- CIMV2_Workgroup
- Caption
- ClusterName
- CreationClassName

- Dedicated
- Description
- ElementName
- EnabledDefault
- EnabledState
- HealthCritical
- HealthOk
- HealthState
- HostCPUUtilisationPercent
- HostMemoryAvailableMB
- IdentifyingDescriptions
- InstallDate
- IsHyperVClusterNode
- NET_ArpAlwaysSourceRoute
- NET_ArpUseEtherSNAP
- NET_Caption
- NET_DHCPEnabled
- NET_DHCPLeaseExpires
- NET_DHCPLeaseObtained
- NET_DHCPServer
- NET_DNSDomain
- NET_DNSDomainSuffixSearchOrder
- NET_DNSEnabledForWINSResolution
- NET_DNSHostName
- NET_DNSServerSearchOrder
- NET_DatabasePath
- NET_DeadGWDetectEnabled
- NET_DefaultIPGateway
- NET_DefaultTOS
- NET_DefaultTTL
- NET_Description
- NET_DomainDNSRegistrationEnabled
- NET_ForwardBufferMemory

- NET_FullDNSRegistrationEnabled
- NET_GatewayCostMetric
- NET_IGMPLevel
- NET_IPAddress
- NET_IPConnectionMetric
- NET_IPEnabled
- NET_IPFilterSecurityEnabled
- NET_IPPortSecurityEnabled
- NET_IPSecPermitIPProtocols
- NET_IPSecPermitTCPPorts
- NET_IPSecPermitUDPPorts
- NET_IPSubnet
- NET_IPUseZeroBroadcast
- NET_IPXAddress
- NET_IPXEnabled
- NET_IPXFrameType
- NET_IPXMediaType
- NET_IPXNetworkNumber
- NET_IPXVirtualNetNumber
- NET_Index
- NET_InterfaceIndex
- NET_KeepAliveInterval
- NET_KeepAliveTime
- NET_MACAddress
- NET_MTU
- NET_NumForwardPackets
- NET_PMTUBHDetectEnabled
- NET_PMTUDiscoveryEnabled
- NET_ServiceName
- NET_SettingID
- NET_TcpMaxConnectRetransmissions
- NET_TcpMaxDataRetransmissions
- NET_TcpNumConnections

- NET_TcpUseRFC1122UrgentPointer
- NET_TcpWindowSize
- NET_TcpipNetbiosOptions
- NET_WINSEnableLMHostsLookup
- NET_WINSHostLookupFile
- NET_WINSPrimaryServer
- NET_WINSScopeID
- NET_WINSSecondaryServer
- Name
- NameFormat
- OnTimeInMilliseconds
- OperationalStatus
- OtherDedicatedDescriptions
- OtherEnabledState
- OtherIdentifyingInfo
- Path
- PowerManagementCapabilities
- PrimaryOwnerContact
- PrimaryOwnerName
- ProcessID
- RequestedState
- ResetCapability
- Roles
- Status
- StatusDescriptions
- TimeOfLastConfigurationChange
- TimeOfLastStateChange
- VSMS_BiosLockString
- VSMS_Caption
- VSMS_DefaultExternalDataRoot
- VSMS_DefaultVirtualHardDiskPath
- VSMS_Description
- VSMS_ElementName

- VSMS_InstanceID
- VSMS_MaximumMacAddress
- VSMS_MinimumMacAddress
- VSMS_NumaSpanningEnabled
- VSMS_PrimaryOwnerContact
- VSMS_PrimaryOwnerName
- VSMS_ScopeOfResidence

Example: Show all Hyper-V Servers

This example returns a list of Hyper-V Servers.

```
dpmhv-showHosts
```

Example: Show all Hyper-V Servers and VMs on a Host

This example displays a list of Hyper-V Servers and displays VMs on each host.

```
a = dpmhv-showHosts -retval -silent
for each(h in a)
{
?
? "Host:", h.Name
dpmhv-showVMs -host (h.Name)
}
```

dpmhv-ShowLogicalDisks (Funclet) Command--Show all Logical Disks

The dpmhv-ShowLogicalDisks command displays all logical disks attached to a Windows system. The specified system can be Hyper-V server or a Windows system with remote WMI enabled. This command determines the location of a VM in the Windows file system of a remote Hyper-V server.

This command has the following format:

```
dpmhv-showLogicalDisks
-host hostname
[-retval]
[-silent]
[-detail]
```

-host hostname

Specifies the name of the host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the list of logical disks.

Return Value

The `dpmhv-ShowLogicalDisks` command returns an array of objects which represents the logical disk attached to a specified host.

The following properties are returned in regular mode:

- Caption
- DeviceID
- FreeSpace
- Name
- Path
- Size

The following properties are returned in detail mode:

- Access
- Availability
- BlockSize
- Caption
- Compressed
- ConfigManagerErrorCode
- ConfigManagerUserConfig
- CreationClassName
- Description
- DeviceID
- DriveType
- ErrorCleared
- ErrorDescription
- ErrorMethodology
- FileSystem
- FreeSpace
- InstallDate
- LastErrorCode
- MaximumComponentLength
- MediaType
- Name
- NumberOfBlocks

- PNPDeviceID
- Path
- PowerManagementCapabilities
- PowerManagementSupported
- ProviderName
- Purpose
- QuotasDisabled
- QuotasIncomplete
- QuotasRebuilding
- Size
- Status
- StatusInfo
- SupportsDiskQuotas
- SupportsFileBasedCompression
- SystemCreationClassName
- SystemName
- VolumeDirty
- VolumeName
- VolumeSerialNumber

Example: Show all Logical Disks

This example returns the list of logical disks attached to the host.

```
dpmhv-showLogicalDisks -host hvserver
```

dpmhv-ShowSCVMMHardwareProfiles (Funclet) Command--Show all SCVMM Hardware Profiles

The `dpmhv-ShowSCVMMHardwareProfiles` command displays a list of hardware profiles, defined by Microsoft System Center Virtual Machine Manager (SCVMM), that are available for a VM when creating a VM using an SCVMM template.

This command has the following format:

```
dpmhv-showSCVMMHardwareProfiles  
-scvmmHost scvmmHostname  
[-retval]  
[-silent]  
[-detail]
```

-scvmmHost *scvmmHostname*

(Optional) Specifies the host name of the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter is valid when you use SCVMM integration to provision VMs.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the list of directories.

Return Value

The `dpmhv-ShowSCVMMHardwareProfiles` command returns an array of objects which represents a hardware profile defined under specified library server.

The following properties are returned in regular mode:

- CPUCount
- Description
- Memory
- Name

The following properties are returned in detail mode:

- Accessibility
- AddedTime
- BootOrder[]
- Connection
- CPUCount
- CPUMax
- CPUReserve
- CPUType
- ID
- IsFullyCached
- MarkedForDeletion
- Name
- ServerConnection
- Description
- DiskIO
- Enabled
- ExpectedCPUUtilization
- HostDrive
- ID
- IsFullyCached
- IsHighlyAvailable
- LimitCPUForMigration

- LimitCPUFunctionality
- MarkedForDeletion
- Memory
- ModifiedTime
- MostRecentTask
- Name
- NetworkUtilization
- NumLockEnabled
- ObjectType
- Owner
- RelativeWeight
- ServerConnection
- CEIPOptIn
- CPUPriority
- Channel
- CompanyName
- DatabaseInstanceName
- DatabaseName
- DatabaseServerName
- DiskIOPriority
- EvaluationDaysLeft
- FQDN
- FullyQualifiedDomainName
- IsConnected
- IsEvaluationVersion
- IsWorkgroupEdition
- LibraryRefresherEnabled
- LibraryRefresherFrequency
- MOMReportingEnabled
- MOMReportingServerURL
- MemoryPriority
- MinimumSupportedAgentVersion
- NetworkPriority

- ObjectCache
- OpsMgrReportingEnabled
- OpsMgrReportingServerURL
- OpsMgrServer
- PROAutomationLevel
- PROMonitoringLevel
- PhysicalAddressRangeEnd
- PhysicalAddressRangeStart
- PlacementGoal
- ProductID
- ProductVersion
- Profile
- SelfServiceContactEmail
- ServerInterfaceVersion
- UserName
- VMConnectDefaultPort
- VMRCAccessAccount
- VMRCDefaultPort
- ShareSCSIBus
- UndoDisksEnabled
- VirtualCOMPorts[]
- VirtualDVDDrives[]
- VirtualFloppyDrive
- ServerConnection
- VirtualFloppyDisk
- VirtualNetworkAdapters[]
- VirtualSCSIAdapters[]

Example: Show all SCVMM Hardware Profiles

This example displays the hardware profiles on the SCVMM library server SCVMMLibSrv.

```
dpmhv -showSCVMMHardwareProfiles  
-scvmmHost SCVMMLibSrv
```

Example: Show Hardware Profiles for more than one CPU

This example displays all hardware profiles defined for more than one CPU core.

```
profs = dpmhv-showSCVMMHardwareProfiles -scvmmHost SCVMM -retval -silent
for each(prof in profs)
{
  if(prof.CPUCount > 1)
    ? prof.Name, "has more than 1 CPU cores"
}
```

dpmhv-ShowSCVMMOSProfiles (Funclet) Command--Show all Guest OS Profiles

The dpmhv-ShowSCVMMOSProfiles command displays all guest operating system (OS) profiles that are available for a VM while creating a VM using the SCVMM template.

This command has the following format:

```
dpmhv-showSCVMMOSProfiles
-scvmmHost scvmmHostname
[-retval]
[-silent]
[-detail]

-scvmmHost scvmmHostname
```

(Optional) Specifies the host name of the Microsoft System Center Virtual Machine Manager (SCVMM) library server. This parameter is valid when you use SCVMM integration to provision VMs.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the list of directories.

Return Value

The `dpmhv-ShowSCVMMOSProfiles` command returns an array of objects which represents a guest OS profile on the specified SCVMM library server.

The following properties are returned in regular mode:

- Admin
- AdminPasswordHasValue
- ComputerName
- Description
- DomainAdmin
- DomainAdminPasswordHasValue
- JoinDomain
- JoinWorkgroup
- Name
- OperatingSystem
- OrgName
- ProductKeyHasValue

The following properties are returned in detail mode:

- Accessibility
- AddedTime
- Admin
- AdminPasswordHasValue
- ComputerName
- Description
- DomainAdmin
- DomainAdminPasswordHasValue
- Enabled
- FullName
- GuiRunOnceCommands
- ID
- IsFullyCached
- JoinDomain
- JoinWorkgroup
- MarkedForDeletion

- MergeAnswerFile
- ModifiedTime
- MostRecentTask
- Name
- ObjectType
- OperatingSystem
- AllowsOrgNameInSysprep
- Architecture
- Description
- Edition
- ID
- IsCustomizationAllowed
- IsFullyCached
- MarkedForDeletion
- Name
- ProductType
- RequiresAdministratorAccountNameInSysprep
- RequiresPIDInSysprep
- RequiresXMLSysprepFormat
- ServerConnection
- Version
- OrgName
- Owner
- ProductKeyHasValue
- ServerConnection
- CEIPOptIn
- CPUPriority
- Channel
- CompanyName
- DatabaseInstanceName
- DatabaseName
- DatabaseServerName
- DiskIOPriority

- EvaluationDaysLeft
- FQDN
- FullyQualifiedDomainName
- IsConnected
- IsEvaluationVersion
- IsWorkgroupEdition
- LibraryRefresherEnabled
- LibraryRefresherFrequency
- MOMReportingEnabled
- MOMReportingServerURL
- MemoryPriority
- MinimumSupportedAgentVersion
- Name
- NetworkPriority
- ObjectCache
- OpsMgrReportingEnabled
- OpsMgrReportingServerURL
- OpsMgrServer
- PROAutomationLevel
- PROMonitoringLevel
- PhysicalAddressRangeEnd
- PhysicalAddressRangeStart
- PlacementGoal
- ProductID
- ProductVersion
- Profile
- SelfServiceContactEmail
- ServerInterfaceVersion
- UserName
- VMConnectDefaultPort
- VMRCAccessAccount
- VMRCDefaultPort
- SysprepScript

- TimeZone

Example: Show all Guest OS Profiles

This example displays the guest OS profiles on the SCVMM library server SCVMMLibSrv:

```
dpmhv-showSCVMMOSProfiles  
-scvmmHost SCVMMLibSrv
```

Example: Show all Guest OS Profiles without Password

This example displays all guest OS profiles without an Administrator password.

```
profs = dpmhv-showSCVMMOSProfiles -scvmmHost SCVMMLibSrv -retval -silent  
for each(prof in profs)  
{  
  if(prof.AdminPasswordHasValue=="false")  
    ? prof.Name,"does not have an Admin password defined"  
}
```

dpmhv-ShowTemplates (Funclet) Command--Show Templates on a Hyper-V Host

The dpmhv-ShowTemplates command returns a list of templates on the Hyper-V host or a Microsoft System Center Virtual Machine Manager (SCVMM) library server. The templates are the exported VMs which are managed in a template catalog and are used to provision new VMs.

This command has the following format:

```
dpmhv-showTemplates  
[-host hostname]  
[-scvmmHost scvmmHost]  
[-detail]  
[-retval]  
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-scvmmHost *scvmmHost*

(Optional) Specifies the name of the SCVMM library server. This parameter returns the templates created and maintained externally using SCVMM on the specified library server. This parameter is valid when you use SCVMM integration to provision VMs.

-detail

(Optional) Displays the details of the list of templates.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-ShowTemplates` command returns an array of objects representing the templates.

The following properties are returned in regular mode:

- ElementName
- Name

The following properties are returned in detail mode:

- DiskSpaceRequired
- ElementName
- MemoryAllocation
- Name
- NetworkInterfaceAddresses
- NetworkInterfaces
- NumberOfProcessors
- TemplateCreationDate
- TemplateDescription
- TemplateFileSystemPath
- TemplateSourceVM

Example: Get a List of VM Templates

This example returns the list of VM templates.

```
dpmhv-showTemplates  
-host hvserver
```

Example: Get detailed information of templates and process

This example returns a list of template with detailed information of templates and process.

```
tt = dpmhv-showTemplates -host hvserver -silent -retval  
for each(t in tt)  
? t.ElementName, t.Description
```

dpmhv-ShowVMDisks (Funclet) Command--Show all Virtual Disks

The dpmhv-ShowVMDisks command displays all virtual disks in a VM on a Hyper-V host server.

This command has the following format:

```
dpmhv-showVMDisks  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the list of virtual disks.

Return Value

The `dpmhv-ShowVMDisks` command returns an array of objects which represents the disks attached to a VM.

The following properties are returned in regular mode:

- Caption
- Connection
- ElementName
- HostResource
- ResourceSubType

The following properties are returned in detail mode:

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- VirtualQuantity
- VirtualSystemIdentifiers
- Weight

Example: Show all Virtual Disks of a VM

This example returns the list of virtual disks attached to the VM.

```
dpmhv-showVMDisks
-host hvserver
-vm TestVM
```

Example: Show all Disks and Virtual Hard Drive Image Files

This example returns the list of disk and virtual hard drive image files attached to the VM.

```
dd = dpmhv-showVMDisks -host hvserver -vm TestVM -detail -retval -silent
for each(d in dd)
{
  if(d.ResourceSubType=="Microsoft Virtual Hard Disk")
  {
    ? d.Connection
  }
}
```

dpmhv-showVMFloppy (Funclet) Command--Show the Properties of Floppy Drive

The dpmhv-showVMFloppy command lets you retrieve information about all floppy drives attached to every VM on the Hyper-V host.

This command has the following format:

```
dpmhv-showVMFloppy
-vm vmname | -vmid vmguid
-host hostname
[-retval]
[-silent]
[-detail]
```

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the information details of the floppy drive of the VM.

Return Value

The `dpmhv-showVMFloppy` command returns the objects representing the information details of the VM floppy drive properties.

The following properties are returned in regular mode:

- Caption
- Connection
- ElementName

The following properties are returned in detail mode:

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- VirtualQuantity
- VirtualSystemIdentifiers
- Weight

Example: Show all Floppy Drives of a VM

This example returns the list of floppy drives attached to the VM.

```
dpmhv-showVMFloppy -host hvserver -vm TestVM
```

dpmhv-showVMNICs (Funclet) Command--Show all Network Interface Cards

The dpmhv-showVMNICs command displays all Network Interface Cards (NIC) in the VM on a Hyper-V host.

This command has the following format:

```
dpmhv-showVMNICs  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the list of NICs.

Return Value

The `dpmhv-showVMNICs` command returns the objects.

The following properties are returned in regular mode:

- Address
- ElementName
- StaticMacAddress

The following properties are returned in detail mode:

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- StaticMacAddress
- VirtualQuantity
- VirtualSystemIdentifiers
- Weight

Example: Show VM NIC Information

This example shows the information of the NICs of the VM, "TestVM" on the host "hvserver."

```
dpmhv-showVMNICs -host hvserver -vm TestVM
//Find virtual network the first adapter of a VM is connected To
anic = dpmhv-showVMNICs -host hvserver -vm TestVM -detail -silent -retval
lines = anic[0].Connection.split(",");
  for each(line in lines)
  {
    if(line.startsWith("SystemName="))
    {
      nsw = line.substr(11);
      nsw = nsw.trim('');
      break;
    }
  }
// Get switches
asw = dpmhv-getHostSwitches -host hvserver -silent -retval
// Find match
for each(sw in asw)
{
  if(sw.Name == nsw)
  {
    ?? "NIC",anic[0].ElementName
    ? " is connected to switch", sw.ElementName
  }
}
```

dpmhv-showVMSCSI (Funclet) Command--Show all SCSI Controllers

The dpmhv-showVMSCSI command displays all SCSI controllers in the VM on a Hyper-V host. You can assign up to four SCSI controllers to a Hyper-V server.

This command has the following format:

```
dpmhv-showVMSCSI
-vm vmname|-vmid vmguid
-host hostname
[-retval]
[-silent]
[-detail]
```

-vm *vmname*

Specifies the name of the VM.

-vmid *vmguid*

Specifies the unique ID of the VM.

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

-detail

(Optional) Displays the details of the list of SCSI Controllers.

Return Value

The `dpmhv-showVMSCSI` command returns an array of objects representing the SCSI controllers of the specified VM.

The following properties are returned in regular mode:

- Caption
- ElementName

The following properties are returned in detail mode:

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- VirtualQuantity
- VirtualSystemIdentifiers
- Weight

Example: Show all SCSI Controllers

This example displays the basic details of all SCSI controllers of the VM, "TestVM" on the host, "hvserver."

```
dpmhv-showVMSCSI
-host hvserver
-vm TestVM
```

Example: Show all SCSI Controllers in Detail

This example displays the details of the SCSI controllers of the VM, "TestVM" on the host, "hvserver."

```
dpmhv-showVMSCSI -host hvserver -vm TestVM -detail
```

dpmhyperv ShowVMs (Funclet) Command--Get all VMs

The `dpmhv-showVMs` command returns a list of VMs on a Hyper-V host server. The command gets the names and corresponding GUIDs of the specified VM.

This command has the following format:

```
dpmhv-showVMs
-host hostname[-retval]
[-silent]
```

-host *hostname*

Specifies the name of the Hyper-V server host.

-retval

Returns a value for further processing.

-silent

Specifies not to direct the output to the screen.

Return Value

The `dpmhv-showVMs` command returns an array of objects representing the VMs on the specified host.

The objects include the following properties:

guid

Represents the unique ID of the VM.

name

Represents the display name of the VM.

Example: Get all VMs

This example returns a list of VMs and their corresponding GUIDs on the host, "hvserver."

```
dpmhv-showVMs
-host hvserver
```

Example: Show VM Properties for all VMs

This example returns displays the properties of all VMs on the host, "hvserver."

```
a = dpmhv-showVMs -host hvserver -retval -silent
for each(vm in a)
{
?
? "VM:", vm.ElementName
dpmhv-getVMProperties -host hvserver -vmid (vm.Name)
}
```

CA IBM LPAR AutoShell Commands

You can use the AutoShell to script and automate CA IBM LPAR commands and run actions based on the command results. Corresponding commands are also available in the CLI.

dpmlpar-cycle Command--Cycle a Logical Partition (Funclet)

The dpmlpar-cycle command powers on, powers off, resets, or suspends a logical partition.

This command has the following format:

```
dpmlpar-cycle
-powerop {activate|restart|shutdown}
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-type {delayed|immediate|os_shutdown|immediate_os_shutdown}]
[-profile_name profilename]
[-activate_bootmode {normal|open_firmware}]
[-activate_keylock {normal|manual}]
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-powerop {activate|restart|shutdown}

Specifies the power operation to perform on the LPAR. Options include the following:

activate

Turns on the LPAR.

restart

Turns off the LPAR, if necessary, and then turns it on.

shutdown

Turns off the LPAR.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {immediate|os_shutdown|immediate_os_shutdown}

Specifies to use the imaging operation type resource group (*res_group*) or individual resources (*individual_res*). Options include the following:

immediate

Shuts down the partition immediately. This option can cause undesirable results if the data has only been partially updated.

os_shutdown

Shuts down the partition by issuing the command for a typical shutdown. The partition must be imaged for this option to succeed.

immediate_os_shutdown

Shuts down the partition by issuing the operating system command to shut down the system as soon as possible. This command bypasses typical shutdown activities including sending messages to other users. The partition must be imaged for this option to succeed.

-profile_name *lparprofile* (HMC only)

[-profile_name *lparname*] (IVM only)

Specifies the partition profile to use when you activate the logical partition. Required for HMC. Optional for IVM, and if specified the name must match the partition name.

-activate_bootmode {normal|open_firmware}

(Optional) Specifies the keylock mode for the activate operation. Options include the following:

normal

Starts the partition in the typical manner.

open_firmware

Starts the partition and opens the open firmware prompt.

-activate_keylock {normal|manual|bypass}

(Optional) Specifies the keylock mode for the activate operation. Options include the following:

normal

Starts the partition in unattended mode and requires no user interaction during activation.

manual

Starts the partition in attended mode and requires user interaction during activation.

bypass

Does not activate the keylock mode.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Activate a Logical Partition with the Default Profile

This example activates the logical partition, "testlpar," using the default profile.

```
dpmlpar-cycle -powerop activate -hmc uslihmc -managed_system testComputer  
-partition_name testlpar -profile_name testlparprofile
```

Example: Activate a Logical Partition with a Specified Profile

This example activates the logical partition, "testlpar," using the profile testlparprofile.

```
dpmlpar-cycle -powerop activate -hmc uslihmc -managed_system testComputer  
-partition_name testlpar -profile_name testlparprofile
```

Example: Activate a Logical Partition to Open Firmware Prompt

This example activates the logical partition "testlpar" using the profile testlparprofile and opens to the open firmware prompt.

```
dpmlpar-cycle -powerop activate -hmc usliivm -managed_system testMS  
-partition_name testlpar -activate_bootmode open_firmware
```

Example: Delayed Shut Down of a Logical Partition

This example performs a delayed shutdown of the logical partition, "testlpar."

```
dpmlpar-cycle -hmc testivm -powerop shutdown -managed_system managedsys1  
-partition testlpar -type delayed
```

dpmlpar-delete Command--Delete a Logical Partition (Funclet)

The dpmlpar-delete command deletes a logical partition that is deactivated.

Important! Verify that you back up any important data *before* you issue this command. This command deletes the LPAR and the data files for the LPAR. The LPAR must be powered off when you issue this command.

This command has the following format:

```
dpmlpar-delete
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the name of a logical partition to delete. This partition must exist on the managed system.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Delete a Logical Partition

This example deletes the logical partition lpar01.

```
dpmlpar-delete -hmc hmc02 -managed_system system05 -partition_name lpar01
```

dpmlpar-getresources Command--Get LPAR Resources (Funclet)

The dpmlpar-getresources command retrieves memory units, processor units, or processors for IBM AIX LPARs.

This command has the following format:

```
dpmlpar-getresources
-resource {all|memory|processors|processor_units}
-hmc name
-managed_system managementsystemname
-partition_name partitionname
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-resource {all|memory|processors|processor_units}

Specifies whether a specific resource is retrieved or all resources. Options include the following:

all

Retrieves and displays all partition resources.

memory

Retrieves and displays only the memory resources for the partition.

processors

Retrieves and displays only the processor resources for the partition.

processor units

Retrieves and displays only the processor unit resources for the partition.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition for which resources are being listed. The partition must exist on the managed system.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Get all Resources for an LPAR

This example retrieves all resources for lpar06.

```
dpm_lpar-getresources -resource all -partition_name lpar06  
-hmc hmc_02 -managed_system abcsystem4
```

Example: Get the Memory Resources for an LPAR

This example retrieves the memory resources for lpartest01.

```
dpm_lpar-getresources -resource memory -partition_name lpartest01  
hmc ivm_02 -managed_system abcsystem2
```

dpmlpar-image Command--Create an IBM AIX LPAR (Funclet)

The dpmlpar-image command creates an IBM AIX LPAR using the HMC and a target IBM AIX managed system. You can optionally deploy the IBM AIX operating system after you create the LPAR.

This command has the following format:

```
dpmlpar-image
-hmc name -managed_system managementsystemname
-partition_name partitionname
-profile_name profilename
-min_mem size
-desired_mem size
-max_mem size

-proc_mode {shared|dedicated}
-min_proc_units units
-desired_proc_units units
-max_proc_units units
-min_procs number
-desired_procs number
-max_procs number

-max_virtual_slots number
-share_mode {capped|uncapped}
[-uncap_weight weight]
[-io_slot DRC index, {true|false}]
[-lpar_io_pool_id id {id, ...}]

[-virtual_serial_adapter slot_num, remote_lpar_name, remote_slot_num, is_required]
[-virtual_scsi_client_adapter slot_num, remote_lpar_name, remote_slot_num,
is_required] |
[-virtual_scsi_server_client_adapter virtual_io_server_name, backing_device_name,
client_slot_num, is_required]
[-virtual_eth_adapter slot_num, is_IEEE, port_vlan_id, (additional_vlan_id,
additional_vlan_id, ...), trunk_priority, is_required]

[-virtual_fc_server_client_adapter virtual_io_server_name, physical_fc_port_name,
client_slot_num, is_required]
[virtual_io_server_name,physical_fc_port_name,client_slot_num,is_required;...]]
[-verbose add_commandinfo]

[-create_logical_volume <false|true>]
[-logical_volume_size <value>]
[-volume_group_names <value>]
[-logical_volume_name <value>]
[-use_logical_volume_name_as_prefix {no|yes}]
[-default_vio_server_name vioserver]
```

```
[-sc URL]  
[-ws_remote_user username]  
[-ws_remote_password password]  
[-pre]  
[-post]
```

Additional parameters for provisioning AIX using a resource group

```
-provision_aix true  
-type res_group  
-res_group_name resourcegroupname  
-machine_res_name machineresourcenname  
-target_username targetusername  
[-target_password targetpassword]  
[-auth_file authorizationfilename]  
[-auth_comp componentID]  
-nim_master_host_name nimmasterhostname  
[-scalability_server servername]  
[-deploy_template templatename]  
-auto_deploy {yes|no}  
[-wait [timeout]]
```

Additional parameters for provisioning AIX using individual resources

```
-provision_aix true  
-type individual_res  
-machine_res_name machineresourcenname  
-lpp lppresource  
-spot spotresource  
-bosinst_data bosdata  
-resolv_conf resolveconf  
-fb_script fbscript  
-post_inst_scripts script1,script2,script3  
-target_username targetusername  
[-target_password targetpassword]  
[-auth_file authorizationfilename]  
[-auth_comp componentID]  
-nim_master_host_name nimmasterhostname  
[-scalability_server servername]  
[-deploy_template templatename]  
-auto_deploy {yes|no}  
[-wait [timeout]]
```

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the name of the logical partition to create.

-profile_name *lparprofile* (HMC only)**[-profile_name *lparname*] (IVM only)**

Specifies the partition profile to use when you activate the logical partition. Required for HMC. Optional for IVM, and if specified the name must match the partition name.

-min_mem *size*

Defines the minimum amount of memory for the partition.

-desired_mem *size*

Defines the desired amount of memory for the partition.

-max_mem *size*

Defines the maximum amount of memory for the partition.

-proc_mode {*shared*|*dedicated*}

Specifies the type of processor mode for the partition. Options include the following:

shared

Shares processor resources with other partitions.

dedicated

Specifies that the partition has dedicated processor resources.

-min_proc_units *units*

Defines the minimum number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-desired_proc_units *units*

Defines the assigned number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-max_proc_units *units*

Defines the maximum number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-min_procs *number*

Defines the minimum number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-desired_procs *number*

Defines the assigned number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-max_procs *number*

Defines the maximum number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-shared_mode {*capped*|*uncapped*}

Specifies whether the managed system allows the logical partition to use idle processing units that are not committed to another partition from the shared processor pool.

Note: This option can only be used with shared processors.

Options include the following:

capped

Specifies that the partition can only use the number of processing units that are committed to it.

uncapped

Specifies that the partition can use idle processor units from the shared processor pool when they are available.

-uncap_weight *weight*

(Optional) Defines a weighted average of processing priority when you select uncapped sharing mode.

Limits: 0 - 255

-max_virtual_slots *number*

Defines the maximum number of virtual adapters for this partition.

Default: 2

Limits: 2 - 65,536

-io_slot *DRC-Index*,{true|false}

(Optional) Specifies the I/O slot for a physical component. You can specify this value multiple times for different I/O devices.

DRC-Index

Specifies the slot dynamic reconfiguration connector (DRC) index.

true

Specifies that a DRC index is required for the I/O slot.

false

Specifies that a DRC index is not required for the I/O slot.

-io_pool_id *id* {*id* ...}

(Optional) Defines the group of I/O adapters that can be taken over and used by any of a specified group of logical partitions without any active intervention from the HMC. The group of partitions can be a comma-separated list of I/O pool IDs.

-virtual_serial_adapter *slot_num*, *remote_lpar_name*, *remote_slot_num*, *is_required*

(Optional) Defines the virtual serial adapters for this logical partition. The variable *is_required* can be set to true or false. Only client serial adapter is supported, but can be specified multiple times.

Default: Two server serial adapters created in slots 1 and 2.

Note: Not supported for creating an LPAR on IVM.

-virtual_eth_adapter_client *slot_num*, *is_IEEE*, *port_vlan_id1* (*vlanid2*,*vlanid3*,...), *trunk_priority*, *is_required*

(Optional) Defines the virtual ethernet adapters for this logical partition and can be specified multiple times. Optional values are *vlan_idn* and *trunk_priority*. The variables *is_IEEE* and *is_required* can be set to true or false. Additional *vlan_ids* must be comma-separated and enclosed in parentheses.

Note: When creating an LPAR on IVM, *slot_num* must be greater than or equal to 4.

Example: 4,false,1,,,true

-virtual_scsi_client_adapter *slot_num*, *remote_lpar_name*, *remote_slot_num*, *is_required*

(Optional) Defines the virtual client SCSI adapters for this logical partition and can be specified multiple times. The variable *is_required* can be set to true or false.

-virtual_scsi_server_client_adapter *virtual_io_server_name, backing_device_name, client_slot_num, is_required*

(Optional) Defines a virtual SCSI server device and virtual SCSI client device, which is automatically linked after the server device is created and the server slot number is identified. This parameter can be specified multiple times. The variable *is_required* can be set to true or false.

Example: CUST-VIOSERVER,hdisk5,3,true

Note: If you create Logical Volume using the `dpmlpar image` command and set the `-create_logical_volume` option to 'yes', the `-virtual_scsi_server_client_adapter` option is not required and shall be omitted.

-virtual_fc_server_client_adapter *virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required [;virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required;...]*

(Optional) Defines a virtual Fibre Channel (FC) client adapter, FC server adapter on VIO server, and associates the FC server adapter to the physical FC port. This option is not valid when the `-provision_aix` option is set to true. This parameter can be specified multiple times. The variable *is_required* can be set to true or false.

-create_logical_volume {no|yes}

(Optional) Specifies whether to create Logical Volume.

Default: no

Note: If you create Logical Volume using the `dpmlpar image` command and set the `-create_logical_volume` option to 'yes', the `-virtual_scsi_server_client_adapter` option is not required and shall be omitted.

-logical_volume_size *logical_volume_size*

(Optional) Defines the size of the Logical Volume in MB.

-volume_group_names *volume_group_1, volume_group_2, volume_group_3*

(Optional) Specifies the Logical Volume group names as a comma-separated list.

-logical_volume_name *logical_volume_name*

(Optional) Defines the name of the Logical Volume.

-use_logical_volume_name_as_prefix {no|yes}

(Optional) Specifies whether to use the Logical Volume Name as a prefix.

Default: no

-default_vio_server_name *vioserver*

(Optional) Specifies the VIO server that is used for creation of Logical Volumes.

-itcm_server *itcm_servername*

(Optional) Specifies the name of the CA ITCM Server.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Required parameters for provisioning**-auth_file *authorizationfilename***

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-auto_deploy {*yes|no*}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-bosinst_data *bos_install_data_resource*

Specifies a file that contains information for the base operating system (BOS) installation program. Optional if `-install_type` is `mksysb`. Required if `-install_type` is `rte`.

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-fb_script *fbscript*

(Optional) Defines the name of the file to use to configure devices when a NIM client is initially booting after the BOS installation process is complete.

-lpp *lpp_resource*

Specifies the licensed program product files to use for an imaging request. Optional if `-install_type` is `mksysb`. Required if `-install_type` is `rte`.

-machine_res_name *machineresourcenam*

Defines the name of the machine resource. This name must be predefined at the NIM Master.

-nim_master_host_name *nimmasterhostname*

Defines the NIM master host name to perform the image deployment.

-post_inst_scripts *script1,script2,script3*

(Optional) Specifies a comma-separated list of scripts to run after installation.

-provision_aix {true|false}

Indicates whether the partition is imaged using NIM, after you create it.

true

Uses NIM to image the partition that you created. If you set to true, the NIM resource group or NIM individual resource parameters are used. See the `dpmnim image Command|Deploy an IBM AIX Image Using a Resource Group` and `dpmnim image Command|Deploy an IBM AIX Image Using an Individual Resource`. The job ID is returned when NIM provisioning starts.

false

Does not use NIM to image the partition that you created. No job ID is returned.

-res_group_name *resource group name*

Defines the name of the resource group.

-resolve_conf *resolveconf*

(Optional) Defines a file that contains valid `/etc/resolv.conf` entries that define Domain Name Protocol name-server information for local resolver routines.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-spot *spotResource* [*mksysbResource*]

Defines the shared product object tree to use for an imaging request.

-target_password *targetpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying the image. If you do not specify the password, it is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-target_username *targetusername*

Defines the user name used for deploying agents to the target host server to which you are deploying the image.

-type {*res_group* | *individual_res*}

Specifies to use the imaging operation type resource group or individual resources.

res_group

Specifies to use the resource group for imaging.

individual_res

Specifies to use individual resources for imaging.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

Example: Create an IBM AIX Logical Partition

This example creates a logical partition, but does not provision it.

```
dpmlpar-image -hmc ivm01 -managed_system testMS -partition_name lpartest01
-profile_name Default -max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5 -proc_mode shared
-share_mode uncapped -uncap_weight 123 -min_proc_units .5 -max_proc_units 3.25
-desired_proc_units 1.75 -virtual_serial_adapter 2,serial_partition,1,true
-virtual_scsi_client_adapter 3,scsi_partition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true -io_slot 21020003,false
```

Example: Create an IBM AIX Logical Partition and Image it Using Individual Resources

This example creates a logical partition and provisions it using individual resources.

```
dpmlpar-image -hmc hmc01 -managed_system testMS -partition_name IRTEST
-profile_name Default -max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 1 -max_procs 1 -proc_mode dedicated
-virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),0,true -io_slot 21020003,false
-provision_aix true -type individual_res -lpp 530lpp_res -spot 530spot_res
-bosinst_data 530_bid_ow -resolv_conf master_net_conf -post_inst_scripts piScript
-machine_res_name MachineName -nim_master_host_name machine.mydomain.com
-auto_deploy no -target_username root -target_password password
```

Example: Create an IBM AIX Logical Partition and Image it Using a Resource Group

This example creates a logical partition and provisions it using a resource group.

```
dpmlpar-image -hmc hmc01 -managed_system testMS -partition_name RGTEST
-profile_name Default -max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5 -proc_mode dedicated
-virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true -io_slot 21020003,false
-provision_aix true -type res_group -res_group_name Res_grp -machine_res_name
Machine
-nim_master_host_name machine.mydomain.com -auto_deploy no
-target_username root -target_password password
```

dpmlpar-imgjobcheck Command--Retrieve Status of IBM AIX Imaging Job (Funclet)

The `dpmlpar-imgjobcheck` command retrieves the status of the IBM AIX logical partition imaging job for a specific CA Server Automation job ID.

This command has the following format:

```
dpmlpar-imgjobcheck -status jobID [-verbose add_commandinfo]
```

-status *jobID*

Specifies the CA Server Automation job ID used to obtain the job status.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

Example: Retrieve the Status of the IBM AIX Imaging Job Using the Job ID

This example obtains the job status of the IBM AIX logical partition imaging job using the CA Server Automation job ID 42.

```
dpmlpar-imgjobcheck -status 42
```

dpmlpar-list Command--List HMC Resources (Funclet)

The dpmlpar-list command lists information from the Hardware Management Console.

This command has the following format:

```
dpmlpar-list
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-display {managed_systems | managed_system_details | partitions | io_components |
backing_devices | scsi_adapters | partition_details | profiles | wwpns}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-display

**{*managed_systems*|*managed_system_details*|*partitions*|*io_components*|*backing_de*
vices|*scsi_adapters*|*partition_details*|*profiles*|*wwpns*}**

Specifies which resources to list. Options include the following:

managed_systems

Lists the managed systems controlled by the management hardware console.

managed_system_details

Provides detailed information about a managed system.

partitions

Lists the partitions on a managed system.

io_components

Lists the I/O components of a managed system.

backing_devices

Lists the devices that are available to be attached as backing devices to the Server Virtual SCSI Adapter.

scsi_adapters

Lists the SCSI adapters of a managed system.

partition_details

Lists the details for a partition.

profiles

Lists the list of profiles for a partition.

wwpns

Displays the worldwide port names for a partition.

-hmc *name*

Specifies the HMC/IVM for which to list information.

-managed_system *managedsystemname*

(Optional) Specifies the managed system on which the logical partition resides for which you want to list information.

Note: Do not use with `managed_systems` option.

-partition_name *partitionname*

(Optional) Specifies the logical partition for which you want to list information.

Note: Do not use with `managed_systems` option.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Display the List of Managed Systems

This example displays the managed systems for the hardware management console "labhmc."

```
dpmlpar-list -display managed_systems -hmc labhmc
```

Example: Display the SCSI Adapters

This example displays the SCSI adapters for the managed system "testMS."

```
dpmlpar-list -display scsi_adapters -hmc labhmc
-managed_system testMS
```

Example: Display Partition Details

This example displays the details for the logical partition "testlpar."

```
dpmlpar-list -display partition_details -hmc uslihmc
-managed_system testMS -partition_name testlpar
```

dpmlpar-setresources Command--Add Memory Resources (Funclet)

The dpmlpar-setresources command adjusts memory units for IBM AIX LPARs.

This command has the following format:

```
dpmlpar-setresources
-add_memory value
-hmc name
-managed_system managementsystemname
-partition_name partitionname
-type {dynamic|all}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-add_memory *value*

Defines the amount of memory to add to the partition.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic* | *all*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Add Memory to a Logical Partition

This example adds 128 MB of memory to the logical partition "lpartest01."

```
dpm\lpar-setresources -add_memory 128 -partition_name lpartest01  
-hmc uni02 -managed_system usil01system1 -type dynamic
```


Example: Add Memory to a Logical Partition and Update a Profile

This example adds 128 MB of memory to the logical partition "lptest01" and updates the partition profile file.

```
dpmlpar-setresources -add_memory 128 -partition_name lptest01
-hmc uni02 -managed_system usil01system1 -type all -profile_name lptest01
```

dpmlpar-setresources Command--Add Processor Resources (Funclet)

The dpmlpar-setresources command adds processors and processor units for IBM AIX LPARs.

This command has the following format:

```
dpmlpar-setresources
{-add_processors|add_processor_units} value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-add_processors value

Defines the number of processors to add to the partition.

-add_processor_units value

Defines the processor units to add to the partition.

Limits: .01 increments

-hmc name

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system managedsystemname

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name partitionname

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic* | *all*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Add Processors for a Partition and Update Profile

This example adds a processor to the partition "lpartest01" and updates the partition profile.

```
dpm\par-setresources -add_processors 1 -partition_name lpartest01  
-hmc uni02 -managed_system MSsystem1 -type all -profile_name lpartest01
```

Example: Add Processors and Processor Units for a Partition

This example temporarily adds the processor and processor unit resources for the partition "lpartest01."

```
dpm\par-setresources -add_processors 1 -add_processor_units 4.5  
partition_name lpartest01 -hmc uni02 -managed_system MSsystem1 -type dynamic
```

dpmlpar-setresources Command--Subtract Memory Resources (Funclet)

The dpmlpar-setresources command removes memory units for IBM AIX LPARs.

This command has the following format:

```
dpmlpar-setresources
-subtract_memory value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-subtract_memory *value*

Defines the amount of memory to remove from the partition.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic|all*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-verbose *add_commandinfo*

Provides additional information about how to execute the command.

-sc *URL*

(Optional) Specifies the URL of the service controller.

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Subtract Memory from a Logical Partition

This example subtracts 128 MB of memory from the logical partition "lpartest01."

```
dpm\par-setresources -subtract_memory 128 -partition_name lpartest01  
-hmc con02 -managed_system MSsystem1 -type dynamic
```

Example: Subtract Memory from a Logical Partition and Update a Profile

This example subtracts 128 MB of memory from the logical partition "lpartest01" and updates the partition profile file.

```
dpm\par-setresources -subtract_memory 128 -partition_name lpartest01  
-hmc con02 -managed_system MSsystem1 -type all
```

dpmlpar-setresources Command--Subtract Processor Resources (Funclet)

The dpmlpar-setresources command removes processors and processor units from IBM AIX LPARs.

This command has the following format:

```
dpmlpar-setresources
{-subtract_processors|subtract_processor_units} value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-subtract_processors *value*

Defines the number of processors to remove from the partition.

-subtract_processor_units *value*

Defines the processor units to remove from the partition.

Limits: .01 increments

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {dynamic|all}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-verbose add_commandinfo

Provides additional information about how to execute the command.

-sc URL

(Optional) Specifies the URL of the service controller.

-ws_remote_user username

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password password

(Optional) Specifies the CA Server Automation remote password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Subtract Processors From a Partition and Update Profile

This example subtracts a processor from the partition "lpartest01" and updates the partition profile

```
dpmlpar-setresources -subtract_processors 1 -partition_name lpartest01  
-hmc hmc02 -managed_system MSsystem1 -type all
```

Example: Subtract Processors and Processor Units From a Partition

This example temporarily adjusts the processor and processor unit resources for the partition "lpartest01."

```
dpmlpar-setresources -subtract_processors 1 -subtract_processor_units 4.5  
-partition_name lpartest01 -hmc ivm02 -managed_system MSsystem1 -type dynamic
```

CA Oracle Solaris Zones AutoShell Commands

You can use the AutoShell to script and automate CA Oracle Solaris Zones commands and run actions based on the command results. Corresponding commands are also available in the CLI.

dpmzone-associateproject Command--Associate a Task with a Project (Funclet)

The `dpmzone-associateproject` command associates a task with a project. Projects and tasks are the basic entities used to identify workloads in a Solaris 10 operating system. Projects are collections of tasks, which are collections of processes. Each process belongs to only one task, and each task belongs to only one project.

This command has the following format:

```
dpmzone-associateproject
-host hostname -name zonename
-proj_name pname -task_id tid
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone that provides the project.

-proj_name *pname*

Specifies the project.

-task_id *tid*

Specifies the ID number of the task.

Example

Associate a task with a project:

```
dpmzone-associateproject -host SolarisServer2 -name myzone1 -project_name myproject1
-task_id 1954
```

dpmzone-clonezone Command--Clone a Zone (Funclet)

The dpmzone-clonezone command creates a duplicate zone from an existing zone. The source zone must be halted to start the cloning process. Cloning occurs on the Solaris 10 server and may take some time.

This command has the following format:

```
dpmzone-clonezone  
-host hostname  
-name zonename  
-new_name nzonename  
-new_path npath
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone that you want to clone.

-new_name *nzonename*

Defines the new zone name.

-new_path *npath*

Defines the name of the path from the global zone to the current zone.

Example

Clone a zone on host SolarisServer2:

```
dpmzone-clonezone -host SolarisServer2 -name myzone1 -new_name myzone2  
-new_path /opt/zones/myzone2
```


dpmzone-createandinstallzone Command--Create and Install a Zone

The `createandinstallzone` command creates and installs a zone with custom parameters on the Solaris host.

This command has the following format:

```
dpmzone-createandinstallzone  
[-sc sc_host]  
-host hostname  
-name zonename  
-path zonepath  
-type type  
[-archive_path path]  
[-autoboot]  
[-if_type name]  
[-ip ip]  
[-pool_name pool]  
[-sched_type sched]  
[-phy_mem pmem]  
[-swap_mem smem]  
[-lock_mem lmem]  
[-desc desc]  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Defines the name of the new zone.

-path *zonepath*

Defines the path of the new zone.

-type *type*

Specifies the type of the new zone. Options include the following:

native

Creates a non-global zone with a Solaris 10 operating environment for running applications.

whole-root

Creates a whole root zone that does not inherit packages.

branded

Creates a non-global zone that contains a non-native operating environment for running applications.

Default: native

-archive_path *path*

(Optional) Specifies the path of the operating environment installer for a branded zone.

Limits: This argument is required for branded zones, but not required for native zones.

-autoboot

(Optional) Specifies that a zone boots automatically at system boot.

Note: If the zones service is disabled on the server, the zone does not autoboot, regardless of the setting of this property. You can enable the service by using the following command:

```
svcadm enable svc:/system/zones:default
```

-if_type *name*

(Optional) Specifies the type of the network interface used by the zone, for example, eri0.

-ip *ip*

(Optional) Defines the IP address of the zone.

-pool_name *pool*

(Optional) Specifies the name of the pool with which you associate the zone.

-sched_type *shed*

(Optional) Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Valid values include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. ts is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

-phy_mem *pmem*

(Optional) Defines the physical memory that is assigned to the zone. A scale (K, M, G, T) can be applied to the value of this number, for example, 1M is one megabyte.

-swap_mem *smem*

(Optional) Defines the swap memory that is assigned to the zone.

-lock_mem *lmem*

(Optional) Defines the locked memory that is assigned to the zone. Locked memory cannot be paged.

-desc *description*

(Optional) Defines a description for the zone.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmzone-createpool Command--Create a Resource Pool (Funclet)

The dpmzone-createpool command creates a resource pool on a Solaris 10 host.

This command has the following format:

```
dpmzone-createpool
-host hostname
-pset_name pset
-cpu_min mincpu
-cpu_max maxcpu
-pool_name pool
-sched_type shed
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-pset_name *pset*

Defines the processor set. Each processor set (grouping of CPUs) can contain zero or more CPUs.

-cpu_min *mincpu*

Defines the minimum number of CPUs in the processor set.

-cpu_max *maxcpu*

Defines the maximum number of CPUs in the processor set.

-pool_name *pool*

Defines the name of the pool with which you associate the zone.

-sched_type *shed*

Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Options include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. ts is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

Example

Create the resource pool POOL1 on host SolarisServer2.

```
dpmzone-createpool -host SolarisServer2 -pset_name PSET1 -cpu_min 1 -cpu_max 20  
-pool_name POOL1 -sched_type FSS
```

dpmzone-createproject Command--Create a Project (Funclet)

The dpmzone-createproject command creates a project in a zone.

This command has the following format:

```
dpmzone-createproject  
-host hostname -name zonename  
-user_id uid  
[-project_id pid]  
-proj_name pname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone name.

-user_id *uid*

Specifies one of the users of the zone.

Example: root

-project_id *pid*

(Optional) Defines the project ID. You can assign the ID or let the system automatically generate one.

-proj_name *pname*

Defines the name of the new project.

Example

Create a project in a zone:

```
dpmzone-createproject -host SolarisServer2 -name myzone1 -user_id root  
-proj_name myProject
```

dpmzone-createzone Command--Create a Zone (Funclet)

The dpmzone-createzone command creates a zone with custom parameters on the Solaris 10 host.

This command has the following format:

```
dpmzone-createzone
-host hostname
-name zonename
-path zonepath
-type type
[-autoboot]
[-if_type name]
[-ip ip]
[-pool_name pool]
[-sched_type sched]
[-phy_mem pmem]
[-swap_mem smem]
[-lock_mem lmem]
[-desc desc]
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Defines the name of the new zone.

-path *zonepath*

Defines the pathname from the global zone to the current zone.

-type *type*

Specifies the type of the new zone. Options include the following:

native

Creates a non-global zone with a Solaris 10 operating environment for running applications.

whole-root

Creates a whole root zone that does not inherit packages.

branded

Creates a non-global zone that contains a non-native operating environment for running applications.

Default: native

-autoboot

Specifies that a zone boots automatically at system boot.

Note: If the zones service is disabled on the server, the zone does not autoboot, regardless of the setting of this property. You can enable the service by using the following command:

```
svcadm enable svc:/system/zones:default
```

-if_type *name*

Specifies the network interface used by the zone, for example, eri0.

-ip *ip*

(Optional) Defines the IP address of the zone.

-pool_name *pool*

Specifies the pool with which you associate the zone.

-sched_type *shed*

Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Options include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. ts is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

-phy_mem *pmem*

Defines the physical memory that is assigned to the zone. A scale (K, M, G, T) can be applied to the value of this number, for example, 1M is one megabyte.

-swap_mem *smem*

Defines the swap memory that is assigned to the zone.

-lock_mem *lmem*

Defines the locked memory that is assigned to the zone. Locked memory cannot be paged.

-desc *description*

(Optional) Defines a description of the zone.

Example

Create a zone called WebServer4:

```
dpmzone-createzone -host MySolarisServer -name WebServer4 -path /Zones/Apache
-type native -autoboot -if_type eri0 -ip 192.168.100.100 -pool_name WebServer
-sched_type fss -phy_mem 1024M -swap_mem 2048M
```

dpmzone-customcommand Command--Run a Custom Command (Funclet)

The dpmzone-customcommand command lets you run external commands or scripts on the Solaris server.

Note: For information about restricted external commands, see the Custom Commands appendix in this guide.

This command has the following format:

```
dpmzone-customcommand
-host hostname
-cmd command
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-cmd *command*

Defines which program or script to run.

Example

Run a command on host SolarisServer2.

```
dpmzone-customcommand -host SolarisServer2 -cmd
/usr/local/lab/scripts/test_connections.pl
```

dpmzone-deleteproject Command--Delete a Project (Funclet)

The dpmzone-deleteproject command deletes a project from a zone.

This command has the following format:

```
dpmzone-deleteproject
-host hostname
-name zonename
-proj_name pname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone that is associated with the project.

-proj_name *pname*

Specifies the project that you want to delete.

Example

Delete a project:

```
dpmzone-deleteproject -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone-deletezone Command--Delete a Zone (Funclet)

The dpmzone-deletezone command lets you delete a zone on the Solaris server.

This command has the following format:

```
dpmzone-deletezone  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to delete.

Example

Delete a zone on host SolarisServer2.

```
dpmzone-deletezone -host SolarisServer2 -name myzone
```

dpmzone-installzone Command--Install a Zone (Funclet)

The dpmzone-installzone command starts the installation command on the Solaris 10 server. The installation process may take some time. Check the status of the zone with the showprocesses command to confirm that the installation process is running.

This command has the following format:

```
dpmzone-installzone  
-host hostname  
-name zonename  
[-archive_path path]
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone that you want to install.

-archive_path *path*

(Optional) Defines the path of the operating environment installer for a branded zone.

Limits: This argument is required for branded zones, but not required for native zones.

Examples

Install a native zone:

```
dpmzone-installzone  
-host SolarisServer  
-name MyZone3
```

Install a branded zone with Solaris 8 on SolarisServer:

```
dpmzone-installzone -host SolarisServer -name MyZone4  
-archive_path /opt/zoneos/branded/Solaris8Installer
```

dpmzone-movezone Command--Move a Zone (Funclet)

The dpmzone-movezone command lets you move a zone from one path to a new path.

This command has the following format:

```
dpmzone-movezone  
-host hostname  
-name zonename  
-new_path path
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to move to a new location.

-new_path *path*

Defines the new path for the zone to move.

Example

Move a zone on host SolarisServer2:

```
dpmzone-movezone -host SolarisServer2 -name myzone1  
-new_path /opt/zones/myzone1
```

dpmzone-rebootzone Command--Reboot a Zone (Funclet)

The dpmzone-rebootzone command lets you reboot a zone on an Oracle Solaris server.

This command has the following format:

```
dpmzone-rebootzone  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to reboot.

Example

Reboot a zone on host SolarisServer2:

```
dpmzone-rebootzone -host SolarisServer2 -name myzone
```

dpmzone-renamezone Command--Rename a Zone (Funclet)

The dpmzone-renamezone command lets you rename a zone on the Oracle Solaris server.

This command has the following format:

```
dpmzone-renamezone  
-host hostname  
-name czonename  
-new_name nzonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *czonename*

Specifies the current zone name.

-new_name *nzonename*

Defines the new zone name.

Example

Rename a zone on host SolarisServer2:

```
dpmzone-renamezone -host SolarisServer2 -name myzone -new_name yourzone
```

dpmzone-setpoolscheduler Command--Set the Pool Scheduler (Funclet)

The `dpmzone-setpoolscheduler` command sets the type of scheduler to use for the resource pool. Resource pools can have two scheduler types, the Fair Share Scheduler (FSS) and the Time Share Scheduler (TS).

This command has the following format:

```
dpmzone-setpoolscheduler
-host hostname
-pool_name poolname
-sched_type shed
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-pool_name *poolname*

Specifies an associated group of resources that can be partitioned.

-sched_type *shed*

Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Options include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. `ts` is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

Examples

Set the scheduling for the Resource Pool to the Fair Share Scheduler:

```
dpmzone-setpoolscheduler
-host SolarisServer
-pool_name test_pool
-sched_type FSS
```

Set the scheduling for the Resource Pool to the Time Share Scheduler.

```
dpmzone-setpoolscheduler
-host SolarisServer
-pool_name test_pool
-sched_type TS
```

dpmzone-setprocessorset Command--Set the Maximum Number of CPUs for a Processor Set (Funclet)

The `dpmzone-setprocessorset` command sets the maximum number of CPUs for a processor set.

This command has the following format:

```
dpmzone-setprocessorset
-host hostname
-processor_set_name psetname
-cpu_max maxcpu
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-processor_set_name *psetname*

Specifies the processor set. Each pset (grouping of CPUs) can contain zero or more processors.

-cpu_max *maxcpu*

Defines the maximum number of CPUs in the processor set.

Example

Set the maximum number of CPUs for a processor set:

```
dpmzone-setprocessorset -host SolarisServer -processor_set_name test_pset -cpu_max
64
```

dpmzone-showinterfaces Command--Show Network Interfaces (Funclet)

The `dpmzone-showinterfaces` command lists all the network interfaces present on the Solaris 10 host.

This command has the following format:

```
dpmzone-showinterfaces
-host hostname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

Example

List all network interfaces on the host `SolarisServer2`:

```
dpmzone-showinterfaces -host SolarisServer2
```

dpmzone-showpools Command--Show Resource Pools (Funclet)

The dpmzone-showpools command lists all the resource pools present on the Solaris 10 host.

This command has the following format:

```
dpmzone-showpools  
-host hostname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

Example

List all resource pools on the host SolarisServer2:

```
dpmzone-showpools -host SolarisServer2
```

dpmzone-showprojects Command--List the Properties of a Project (Funclet)

The dpmzone-showprojects command lists the properties of a project.

This command has the following format:

```
dpmzone-showprojects  
-host hostname  
-name zonename  
-proj_name pname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-proj_name *pname*

Specifies the project.

Example

List the properties of a project:

```
dpmzone-showprojects -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone-showresourceset Command--Show Resource Sets (Funclet)

The dpmzone-showresourceset command lists the resources of a resource pool.

This command has the following format:

```
dpmzone-showresourceset  
-host hostname  
-pool_name poolname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-pool_name *poolname*

Specifies an associated group of resources that can be partitioned.

Example

List the resource sets in MyZonesResourcePool:

```
dpmzone-showresource_set -host SolarisServer2 -pool_name MyZonesResourcePool
```

dpmzone-showtasks Command--List the Tasks of a Project (Funclet)

The dpmzone-showtasks command lists the tasks of a project.

This command has the following format:

```
dpmzone-showtasks  
-host hostname  
-name zonename  
-proj_name pname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-proj_name *pname*

Specifies the project.

Example

List the tasks of a project:

```
dpmzone-showtasks -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone-showzoneinterfaces Command--Show Network Interfaces of a Zone (Funclet)

The dpmzone-showzoneinterfaces command lists all the network interfaces on a zone.

This command has the following format:

```
dpmzone-showzoneinterfaces  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

Example

List all network interfaces on the zone MyZone3 on SolarisServer2:

```
dpmzone-showzoneinterfaces -host SolarisServer2 -name MyZone3
```

dpmzone-showzoneprocess Command--Show Processes of a Zone (Funclet)

The dpmzone-showzoneprocess command lists all processes on a zone.

This command has the following format:

```
dpmzone-showzoneprocess  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

Example

List the processes of zone MyZone3 on SolarisServer2:

```
dpmzone-showzoneprocess -host SolarisServer2 -name MyZone3
```


dpmzone-showzoneprojects Command--Show Projects of a Zone (Funclet)

The dpmzone-showzoneprojects command lists all the projects on a zone.

This command has the following format:

```
dpmzone-showzoneprojects  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

Example

List all projects of zone MyZone3 on SolarisServer2:

```
dpmzone-showzoneprojects -host SolarisServer2 -name MyZone3
```

dpmzone-showzones Command--Show Zones (Funclet)

The dpmzone-showzones command lists all the zones present on the Solaris host.

This command has the following format:

```
dpmzone-showzones  
-host hostname
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

Example

List all zones on host SolarisServer2:

```
dpmzone-showzones -host SolarisServer2
```

dpmzone-startzone Command--Start a Zone (Funclet)

The dpmzone-startzone command lets you start a zone on the Solaris server.

This command has the following format:

```
dpmzone-startzone  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to start.

Example

Start a zone on host SolarisServer2:

```
dpmzone-startzone -host SolarisServer2 -name myzone
```

dpmzone-stopzone Command--Stop a Zone (Funclet)

The dpmzone-stopzone command lets you stop a zone on the Solaris server.

This command has the following format:

```
dpmzone-stopzone  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to stop.

Example: Stop a Zone on the Host

Stop a zone on host SolarisServer2:

```
dpmzone-stopzone -host SolarisServer2 -name myzone
```

dpmzone-uninstallzone Command--Uninstall a Zone (Funclet)

The dpmzone-uninstallzone command lets you uninstall a zone on the Solaris server.

This command has the following format:

```
dpmzone-uninstallzone  
-host hostname  
-name zonename
```

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to uninstall.

Example

Uninstall a zone on host SolarisServer2:

```
dpmzone-uninstallzone -host SolarisServer2 -name myzone
```

Deployment and Policy Configuration AutoShell Commands

This section details the Deployment and Policy Configuration AutoShell commands in an alphabetic order. You can access ISM functions from the AutoShell command line. You can display the full list of functions using the command, help ism-* from the AutoShell command line prompt.

ism-applyPolicyForService Command--Apply a Policy to all Systems (Funclet)

The ism-applyPolicyForService command applies a policy to all computer systems in a managed service.

This command has the following format:

```
ism-applyPolicyForService  
-policyName policy_name  
-policyType policy_type  
-serviceName service_name  
-templateName templateName  
-templateOs templateOs
```

-policyName *policy_name*

Specifies the policy name to apply.

Default: Null

-policyType *policy_type*

Specifies the policy type.

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as "Enterprise\\Data Center\\servicename", instead of "Enterprise\Data Center\servicename".

-templateName *templateName*

Specifies the name of the template.

-templateOs *templateOs*

Specifies the policy template operating system.

Example: Apply a Policy to all Systems

This example applies the policy, SystemEDGE to the managed service, CA Virtual Assurance.

```
ism-applyPolicyForService -policyName policy5 -policyType SystemEDGE -serviceName "Enterprise\\Data Center\\CA Virtual Assurance Services" -templateName template2 -templateOs windows
```

ism-applyPolicies Command--Apply Policies to a Group of Hosts (Funclet)

The ism-applyPolicies command applies the specified policy to a group of hosts.

This command has the following format:

```
ism-applyPolicies  
-policyName policy_name  
-policyType policy_type  
-hostName host_name  
[-timeout time]
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

-hostName *host_name*

Specifies the name of the host computer.

-timeout *time*

(Optional) Specifies the time limit in seconds for this command to wait until the hosts are successfully configured after which the timeout occurs. If not specified, this command does not wait for the hosts to be successfully configured.

Default: 30 seconds.

Note: If you specify zero, the default system wide timeout occurs.

Example: Apply Policies to a Group of Hosts

This example applies the specified policy to a group of hosts.

```
ism-applyPolicies -policyName policy5 -policyType SystemEDGE -hostName "s1" -timeout 30
```

ism-applyPolicyWait Command--Apply a Policy and Wait (Funclet)

The `ism-applyPolicyWait` command applies a policy to a host and returns the host name on which the policy is applied. This command does not return the host name until the policy is applied or timeout occurs.

This command has the following format:

```
ism-applyPolicyWait  
-policyName policy_name  
-policyType policy_type  
-hostName host_name  
[-timeout time]
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

-hostName *host_name*

Specifies the name of the host computer.

-timeout *time*

(Optional) Specifies the time limit in seconds for this command to wait until the hosts are successfully configured after which the timeout occurs. If not specified, this command does not wait for the hosts to be successfully configured.

Default: 30 seconds.

Example: Apply a policy and Wait

Apply a policy, policy5 to the host, hvserver.

```
ism-applyPolicy -policyName policy5 -policyType SystemEDGE.SRM.Policies  
-hostname hvserver -timeout 400
```

ism-applyPolicyTemplates Command--Apply Policy Templates to a Group of Hosts (Funclet)

The ism-applyPolicyTemplates command applies the specified policy templates to a group of hosts.

This command has the following format:

```
ism-applyPolicyTemplates  
-templateName templateName  
-templateOs templateOs  
-hostName hostname  
[-timeout time]  
[-replaceExisting replaceExisting]
```

-templateName *templateName*

Specifies the name of the template.

-templateOs *templateOs*

Specifies the policy template operating system.

-hostName *host_name*

Specifies the name of the host computer.

-timeout *time*

(Optional) Specifies the time limit in seconds for this command to wait until the hosts are successfully configured after which the timeout occurs. If not specified, this command does not wait for the hosts to be successfully configured.

Default: 30 seconds.

Note: If you specify zero, the default system wide timeout occurs.

-replaceExisting *replaceExisting*

(Optional) Replaces the existing configuration of the host with the new configuration.

Example: Apply policy templates to a Group of Hosts

Apply policy templates to a group of hosts.

```
ism-applyPolicyTemplates -templateName template2 -templateOs windows -hostName  
"s1","s2","s3"-timeout 400 -replaceExisting yes
```

ism-applyPoliciesAndTemplates Command--Apply Policies and Templates to a Group of Hosts (Funclet)

The `ism-applyPoliciesAndTemplates` command applies both policies and templates to a group of hosts.

This command has the following format:

```
ism-applyPoliciesAndTemplates
-policyName policyName
-policyType policyType
-hostName hostname
-templateName templateName
-templateOs templateOs
[-timeout time]
[-replaceExisting replaceExisting]
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

-hostName *host_name*

Specifies the name of the host computer.

-templateName *templateName*

Specifies the name of the template.

-templateOs *templateOs*

Specifies the policy template operating system.

-timeout *time*

(Optional) Specifies the time limit in seconds for this command to wait until the hosts are successfully configured after which the timeout occurs. If not specified, this command does not wait for the hosts to be successfully configured.

Default: 30 seconds.

Note: If you specify zero, the default system wide timeout occurs.

-replaceExisting *replaceExisting*

(Optional) Replaces the existing configuration of the host with the new configuration.

Example: Apply policies and templates to a group of Hosts

Apply policies and templates to a group of hosts.

```
ism-applyPoliciesAndTemplates -policyName policy6 -policyType
SystemEDGE.SRM.Policies -hostName "s4","s5","s6" -templateName template2
-templateOs windows -timeout 400 -replaceExisting yes
```

ism-applyPoliciesForService Command--Apply Policies to all Systems (Funclet)

The ism-applyPoliciesForService command applies the policies to all computer systems in a managed service.

This command has the following format:

```
ism-applyPoliciesForService
-policyName policy_name
-policyType policy_type
-serviceName service_name
[-timeout time]
```

-policyName *policy_name*

Specifies the policy name to apply.

Default: Null

-policyType *policy_type*

Specifies the policy type.

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as "Enterprise\\Data Center\\servicename", instead of "Enterprise\Data Center\servicename".

-timeout *time*

(Optional) Specifies the time limit in seconds for this command to wait until the hosts are successfully configured after which the timeout occurs. If not specified, this command does not wait for the hosts to be successfully configured.

Default: 30 seconds.

Note: If you specify zero, the default system wide timeout occurs.

Example: Apply Policies to all Systems

This example applies the policies to the managed service, CA Virtual Assurance.

```
ism-applyPoliciesForService -policyName policy7 -policyType SystemEDGE -serviceName
"Enterprise\\Data Center\\CA Virtual Assurance Services" -timeout 400
```

ism-applyPolicyTemplatesForService Command--Apply Policy Templates to all Systems (Funclet)

The `ism-applyPolicyTemplatesForService` command applies a policy templates to all computer systems in a managed service.

This command has the following format:

```
ism-applyPolicyTemplatesForService
-templateName templateName
-templateOs templateOs
-serviceName serviceName
[-timeout time]
[-replaceExisting replaceExisting]
```

-templateName *templateName*

Specifies the name of the template.

-templateOs *templateOs*

Specifies the policy template operating system.

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as "Enterprise\\Data Center\\servicename", instead of "Enterprise\Data Center\servicename".

-timeout *time*

(Optional) Specifies the time limit in seconds for this command to wait until the hosts are successfully configured after which the timeout occurs. If not specified, this command does not wait for the hosts to be successfully configured.

Default: 30 seconds.

Note: If you specify zero, the default system wide timeout occurs.

-replaceExisting *replaceExisting*

(Optional) Replaces the existing configuration of the host with the new configuration.

Example: Apply Policy Templates to all Systems

This example applies policy templates, SystemEDGE to the managed service, CA Virtual Assurance.

```
ism-applyPolicyTemplatesForService -templateName template3 -templateOs windows  
-serviceName "Enterprise\\Data Center\\CA Virtual Assurance Services" -timeout 200  
-replaceExisting yes
```

ism-applyPoliciesAndTemplatesForService Command--Apply Policies and Templates to all Systems (Funclet)

The `ism-applyPoliciesAndTemplatesForService` command applies both policies and templates to all computer systems in a managed service.

This command has the following format:

```
ism-applyPoliciesAndTemplatesForService  
-policyName policy_name  
-policyType policy_type  
-serviceName service_name  
-templateName templateName  
-templateOs templateOs  
[-timeout time]  
[-replaceExisting replaceExisting]
```

-policyName *policy_name*

Specifies the policy name to apply.

Default: Null

-policyType *policy_type*

Specifies the policy type.

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as "Enterprise\\Data Center\\servicename", instead of "Enterprise\Data Center\servicename".

-templateName *templateName*

Specifies the name of the template.

-templateOs *templateOs*

Specifies the policy template operating system.

-timeout *time*

(Optional) Specifies the time limit in seconds for this command to wait until the hosts are successfully configured after which the timeout occurs. If not specified, this command does not wait for the hosts to be successfully configured.

Default: 30 seconds.

Note: If you specify zero, the default system wide timeout occurs.

-replaceExisting *replaceExisting*

(Optional) Replaces the existing configuration of the host with the new configuration.

Example: Apply Policies and Templates to all Systems

This example applies both policies and templates to the managed service, CA Virtual Assurance.

```
ism-applyPoliciesAndTemplatesForService -policyName policy5 -policyType SystemEDGE  
-serviceName "Enterprise\\Data Center\\CA Virtual Assurance Services" -templateName  
template2 -templateOs windows -timeout 400 -replaceExisting yes
```

ism-cancelJob Command--Cancel a Job (Funclet)

The `ism-cancelJob` command cancels any pending tasks in a job. You can specify either the job UUID or friendly ID.

Note: The command cannot cancel active tasks.

This command has the following format:

```
ism-cancelJob -jobID job_ID
```

-jobID *job_ID*

Specifies the job UUID or friendly ID.

Example: Cancel a Job

This example cancels all the pending tasks in the job, J00004.

```
ism-cancelJob -jobID J00004
```

ism-copyPolicy Command--Copy a Policy (Funclet)

The ism-copyPolicy command creates a policy by copying from an existing policy.

This command has the following format:

```
ism-copyPolicy -policyName policy_name  
-policyType policy_type  
-policyVersion [policy_version]  
-newName new_policy_name
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

-hostName *host_name*

Specifies the name of the host computer.

-policyVersion *policy_version*

(Optional) Specifies the version of the policy to apply.

Default: Null

-newName *new_policy_name*

Defines the name of the copied policy.

Example: Copy a Policy

The following example creates a policy, test1ver1 using the existing default policy.

```
ism-copyPolicy -policyName default -policyType SystemEDGE -policyVersion 1 -newName  
test1ver1
```

ism-createJobTemplate Command--Create a Job Template (Funclet)

The ism-createJobTemplate command creates a job template from an existing job. This command returns the template ID of the created job.

This command has the following format:

```
ism-createJobTemplate -name new_template_name  
-jobID existing_job_ID
```

-name *new_template_name*

Defines the name of the template.

-jobID *job_ID existing_job_ID*

Specifies the existing job UUID or friendly ID.

Example: Create a Job Template

The following example creates a job template, Job_template_08 from the existing job ID, J00004.

```
ism-createJobTemplate -name Job_template_08 -jobID J00004
```

ism-deleteJobTemplate Command--Delete a Job Template (Funclet)

The ism-deleteJobTemplate command deletes an existing job template. If you specify a duplicate name, the command returns an error.

This command has the following format:

```
ism-deletejobTemplate job_template_name
```

job_template_name

Specifies the template name to delete. You can also specify the template UUID.

Example: Delete a Job Template

This example deletes an existing job template, Job_template_08.

```
ism-deleteJobTemplate Job_template_08
```

ism-deletePolicy Command--Delete a Policy (Funclet)

The ism-deletePolicy command deletes an existing policy. If you do not specify the policy version, the latest version is selected.

This command has the following format:

```
ism-deletePolicy -policyName policy_name  
-policyType policy_type [-policyVersion policy_version]
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

-policyVersion *policy_version*

(Optional) Specifies the version of the policy to apply.

Default: Null

Example: Delete a Policy

This example deletes an existing policy, policy5.

```
ism-deletePolicy -policyName policy5 -policyType SystemEDGE.SRM.Policies  
-policyVersion 3.2
```

ism-deployPackageToHost Command--Deploy a Package to a Host (Funclet)

The `ism-deployPackageToHost` command deploys a specified package wrapper to a host. This command returns the UUID of the deployment job.

This command has the following format:

```
ism-deployPackageToHost  
[-jobName job_name]  
-package package_name  
[-packageVer package_version]  
[-wrapper wrapper_name]  
-hostName host_name  
-user user_name  
-password password  
-acceptEula  
[-encrypted]  
[-redeploy]  
[-ignoreDepends]
```

-jobName *job_name*

(Optional) Specifies the job name that is used in the deployment job list and audit trail.

-package *package_name*

Specifies the name or ID of the deployment package.

-packageVer *package_version*

(Optional) Specifies the deployment package version.

-wrapper *wrapper_name*

(Optional) Specifies the name package wrapper to use. If the wrapper is not specified, the default wrapper is used.

-hostName *host_name*

Specifies the name of the host computer.

-user *user_name*

Specifies the user name to log in.

-password *user_password*

Specifies the user password to log in.

-acceptEula

Indicates to accept all End User License Agreements (EULAs) for all deployed packages.

-encrypted

(Optional) Indicates that the username and password are encrypted.

-redeploy

(Optional) Indicates forced redeployment of existing packages.

-ignoreDepends

(Optional) Indicates forced deployment even when there are missing package dependencies.

Example: Deploy a Package to a Host

This example deploys the package, SystemEDGE on the host, hvserver.

```
ism-deployPackageHost
-package "CA SystemEDGE"
-hostName hvserver
-user admin
-password #test#
-acceptEula
-redeploy
-ignoreDepends
```

ism-deployTemplate Command--Deploy Packages from a Template (Funclet)

The `ism-deployTemplate` command deploys all packages defined in a template to all computers. This command returns the UUID of the deployment job.

This command has the following format:

```
ism-deployTemplate
[-jobName job_name]
-template template_name
-user user_name
-password password
-acceptEula
[-encrypted]
[-redeploy]
[-ignoreDepends]
```

-jobName *job_name*

(Optional) Specifies the job name that is used in the deployment job list and audit trail.

-template *template_name*

Specifies the name or UUID of the job template used as a source for the deployment. Use the template UUID if there are more than one template with the same name.

-user *user_name*

Specifies the user name to log in.

-password *user_password*

Specifies the user password to log in.

-hostName *host_name*

Specifies the name of the host computer.

-acceptEula

Indicates to accept all End User License Agreements (EULAs) for all deployed packages.

-encrypted

(Optional) Indicates that the username and password are encrypted.

-redeploy

(Optional) Indicates forced redeployment of existing packages.

-ignoreDepends

(Optional) Indicates forced deployment even when there are missing package dependencies.

Example: Deploy all Packages from a Template

This example deploys all packages defined in the template, SysTemplate to all computers.

```
ism-deployTemplate
-template SysTemplate
-user admin
-password #test#
-hostName hvserver
-acceptEula
-redeploy
-ignoreDepends
```


ism-deployTemplateForService Command--Deploy Template Packages in a Service (Funclet)

The `ism-deployTemplate` command deploys all packages defined in a job template to all computers listed in a service. This command returns the UUID of the deployment job.

This command has the following format:

```
ism-deployTemplateForService
[-jobName job_name]
-template template_name
-service service_name
-user user_name
-password password
-acceptEula
[-encrypted]
[-redeploy]
[-ignoreDepends]
```

-jobName *job_name*

(Optional) Specifies the job name that is used in the deployment job list and audit trail.

-template *template_name*

Specifies the name or UUID of the job template used as a source for the deployment. Use the template UUID if there are more than one template with the same name.

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as “Enterprise\\Data Center\\servicename”, instead of “Enterprise\Data Center\servicename”.

-user *user_name*

Specifies the user name to log in.

-password *user_password*

Specifies the user password to log in.

-acceptEula

Indicates to accept all End User License Agreements (EULAs) for all deployed packages.

-encrypted

(Optional) Indicates that the username and password are encrypted.

-redeploy

(Optional) Indicates forced redeployment of existing packages.

-ignoreDepends

(Optional) Indicates forced deployment even when there are missing package dependencies.

Example: Deploy Template Packages in a Service

This example deploys all packages defined in the template, SysTemplate in the service, Enterprise\Data Center\MyNewService.

```
ism-deployTemplateForService -template SysTemplate
-service Enterprise\Data Center\MyNewService
-user admin -password #test# -hostName hvserver
-acceptEula -redeploy -ignoreDepends
```

ism-deployTemplateToHost Command--Deploy Template Packages on a Host (Funclet)

The ism-deployTemplate command deploys all packages defined in a job template to a specified host. This command returns the UUID of the deployment job.

This command has the following format:

```
ism-deployTemplate
[-jobName job_name]
-template template_name
-hostname host_name
-user user_name
-password password
-acceptEula
-encrypted
[-redeploy]
[-ignoreDepends]
```

-jobName *job_name*

(Optional) Specifies the job name that is used in the deployment job list and audit trail.

-template *template_name*

Specifies the name or UUID of the job template used as a source for the deployment. Use the template UUID if there are more than one template with the same name.

-hostName *host_name*

Specifies the name of the host computer.

-user *user_name*

Specifies the user name to log in.

-password *user_password*

Specifies the user password to log in.

-hostName *host_name*

Specifies the name of the host computer.

-acceptEula

Indicates to accept all End User License Agreements (EULAs) for all deployed packages.

-encrypted

(Optional) Indicates that the username and password are encrypted.

-redeploy

(Optional) Indicates forced redeployment of existing packages.

-ignoreDepends

(Optional) Indicates forced deployment even when there are missing package dependencies.

Example: Deploy Template Packages on a Host

This example deploys all packages defined in the template, SysTemplate to the host, hvserver.

```
ism-deployTemplate -template SysTemplate
-hostname hvserver
-user admin
-password #test#
-hostName hvserver
-acceptEula
-redeploy
-ignoreDepends
```

ism-encryptString Command--Encrypt a String (Funclet)

The ism-encryptString command encrypts a given text and returns the encrypted text to the console.

This command has the following format:

```
ism-ism-encryptString encrypted_text
```

encrypted_text

Specifies the text to encrypt.

Example: Encrypt a String

This example encrypts the text string, passwordText.

```
ism-deleteJobTemplate passwordText
```

ism-listComputerSystems Command--Displays all Computer Systems (Funclet)

The ism-listComputerSystems command lists all computer systems including host names, platforms, managed services, and deployed packages.

This command has the following format:

```
ism-listComputerSystems [hostFilter]
```

hostFilter

(Optional) Specifies a partial host name. Use wildcards such as asterisk *.

Default: Null

Example: Display all Computer Systems

The following example displays all computer systems having h as the first character in the host name.

```
ism-listComputerSystems h*
```

ism-listDeployedSystems Command--List all Hosts With a Package Deployed (Funclet)

The ism-listDeployedSystems command lists all host names on which a specific package is deployed to.

Note: This view is an audit-trail view, so if hosts are removed or reimaged later, they still appear in this list.

This command has the following format:

```
ism-listDeployedSystems -package package_name  
-platform platform_name [-version package_version]
```

-package *package_name*

Specifies the name or ID of the deployment package.

-platform *platform_name*

Specifies the operating system of the package.

-version *package_version*

(Optional) Specifies the version of the package.

Note: Specify the version only if there are more than one package versions.

Example: List all Hosts Having a Package Deployed

This example lists all host names on which the package, 7e033e5d-test-404d-82in-a2f7bbe960a4 is deployed to.

```
ism-listDeployedSystems -package 7e033e5d-test-404d-82in-a2f7bbe960a4 -platform Windows
```

ism-listJobs Command--List all Jobs (Funclet)

The `ism-listJob` command lists all deployment jobs. You can filter by job status and partial operating environment name using wildcards.

This command has the following format:

```
ism-listJobs [-status job_status] [-platformFilter platform_filter]
```

-status *job_status*

(Optional) Indicates the status of the deployment job. You can specify a filter by a comma-separated list of status names: Active, Pending, Complete, or Failed.

-platformFilter *platform_filter*

(Optional) Indicates the full or partial operating environment name using wildcards.

Example: List all jobs with status Active

This example lists all jobs with the Active status.

```
ism-listJobs -status Active
```

ism-listJobStatus Command--List a Job Status (Funclet)

The `ism-listJobStatus` command displays the status of a deployment job.

This command has the following format:

```
ism-listJobStatus -jobID job_ID
```

-jobID *job_ID*

Specifies the job UUID or friendly ID.

Example: List a Job Status

This example displays the status of the job, aa-12348-test0-9883

```
ism-listJobStatus -jobId aa-12348-test0-9883
```

ism-listJobTemplates Command--List all Job Templates (Funclet)

The ism-listJobTemplates command lists UUIDs and names of all defined deployment job templates.

This command has the following format:

```
ism-listJobTemplates [-detail]
```

-detail

Show package and target details of each job template.

Example: List a Job Status

This example displays the deployment job templates.

```
ism-listJobTemplates
```

ism-listPackages Command--List all Packages (Funclet)

The ism-listPackages command lists all wrappers and packages.

Note: Packages that are referenced using an existing wrapper are not displayed.

This command has the following format:

```
ism-listJob [-platformFilter platform_filter]
```

-platformFilter *platform_filter*

(Optional) Indicates the full or partial operating environment name. Use wildcards such as asterisk *.

Example: List all Packages

This example lists all wrappers and packages with operating environment name starting with the letter w.

```
ism-listPackages -platformFilter w*.
```

ism-listPolicies Command--List all Policies (Funclet)

The ism-listPolicies command lists all policies. You can filter policies by policy type.

This command has the following format:

```
ism-listPolicies [-policyTypeFilter policytype_filter] [-allVersions]
```

-policyTypeFilter *policytype_filter*

(Optional) Indicates the policy type filter.

-allVersions

(Optional) Indicates to list all versions of all policies.

Example: List all Policies

This example lists all versions of all policies.

```
ism-listPolicies -allVersions
```

ism-listPolicyStatus Command--List a Policy Status (Funclet)

The `ism-listPolicyStatus` command lists a policy delivery status of all hosts.

This command has the following format:

```
ism-listPolicyStatus -policyName policy_name  
-policyType policy_type [-policyVersion policy_version]
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

-policyVersion *policy_version*

(Optional) Specifies the version of the policy to apply.

Default: Null

Example: List a Policy Status

This example lists the policy delivery status of the policy, `policy5`

```
ism-listPolicyStatus -policyName policy5  
-policyType SystemEDGE.SRM.Policies -policyVersion 3.2
```

ism-listServiceSystems Command--List all Computer Systems in Services (Funclet)

The `ism-listServiceSystems` command lists all computer systems in services matching a filter string. The list includes the host names, platforms, managed services, and deployed packages.

This command has the following format:

```
ism-listServiceSystems service_filter
```

service_filter

Specifies the full or partial service name. Use wildcards such as asterisk (*).

Example: List all Computer Systems in Services

This example lists all computer systems in all services starting with the character b.

```
ism-listServiceSystems b*
```

ism-listSysEdgeConfig Command--List the SystemEDGE Configuration (Funclet)

The listSysEdgeConfig command displays the SystemEDGE configuration that has been cached on the manager server for a given host name and a policy type.

This command has the following format:

```
ism-listSysEdgeConfig -hostName host_name -policyType policy_type
```

-hostName *host_name*

Specifies the name of the host computer.

-policyType *policy_type*

Specifies the policy type.

Example: List the SystemEDGE Configuration

This example lists the systemEDGE configuration.

```
ism-listSysEdgeConfig -hostname hvserver -policyType SystemEDGE.SRM.Policies
```

ism-listSystemPoliciesForService Command--List all Policies in a Service (Funclet)

The ism-listSystemPoliciesForService command lists all policies deployed to each hosts in a service.

This command has the following format:

```
ism-listSystemPoliciesForService -serviceName service_name
```

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as "Enterprise\\Data Center\\servicename", instead of "Enterprise\Data Center\servicename".

Example: List all Policies in a Service

This example lists all policies in the service, Enterprise\DataCenter\MyNewService.

```
ism-listServiceSystems Enterprise\DataCenter\MyNewService
```


ism-listTemplateEulas Command--Display EULAs in a Template (Funclet)

The `ism-listTemplateEulas` command displays the full text of all End-User License Agreements (EULAs) in a template. If there are more than one template names, use the template UUID.

This command has the following format:

```
ism-listTemplateEulas -template template_name
```

-template *template_name*

Specifies the template name. You can also specify the template UUID.

Example: Display all EULAs in a Template

This example displays all EULAs in the template, `test_template`.

```
ism-listTemplateEulas -template test_template
```

ism-renamePolicy Command--Rename a Policy (Funclet)

The `ism-renamePolicy` command renames the latest version of a policy.

Note: You cannot rename a previous policy version, create a copy instead.

This command has the following format:

```
ism-renamePolicy -policyName policy_name  
-policyType policy_type  
-newName new_policy_name
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

-newName *new_policy_name*

Defines the new policy name.

Example: Rename a Policy

This example renames a policy, `policy5` to a new name, `NewPolicy5`.

```
ism-applyPolicy -policyName policy5  
-policyType SystemEDGE.SRM.Policies -newName NewPolicy5
```

ism-setMaintenanceMode Command--Set the Maintenance Mode of a Host (Funclet)

The ism-setMaintenanceMode command sets the maintenance mode of a host.

This command has the following format:

```
ism-setMaintenanceMode -hostname host_name
[-on {true|false}]
```

-hostName *host_name*

Specifies the name of the host computer.

-on {true|false}

(Optional) Defines whether the host is set to the maintenance mode. Specify false to set the system out of the managed mode.

Default: true

Example: Set the Host Maintenance Mode

This example sets the host, hvserver to the maintenance mode.

```
ism-setMaintenanceMode -hostname hvserver
```

ism-listSystemPolicies Command--List all Policies (Funclet)

The ism-listSystemPolicies command lists all policies and policy templates deployed to each host.

This command has the following format:

```
ism-listSystemPolicies -hostName hostName
```

-hostName *hostName*

Specifies the name of the host.

Example: List all Policies in a Service

This example lists all policies and policy templates deployed to the given host, test2, along with the delivery status. The time when policies and policy templates were last applied and exceptions, if any.

```
ism-listSystemPolicies -hostName test2
```

Output

```
Policy                Status                OS Type Ver   Applied          Exc
-----
-- test2 -----
-- Policies-----
default SystemEDGE.SR Delivery requested Windows 0 2011-08-02 18:35:36 No
default SystemEDGE    Configured            Windows 0 2011-08-08 12:54:59 No
-- Templates -----
test3 SystemEDGE      Configured            Windows 0 2011-08-08 12:54:59 No
```

ism-setMaintenanceModeForService Command--Set the Maintenance Mode in a Service (Funclet)

The `ism-setMaintenanceModeForService` command sets the maintenance mode of all computer systems in a service.

This command has the following format:

```
ism-setMaintenanceModeForService
-servicename service_name
[-on {true|false}]
```

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as "Enterprise\\Data Center\\servicename", instead of "Enterprise\Data Center\servicename".

-on {true|false}

Defines whether all computer systems in the service are set to the maintenance mode. Specify false to set these systems out of the managed mode.

Default: true

Example: Set the Maintenance Mode in a Service

This example sets all hosts in the service, Enterprise\DataCenter\MyNewService to the maintenance mode.

```
ism-setMaintenanceModeForService -servicename Enterprise\DataCenter\MyNewService  
-on true
```

ism-setSysEdgeLogLevel Command--Set the SystemEDGE Log Level (Funclet)

The `ism-setSysEdgeLogLevel` command sets the SystemEDGE log level for a host.

This command has the following format:

```
ism-setSysEdgeLogLevel -hostname host_name -level level
```

-hostName *host_name*

Specifies the name of the host computer.

-level *level*

Indicates the agent to log messages up to a given level. The levels are: fatal, critical, warning, info, debug, debug1, debug2, and debug3. The higher the value, the more information is logged. Fatal is the lowest level, and debug3 is the highest level.

Example: Set the SystemEDGE Log Level

This example sets the System Edge Log level to fatal.

```
ism-setSysEdgeLogLevel -hostname hvserver -level fatal
```

ism-setSysEdgeLogLevelForService Command--Set the SystemEDGE Log Level in a Service (Funclet)

The `ism-setSysEdgeLogLevelForService` command sets the SystemEDGE log level for all hosts in a service.

This command has the following format:

```
ism-setSysEdgeLogLevelForService -servicename service_name -level level
```

-serviceName *service_name*

Specifies the name of the managed service.

Note: In the service name, use double backslash as a delimiter. Using single backslash as a delimiter is interpreted as an escape sequence. For example, specify service name as "Enterprise\\Data Center\\servicename", instead of "Enterprise\Data Center\servicename".

-level *level*

Indicates the agent to log messages up to a given level. The levels are: fatal, critical, warning, info, debug, debug1, debug2, and debug3. The higher the value, the more information is logged. Fatal is the lowest level, and debug3 is the highest level.

Example: Set the SystemEDGE Log Level in a Service

This example sets the SystemEDGE Log level in the service, Enterprise\DataCenter\MyNewService.

```
ism-setSysEdgeLogLevel -servicename Enterprise\DataCenter\MyNewService -level info
```

ism-updateAllExceptionSystems Command--Apply the Latest Policy (Funclet)

The `ism-updateAllExceptionSystems` command finds all hosts with policy exceptions and applies the latest policy. The command returns OK on success or displays an error message.

This command has the following format:

```
ism-updateAllExceptionSystems
```

Example: Apply the Latest Policy

This example finds all hosts with policy exceptions and applies the latest policy.

```
ism-updateAllExceptionSystems
```

ism-updateAllOldPolicies Command--Update all Old Policies (Funclet)

The `ism-updateAllOldPolicies` command finds all hosts with out-of-date policy configurations and applies the latest policy.

This command has the following format:

```
ism-updateAllOldPolicies
```

Example: Update all Old Policies

This example finds all hosts with old policies and applies the latest policies.

```
ism-updateAllOldPolicies
```

ism-updateAllPolicies Command--Update all Policies (Funclet)

The `ism-updateAllPolicies` command applies the latest policies to all computers.

This command has the following format:

```
ism-updateAllPolicies
```

Example: Update all Policies

This example applies the latest policies to all computers.

```
ism-updateAllPolicies
```

ism-updateAllUpToDateSystems Command--Update all Up-To-Date Systems (Funclet)

The `ism-updateAllUpToDateSystems` command reapplies the latest policies to all hosts that are up-to-date.

This command has the following format:

```
ism-updateAllUpToDateSystems
```

Example: Update all Up-To-Date Systems

This example reapplies the latest policies to all computers that are up-to-date.

```
ism-updateAllUpToDateSystems
```

ism-updateExceptionSystems Command--Apply the Latest Policy (Funclet)

The `ism-updateExceptionSystems` command finds all hosts with policy exceptions for a policy and applies the latest configuration.

This command has the following format:

```
ism-updateExceptionSystems -policyName policy_name  
-policyType policy_type
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

Example: Update to the Latest Policy

This example updates all policy exceptions for the default policy and applies the latest configuration.

```
ism-updateExceptionSystems -policyName default -policyType SystemEDGE
```

ism-updateOldSystems Command--Update an old Policy (Funclet)

The `ism-updateOldSystems` command finds all hosts that have out-of-date configurations for a specified policy and applies the latest policy configuration.

This command has the following format:

```
ism-updateOldSystems -policyName policy_name  
-policyType policy_type
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

Example: Update an Old Policy

This example updates all hosts that have out-of-date configurations for the default policy and applies the latest configuration.

```
ism-updateOldSystems -policyName default -policyType SystemEDGE
```

ism-updatePolicies Command--Update Configuration of a Policy (Funclet)

The `ism-updateSystems` command applies the latest configuration to all hosts running the specified policy.

This command has the following format:

```
ism-updateSystems -policyName policy_name  
-policyType policy_type
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

Example: Update Configuration of a Policy

This example applies the latest configuration to all hosts running the default policy.

```
ism-updateSystems -policyName default -policyType SystemEDGE
```

ism-updateUpToDateSystems Command--Reapply a Policy (Funclet)

The `ism-updateUpToDateSystems` command reapplies the latest configuration of the specified policy to all hosts that are considered up-to-date for that policy.

This command has the following format:

```
ism-updateUpToDateSystems -policyName policy_name  
-policyType policy_type
```

-policyName *policy_name*

Specifies the policy name to apply.

-policyType *policy_type*

Specifies the policy type.

Example: Reapply a Policy

This example reapplies the latest configuration to all hosts running the default policy.

```
ism-updateUpToDateSystems -policyName default -policyType SystemEDGE
```

ism-copyPolicyTemplate Command--Copy a Policy Template (Funclet)

The `ism-copyPolicyTemplate` command creates a policy template by copying from an existing policy template.

This command has the following format:

```
ism-copyPolicyTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs  
-newName newName
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

-newName *newName*

Defines the new name of the copied policy template.

Example: Copy a Policy Template

This example creates a new copy of the selected policy template, test 1. The copy is created with the new name test 2.

```
ism-copyPolicyTemplate -templateName test1 -templateType SystemEDGE -templateOs windows -newName test2
```

ism-deletePolicyTemplate Command--Delete a Policy Template (Funclet)

The `ism-deletePolicyTemplate` command deletes an existing policy template.

This command has the following format:

```
ism-copyPolicyTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

Example: Delete a Policy Template

This example deletes an existing policy template.

```
ism-deletePolicyTemplate -templateName test3 -templateType SystemEDGE -templateOs windows
```

ism-listPolicyTemplateStatus Command--List a Policy Template Status (Funclet)

The `ism-listPolicyTemplateStatus` command lists a policy template delivery status of all computer systems using the named policy template.

This command has the following format:

```
ism-copyPolicyTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

Example: List a Policy Template Status

This example lists the policy template delivery status of all computer systems.

```
ism-listPolicyTemplateStatus -templateName Test1 -templateType SystemEDGE  
-templateOs windows
```

ism-removePolicyTemplate Command--Remove a Policy Template (Funclet)

The `ism-removePolicyTemplate` command removes a policy template from the list of policies associated with the given host.

This command has the following format:

```
ism-removePolicyTemplate -templateName templateName  
-templateType templateType  
-hostName hostName
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-hostName *hostName*

Specifies the name of the host computer.

Example: Remove a Policy Template

This example removes a policy template from the list of policies associated with the host name specified.

```
ism-removePolicyTemplate -templateName templateName -templateType templateType  
-hostName hostName
```

ism-renamePolicyTemplate Command--Rename a Policy Template (Funclet)

The ism-renamePolicyTemplate command renames a policy template.

This command has the following format:

```
ism-renamePolicyTemplate -templateName templateName
-templateType templateType
-templateOs templateOs
-newName newName
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

-newName *newName*

Specifies the new name of the policy template renamed.

Example: Rename a Policy Template

This example renames the selected policy template.

```
ism-renamePolicyTemplate -templateName test2 -templateType SystemEDGE -templateOs
windows -newName test3
```

ism-updateExceptionSystemsTemplate Command--Apply the Latest Policy Template (Funclet)

The ism-updateExceptionSystemsTemplate command finds all hosts with policy exceptions for a policy template and applies the latest configuration.

This command has the following format:

```
ism-updateExceptionSystemsTemplate -templateName templateName
-templateType templateType
-templateOs templateOs
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

Example: Update to the Latest Policy Template

This example updates all policy exceptions for the policy template and applies the latest configuration.

```
ism-updateExceptionSystemsTemplate -templateName tpl1 -templateType SystemEDGE  
-templateOs windows
```

ism-updateOldSystemsTemplate Command--Update all Old System Templates (Funclet)

The `ism-updateOldSystemsTemplate` command finds all hosts with out-of-date configurations for all system templates and applies the latest configuration.

This command has the following format:

```
ism-updateOldSystemsTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

Example: Update all Old System Templates

This example finds all hosts with old configurations for all system templates and apply the latest policies.

```
ism-updateOldSystemsTemplate -templateName tpl1 -templateType SystemEDGE  
-templateOs windows
```

ism-updateSystemsTemplate Command--Apply a System Template (Funclet)

The `ism-updateSystemsTemplate` command applies the latest configuration of the specified system template to all hosts.

This command has the following format:

```
ism-updateSystemsTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

Example: Apply a System Template

This example applies the latest configuration to all hosts running the system template.

```
ism-updateSystemsTemplate -templateName tpl1 -templateType SystemEDGE -templateOs windows
```

ism-updateUpToDateSystemsTemplate Command--Reapply a System Template (Funclet)

The `ism-updateUpToDateSystemsTemplate` command reapplies the latest configuration of the specified system template to all hosts that are considered up-to-date for that system template.

This command has the following format:

```
ism-updateUpToDateSystemsTemplate --templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

Specifies the name of the template.

-templateType *templateType*

Specifies the policy template type.

-templateOs *templateOs*

Specifies the policy template operating system.

Example: Reapply a System Template

This example reapplies the latest configuration to all hosts running the system template.

```
ism-updateUpToDateSystemsTemplate -templateName tpl1 -templateType SystemEDGE  
-templateOs windows
```

ism-listPolicyTemplate Command--List all Policy Templates (Funclet)

The `ism-listPolicyTemplate` command lists all policy templates of all computer systems.

This command has the following format:

```
ism-listPolicyTemplate -policyTypeFilter policyTypeFilter
```

-policyTypeFilter policyTypeFilter

Specifies the filter type to sort and display all policy templates.

Default: Null

Example: List all Policy Templates

This example lists all policy templates of all computer systems.

```
ism-listPolicyTemplate -policyTypeFilter policyTypeFilter
```

Chapter 3: Command Line Scripting

This section contains the following topics:

- [Command-Line Instructions \(CLIs\)](#) (see page 383)
- [Amazon Elastic Compute Cloud Commands](#) (see page 384)
- [CA Cisco UCS Commands](#) (see page 395)
- [CA Hyper-V CLI Commands](#) (see page 436)
- [CA IBM LPAR CLI Commands](#) (see page 508)
- [CA Microsoft Cluster Server CLI Commands](#) (see page 547)
- [CA Solaris Zones CLI Commands](#) (see page 567)
- [CA VMware vCenter Server CLI Commands](#) (see page 596)
- [General Shell Commands](#) (see page 685)
- [Verification Commands](#) (see page 687)
- [Discovery Commands](#) (see page 733)
- [Collection Engine Commands](#) (see page 738)
- [Event Commands](#) (see page 758)
- [Help Desk Commands](#) (see page 762)
- [Solaris Imaging Commands](#) (see page 769)
- [Network Installation Management Commands](#) (see page 777)
- [Object Model Utility Commands](#) (see page 795)
- [Policy Commands](#) (see page 803)
- [Policy Configuration](#) (see page 818)
- [CA Process Automation Commands](#) (see page 821)
- [Remote Monitoring Commands](#) (see page 826)
- [Reservation Manager Commands](#) (see page 835)
- [Resource Manager Commands](#) (see page 847)
- [Service Response Monitor CLI Commands](#) (see page 894)
- [Software Delivery Commands](#) (see page 1016)
- [Storage Provisioning Manager Command](#) (see page 1042)
- [Log Files](#) (see page 1055)
- [Log File Settings](#) (see page 1056)

Command-Line Instructions (CLIs)

Command line instructions (CLIs) are available for scripting and automating tasks. You can use these commands to write scripts for anything that you can do in the CA Server Automation user interface. CLI commands generate their own log files.

Important: Verify that you have sufficient privileges to run CLI commands from a Command Prompt.

Amazon Elastic Compute Cloud Commands

You can use the `dpmec2` CLI to script and automate Amazon Elastic Compute Cloud (EC2) commands and run actions based on the command results.

More information:

[dpmec2 createImage--Create AMI Image](#) (see page 384)
[dpmec2 ec2jobcheck Command--Get Amazon EC2 Job Status](#) (see page 385)
[dpmec2 getimages Command--Get Amazon EC2 Images](#) (see page 387)
[dpmec2 getinstances Command--Get Amazon EC2 Instances](#) (see page 388)
[dpmec2 getlist Command--Get Amazon EC2 List](#) (see page 389)
[dpmec2 imgjobcheck Command--Get Amazon EC2 Imaging Job Status](#) (see page 390)
[dpmec2 manage Command--Manage Amazon Elastic Compute Cloud Instances](#) (see page 392)
[dpmec2 run Command--Create Amazon EC2 Instance](#) (see page 393)

dpmec2 createImage--Create AMI Image

The `dpmec2 createImage` command creates an image from a running or stopped Elastic Block Storage (EBS) instance.

This command has the following format:

```
dpmec2 createImage [-sc sc_url] --instance instanceID --imageName "imagename"  
[-imageDesc description] [-noReboot true|false] [-region region] [-pre]  
[-post][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-instance *instanceID*

Defines the instance ID.

-imageName "imagename"

Defines the image name. Valid entry: 3-128 alphanumeric characters including parenthesis, commas, slashes, dashes, and underscores. Must be enclosed in double quotes.

-imageDesc description

Defines a description of the image. Valid entry: any alphanumeric character.

-noReboot true|false

Defines whether the instance is rebooted after it is created. Valid entries: true = reboot, false = do not reboot (default).

-region region

Defines any of the regions supported and defined by Amazon.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create an EBS Instance

This example creates an EBS instance that is not rebooted after it is created.

```
dpmec2 createImage --image ami-1234abcd -noReboot false
```

dpmec2 ec2jobcheck Command--Get Amazon EC2 Job Status

The `dpmec2 ec2jobcheck` command retrieves the job status for a specific instance.

This command has the following format:

```
dpmec2 ec2jobcheck [-sc sc_url] [--instance instanceID --operation
reboot|run|terminate|start|stop|createImage | -image imageID --operation
createImage} [-region region] [-pre] [-post][--locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-instance *instanceID* -operation `reboot|run|terminate|start|stop` |

-image *imageID* -operation `createImage`

Specifies the operation for retrieving job status for an instance or image.

-instance *instanceID*

Defines the instance ID.

-operation `reboot|run|terminate|start|stop`

Defines the operation for which you want to check the instance job status.

Valid entries: `reboot`, `run`, `terminate`, `start`, or `stop`.

-image *imageID*

Defines the image ID.

-operation `createImage`

-region *region*

Defines any of the regions supported and defined by Amazon.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Job Status for Instance Reboot

This example retrieves the job status of the reboot of instance i-1234abcd.

```
dpmec2 ec2jobcheck --instance i-1234abcd --operation reboot
```

dpmec2 getimages Command--Get Amazon EC2 Images

The `dpmec2 getimages` command lists Amazon EC2 images.

This command has the following format:

```
dpmec2 getimages [-sc sc_url] {-image all|imageID[,imageID] | -owner
ownerID[,ownerID]} [-region region] [-pre] [-post][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-image all|*imageID*[,*imageID*] |**-owner *ownerID* [,*ownerID*]**

Defines the list of all images, specific images, or images by owners.

all

Lists all images for a specified region. If `-region` is not specified, the default region is used.

***imageID*[,*imageID*]**

Lists specific images.

-owner *ownerID*[,*ownerID*]

Lists all images for a specified region. If `-region` is not specified, the default region is used.

-region *region*

Lists AMIs for the default or specified region. If you do not specify this option, the default region (set during or after installation in the UI or CLI) is used.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get a Specific Image

This example lists details for image AMI-40fb5892.

```
dpmec2 getimages --image ami-40fb5892
```

dpmec2 getinstances Command--Get Amazon EC2 Instances

The dpmec2 getinstances command lists Amazon EC2 reservations.

This command has the following format:

```
dpmec2 getinstances [-sc sc_url] -reservation all|reservationID[,reservationID]  
[-region region] [-pre] [-post][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-reservation all|reservationID[,reservationID]

Defines all instance reservations or specific reservations.

all

Lists all instances.

reservationID

Lists the instances with specific reservation IDs

-region region

Defines any of the regions supported and defined by Amazon.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get All Instances

This example lists reservations for all instances.

```
dpmec2 getinstances -reservation all
```

dpmec2 getlist Command--Get Amazon EC2 List

The dpmec2 getlist command lists various Amazon EC2 objects.

This command has the following format:

```
dpmec2 getlist [-sc sc_url] -display zones|keypairs|security|types|vpcs|subnets|
customer_gateways|vpn_gateways|vpn_connections -image imageID [-region
region][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-display *zones* | *keypairs* | *security* | *types* | *vpcs* | *subnets* | *customer_gateways* | *vpn_gateways* | *vpm_connections*

Defines the objects to list.

-image *imageID*

Defines the instance types to list for the AMI.

-region *region*

Defines any of the regions supported and defined by Amazon.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: List Amazon EC2 objects

This example lists the VPC subnets.

```
dpmec2 getlist -display subnets
```

dpmec2 imgjobcheck Command--Get Amazon EC2 Imaging Job Status

The `dpmec2 imgjobcheck` command lists the Amazon EC2 job status for a specific job ID.

This command has the following format:

```
dpmec2 imgjobcheck [-sc sc_url] -status jobID [-region region][-pre] [-post][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-status *jobID*

Defines the job ID.

-region *region*

Defines any of the regions supported and defined by Amazon.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Image Status Using Job ID

This example lists the job status of the Amazon EC2 run action.

```
dpmec2 imgjobcheck -status 42
```

dpmec2 manage Command--Manage Amazon Elastic Compute Cloud Instances

The `dpmec2 manage` command reboots or terminates Amazon EC2 instances.

Important! Back up all important data before you issue this command to avoid permanently losing unsaved data.

This command has the following format:

```
dpmec2 manage -instance all|instanceID[,instanceID]} [-sc sc_url] -operation {reboot|terminate|password} [-wait [timeout]] [-region region] [-pre] [-post] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-instance all|instanceID[,instanceID]

Specifies whether all instances or specific instances are managed.

all

Lists all instances.

instanceID[,instanceID]

Lists the specific instances.

-operation reboot|terminate|password

Specifies the action to perform on the specified list of instances.

reboot

Reboots the list of instances.

terminate

Permanently deletes the list of instances.

password

The Windows instance password.

-wait [timeout]

Defines a time in minutes to wait for an operation to complete or fail. Valid entries: 0=wait indefinitely. If you do not specify this option, an operation updates without waiting for completion. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `casdaconf.cfg` (dpmsd only) or `caimgconf.cfg` (all other CLIs) files, or defaults to 120 minutes.

-region *region*

Defines any of the regions supported and defined by Amazon.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Reboot All Instances

This example reboots all instances.

```
dpmec2 manage --instance all --operation reboot
```

dpmec2 run Command--Create Amazon EC2 Instance

The `dpmec2 run` command creates an Amazon EC2 instance.

Important! Amazon EC2 lets you create an instance of a public image specifying a key pair. If the key is not provided, the instances are inaccessible.

This command has the following format:

```
dpmec2 run [-sc sc_url] --image imageID [-group groupID[,groupID]] [-key_pair keypair]
[-os_type {linux|windows|other}] [-type instancesize] [-user_data string] [-zone
placement] [--wait [timeout]] -subnet subnet [-pre] [-post]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-image *imageID*

Defines the image from which to create an instance.

-group *groupID*

Defines the security group to which the instance belongs.

-key_pair *keypair*

Defines the key pair the instance uses.

-os_type *linux | windows | other*

Defines the operating system that runs on the instance.

-type *instancesize*

Defines the instance size. Valid entries: `micro | small | large | xl | high_CPU_med | high_CPU_xl | high_Mem_xl | high_Mem_x2 | high_Mem_x4 | Cluster | default | small`

-user_data *string*

Defines the data available to the instance. The user data must be quoted if it contains any spaces.

-zone *placement*

Defines the Availability zone where the instance is run.

-wait [*timeout*]

Specifies that the CLI displays the job status and does not return until the operation completes, fails, or the timeout period is met (if timeout is specified). If you do not specify the `-wait` option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you specify the `-wait` option with no timeout value, the CLI uses the default wait time from the `sdadapter.conf` (`dpmsd` only) or `imaging.conf` (all other CLIs) files, or defaults to 120 minutes. In addition to any positive integer, you can specify zero (0) to wait indefinitely.

-subnet *subnet*

Defines the subnet.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Create and Instance with User Data

This example creates an instance and passes user data and waits up to 20 minutes for it to finish.

```
dpmec2 run --image ami-54bc27fg --key_pair pr_kp --user_data "Payroll System" --wait 20
```

CA Cisco UCS Commands

You can use `dpmucs` CLI commands to retrieve Cisco Unified Computing System (UCS) device information, statistics, service profiles, pools, and execute blade and chassis operations.

More Information

- [dpmucs gettopsystem Command--Get System Information](#) (see page 428)
- [dpmucs getchassis Command--Get Chassis Information](#) (see page 423)
- [dpmucs getblades Command--Get Blade Information](#) (see page 422)
- [dpmucs getswitches Command--Get Switch Information](#) (see page 427)
- [dpmucs powercycle Command--Power Cycle Blades](#) (see page 431)
- [dpmucs createmacpool Command--Create MAC Address Pool](#) (see page 399)
- [dpmucs createwwpool Command--Create WWNN Pool](#) (see page 408)
- [dpmucs deleteuuidpool Command--Delete UUID Pool](#) (see page 415)
- [dpmucs deletemacpool Command--Delete MAC Address Pool](#) (see page 410)
- [dpmucs deletewwpool Command--Delete WWNN Pool](#) (see page 416)
- [dpmucs listserviceprofiles Command--List Service Profiles](#) (see page 429)
- [dpmucs associateserviceprofile--Associate Service Profile With a Blade](#) (see page 397)
- [dpmucs disassociateserviceprofile Command--Disassociate Service Profile](#) (see page 419)
- [dpmucs getserviceprofilestatus Command--Get Service Profile Status](#) (see page 426)
- [dpmucs failoverserviceprofile Command--Associate a Service Profile With a Blade for Failover](#) (see page 421)
- [dpmucs createorg Command--Create Organization](#) (see page 401)
- [dpmucs deleteorg Command--Delete Organization](#) (see page 411)
- [dpmucs deleteserverpool Command--Delete Server Pool](#) (see page 413)
- [dpmucs serviceprofileupdate Command--Execute Service Profile Operations](#) (see page 434)
- [dpmucs createserviceprofile Command--Create a Service Profile](#) (see page 404)
- [dpmucs cloneserviceprofile--Clone a Service Profile](#) (see page 398)
- [dpmucs createtemplatefromserviceprofile Command--Create a Template from a Service Profile](#) (see page 405)
- [dpmucs deleteserviceprofile--Delete a Service Profile](#) (see page 414)
- [dpmucs getpolicy Command--Gets Policy from a UCS Manager](#) (see page 425)
- [dpmucs createuuidpool Command--Create UUID Pool](#) (see page 407)
- [dpmucs createserverpool Command--Create Server Pool](#) (see page 402)
- [dpmucs renamepool Command--Rename Pool](#) (see page 433)

dpmucs associateserviceprofile--Associate Service Profile With a Blade

Use this command to associate a service profile with a specific blade. This command shows the status of the blade before the association. To verify the association, use the `getserviceprofilestatus` command.

This command has the following format:

```
dpmucs associateserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagename
-service_profile_dn serviceprofiledn
-blade_dn bladename
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagename*

Defines the name of the UCS Manager for retrieving information or an operation.

-service_profile_dn *serviceprofiledn*

The fully qualified and unique name of a blade service profile. The blade service profile must be created before provisioning an OS. A service profile contains boot order, network settings, and storage (local or remote). Valid format: `org-root/serviceprofile`.

-blade_dn *bladename*

The fully qualified and unique name of a blade. If a blade is not provided, an unassociated, available blade is selected. Valid format: `systemid/chassisid/bladeid`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Associate a Profile With a Blade

This example associates a profile with the specified blade.

```
dpmucs associateserviceprofile -ucs_manager EngManager200
-service_profile_dn org-root/ls-dev-boot-from-san-esx-2
-blade_dn sys/chassis-1/blade-1
```

dpmucs cloneserviceprofile--Clone a Service Profile

Use this command to clone a service profile.

This command has the following format:

```
dpmucs cloneserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagename
-service_profile_dn serviceprofiledn
-cloneName clonename
[-org_path org_path]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-ucs_manager *ucsmanagename*

Defines the name of the UCS Manager for retrieving information or an operation.

-service_profile_dn *serviceprofiledn*

The fully qualified and unique name of a blade service profile. The blade service profile must be created before provisioning an OS. A service profile contains boot order, network settings, and storage (local or remote). Valid format: *org-root/serviceprofile*.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using *caaipsecurity*.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, *fr_FR* for French. To use the locale of the command prompt, specify "native".

Example: Clone a Service Profile

This example clones a service profile.

```
dpmucs cloneserviceprofile -ucs_manager EngManager200
-service_profile_dn org-root/ls-dev-boot-from-san-esx-2
-cloneName org-root/ls-dev-boot-from-san-esx-2
```

dpmucs createmacpool Command--Create MAC Address Pool

Use this command to create a MAC address pool with one or multiple slot ranges. The pool can be created at any organization level.

This command has the following format:

```
dpmucs createmacpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-description pooldescription
-org_path organizationpath
-ranges fromrange|torange fromrange|torange...
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager ucsmanagername

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_name poolname

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-description pooldescription

Defines the pool being created.

-org_path organizationpath

Defines the fully qualified path name to the organization. Valid format:
`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-ranges fromrange | torange fromrange | torange...

Defines the slot ranges for pools. Valid entries per pool type: MAC addresses in the format: `XX:XX:XX:XX:XX:XX`, UUID in the format: `XXXX-XXXXXXXXXXXX`, and WWNN/WWPN in the format: `XX:XX:XX:XX:XX:XX:XX:XX`. Delimit from-to range entries with `|`, and slot ranges with a single space.

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Create a MAC Address Pool

This example creates a MAC address pool with multiple pool slots.

```
dpmucs createmacpool
-ucs_manager ucsmanager
-pool_name TEST_MAC_POOL
-description TEST POOL -org_path ""
-ranges 00:25:B5:00:00:C0|00:25:B5:00:00:C3 00:25:B5:00:00:C4|00:25:B5:00:00:C9
00:25:B5:00:00:CA|00:25:B5:00:00:CF
-ws_user ba
-ws_password ba
```

dpmucs createorg Command--Create Organization

Use this command to create an organization.

This command has the following format:

```
dpmucs createorg [-sc sc_url]
-ucs_manager ucsmanagername
-parentOrg_dn parentorganization
-org_name organizationname
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-parentOrg_dn *parentorganization*

Defines the fully qualified name of the parent organization of the current component. Valid format: `parentorganization/org`.

-org_name *organizationname*

Defines the fully qualified name of the organization. Valid format:
org-root/org-name

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Create an Organization

This example creates a parent and subordinate organization.

```
dpmucs createorg
-ucs_manager EngManager200
-parentOrg_dn org-root
-org_name TEST_ORG
```

dpmucs createserverpool Command--Create Server Pool

Use this command to create a server pool.

This command has the following format:

```
dpmucs createserverpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-description pooldescription
-org_path organizationpath
-chassisbladepairs pairs
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager ucsmanagername

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_name poolname

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-description pooldescription

Defines the pool being created.

-org_path organizationpath

Defines the fully qualified path name to the organization. Valid format:
`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-chassisbladepairs pairs

Defines chassis-blade pairs for a server pool. Valid entries: chassis name, followed by blade name, separated by bar (|); separate pairs with a single space. Example:
`chassis1|blade1 chassis2|blade2.`

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Create a Server Pool

This example creates a server pool with multiple chassis blade pairs.

```
createserverpool
-ucs_manager ucsmanager
-pool_name TEST_SERVER_POOL
-description TEST POOL
-org_path ""
-chassisbladepairs Eng1|Blade2 Test1|Blade1 QA1|Blade3
```

dpmucs createserviceprofile Command--Create a Service Profile

Use this command to create one or more simple service profiles that are based on a hardware default, with the option to create a default vNIC or vHBA. The service profiles also can be created based on a template.

This command has the following format:

```
dpmucs createserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagername
-service_profile_name serviceprofilename
[-org_path org_path -description description]
-policyType profiletype
[-templatename templatename]
-number number
-creativnic creativnic
-creativhba creativhba]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-service_profile_name *serviceprofilename*

Specifies name for new service profile.

-templatename *templatename*

Specifies the template name.

-number *number*

Specifies the number of objects created.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a Service Profile

This example creates a service profile for the specified UCS Manager.

```
dpmucs.exe createserviceprofile
-ucs_manager Engmanager
-service_profile_name sptest_1
```

dpmucs createtemplatefromserviceprofile Command--Create a Template from a Service Profile

Use this command to create a template from an existing service profile.

This command has the following format:

```
dpmucs createtemplatefromserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagename
-service_profile_dn serviceprofiledn
-templatename templatename
[-templatetype templatetype -org_path org_path]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager ucsmanagername

Defines the name of the UCS Manager for retrieving information or an operation.

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Create Template from Service Profile

This example creates a template from a service profile for the specified UCS Manager.

```
dpmucs.exe createtemplatefromserviceprofile
-ucs_manager Engmanager
-service_profile_dn sptest_1
-templatename template_sp_1
```

dpmucs createuuidpool Command--Create UUID Pool

Use this command to create a pool of Universal Unique Identifiers (UUIDs) for blades with one or more slot ranges.

This command has the following format:

```
dpmucs createuuidpool
[-sc sc_url]
-ucs_manager ucsmanagename
-pool_name poolname
-description pooldescription
-org_path organizationpath
-ranges fromrange|torange fromrange|torange...
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagename*

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_name *poolname*

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-description *pooldescription*

Defines the pool being created.

-org_path *organizationpath*

Defines the fully qualified path name to the organization. Valid format:

Org-root/org-suborg1/org-suborg2/org-suborg3...

-ranges *fromrange|torange fromrange|torange...*

Defines the slot ranges for pools. Valid entries per pool type: MAC addresses in the format: XX:XX:XX:XX:XX:XX, UUID in the format: XXXX-XXXXXXXXXXXX, and WWNN/WWPN in the format: XX:XX:XX:XX:XX:XX:XX:XX. Delimit from-to range entries with |, and slot ranges with a single space.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a UUID Pool

This example creates a UUID pool with multiple pool slots.

```
dpmucs createuuidpool
-ucs_manager EngManager200
-pool_name TEST_POOL_UUID
-description TEST POOL
-org_path ""
-ranges 5500-000000000030|5500-000000000032 5500-000000000035|5500-000000000040
5500-000000000042|5500-000000000045
-ws_user cba
-ws_password cba
```

dpmucs createwwpool Command--Create WWNN Pool

Use this command to create a World Wide Note Name (WWNN) port or node pool for blades.

This command has the following format:

```
dpmucs createwwpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-description pooldescription
-org_path organizationpath
-ranges fromrange|torange fromrange|torange...-wppool_type type
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```


-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager ucsmanagername

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_name poolname

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-description pooldescription

Defines the pool being created.

-org_path organizationpath

Defines the fully qualified path name to the organization. Valid format:
`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-ranges fromrange | torange fromrange | torange...

Defines the slot ranges for pools. Valid entries per pool type: MAC addresses in the format: `XX:XX:XX:XX:XX:XX`, UUID in the format: `XXXX-XXXXXXXXXXXX`, and WWNN/WWPN in the format: `XX:XX:XX:XX:XX:XX:XX:XX`. Delimit from-to range entries with `|`, and slot ranges with a single space.

-wwpool_type type

Defines the pool type as node or port. Valid entries: node or pool. Default = node.

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Create a WWNN Pool

This example creates a WWNN node pool.

```
dpmucs createwwpool
-ucs_manager EmgManager200
-pool_name TEST_POOL_WW
-description TEST POOL
-org_path "" -ranges 20:00:00:25:B6:01:00:F0|20:00:00:25:B6:01:00:F3
20:00:00:25:B6:01:00:F5|20:00:00:25:B6:01:00:F7 -wwpool_type node -ws_user cba
-ws_password cba
```

dpmucs deletemacpool Command--Delete MAC Address Pool

Use this command to delete a MAC address pool.

This command has the following format:

```
dpmucs deletemacpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-org_path organizationpath
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_name *poolname*

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-org_path *organizationpath*

Defines the fully qualified path name to the organization. Valid format:
Org-root/org-suborg1/org-suborg2/org-suborg3...

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Delete a MAC Address Pool

This example deletes a MAC address pool.

```
dpmucs deletemacpool
-ucs_manager ucsmanager
-pool_name TEST_POOL
-org_path ""
-ws_user cba -ws_password cba
```

dpmucs deleteorg Command--Delete Organization

Use this command to delete an organization.

This command has the following format:

```
dpmucs createorg
[-sc sc_url]
-ucs_manager ucsmanagername
-org_name organizationname
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-org_name *organizationname*

Defines the fully qualified name of the organization. Valid format:

org-root/org-name

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Delete an Organization

This example deletes an organization.

```
dpmucs deleteorg
-ucs_manager EngManager200
-org_dn org-root/org-TEST_ORG
```

dpmucs deleteserverpool Command--Delete Server Pool

Use this command to delete a server pool.

This command has the following format:

```
dpmucs deleteserverpool
[-sc sc_url]
-ucs_manager ucsmanagername
-org_path orgpath
-pool_name poolname
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-org_path *organizationpath*

Defines the fully qualified path name to the organization. Valid format:

`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-pool_name *poolname*

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a Server Pool

This example deletes a server pool.

```
dpmucs.exe deleteserverpool
-ucs_manager ucsmanager
-org_path1/Eng/test
-pool_name TEST_POOL
```

dpmucs deleteserviceprofile--Delete a Service Profile

Use this command to delete a service profile.

This command has the following format:

```
dpmucs deleteserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagername
-service_profile_dn serviceprofiledn
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-service_profile_dn *serviceprofiledn*

The fully qualified and unique name of a blade service profile. The blade service profile must be created before provisioning an OS. A service profile contains boot order, network settings, and storage (local or remote). Valid format: `org-root/serviceprofile`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmucs deleteuuidpool Command--Delete UUID Pool

Use this command to delete a UUID pool.

This command has the following format:

```
dpmucs deleteuuidpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-org_path organizationpath
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_name *poolname*

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-org_path *organizationpath*

Defines the fully qualified path name to the organization. Valid format:
Org-root/org-suborg1/org-suborg2/org-suborg3...

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a UUID Pool

This example deletes a UUID pool.

```
dpmucs deleteuuidpool
-ucs_manager EngManager200
-pool_name "TEST_POOL"
-org_path ""
-ws_user cba
-ws_password cba
```

dpmucs deletewwpool Command--Delete WWNN Pool

Use this command to delete a WWNN pool.

This command has the following format:

```
dpmucs deletewwpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-org_path organizationpath
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```


-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager ucsmanagername

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_name poolname

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-org_path organizationpath

Defines the fully qualified path name to the organization. Valid format:

Org-root/org-suborg1/org-suborg2/org-suborg3...

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a WWNN Pool

This example deletes a WWNN pool.

```
dpmucs deletewwpool
-ucs_manager EngManager200
-pool_name "TEST_POOL"
-org_path ""
-ws_user cba
-ws_password cba
```

dpmucs deployimage Command--Deploy a Cisco UCS Image

Use this command to deploy an image to a Cisco UCS blade using ITCM.

This command has the following format:

```
dpmucs deployimage
[-sc sc_url] -deploytype ITCM
ITCM Only: -passwordEncrypted yes|no -osPassword password
-inputparam "ucsManagerHost|ucsManager"
-inputparam "bladeDn|bladeDN"
-inputparam "macAddress|macaddress"
-inputparam "osType|os_type"
-inputparam "profileDn|profileDN"
-inputparam "profileTemplateDn|profileTemplateDn"
-inputparam "selectFirstBlade|{false|true}"
-inputparam "serviceProfileName|serviceprofilename"
-inputparam "... deploytype_options"
[-ws_user username][-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-inputparam *argument*|*value*

Specifies input parameters for image deployment.

-bootServerName|*bootserver*

Specifies boot server for ITCM deployment.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deploy Image Using ITCM

This example deploys an image to a Cisco UCS blade using ITCM.

```
dpmucs.exe deployimage -deploytype ITCM
-paswordEncrypted no -osPassword secret123
-inputparam "ucsManagerHost|loddisco1-mc1.ca.com"
-inputparam "computerName|hostname1"
-inputparam "ipAddress|blade-2"
-inputparam "macAddress|00:26:51:09:5D:E4"
-inputparam "osType|Any"
-inputparam "osImage|Windows2003_Image -bootServerName|ITCM_BootServer1"
-inputparam "osUser|Administrator"
-inputparam "osPassword|password"
-inputparam "bladeDn|sys/chassis-1/blade-2"
-inputparam "serviceProfileName|blade2_netboot1"
-inputparam "profileDn|org-root/ls-blade2_netboot1"
-inputparam "selectFirstBlade|false"
-inputparam "templateName|Default_Software_List"
-inputparam "profileTemplateDn|org-root/ls-template1"
```

dpmucs disassociateserviceprofile Command--Disassociate Service Profile

Use this command to disassociate a service profile from a blade.

This command has the following format:

```
dpmucs diassociateserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagername
-service_profile_dn serviceprofiledn
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-service_profile_dn *serviceprofiledn*

The fully qualified and unique name of a blade service profile. The blade service profile must be created before provisioning an OS. A service profile contains boot order, network settings, and storage (local or remote). Valid format: *org-root/serviceprofile*.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Disassociate a Service Profile

This example disassociates a service profile.

```
dpmucs diassociateserviceprofile
-ucs_manager EngManager200
-service_profile_dn org-root/l3-dev-boot-from-san-esx-10
```

dpmucs failoverserviceprofile Command--Associate a Service Profile With a Blade for Failover

Use this command to transfer an existing service profile association to another profile for the purpose of blade failover.

This command has the following format:

```
dpmucs failoverserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagername
-service_profile_dn serviceprofiledn
-blade_dn bladename
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-service_profile_dn *serviceprofiledn*

The fully qualified and unique name of a blade service profile. The blade service profile must be created before provisioning an OS. A service profile contains boot order, network settings, and storage (local or remote). Valid format: `org-root/serviceprofile`. The profile specified must be an associated service profile.

-blade_dn *bladename*

The fully qualified and unique name of a blade. If a blade is not provided, an unassociated, available blade is selected. Valid format: `systemid/chassisid/bladeid`. If a blade is not provided (""), a random available blade is selected.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create Failover Service Profile

This example associates a random, available blade with a service profile for failover.

```
dpmucs.exe failoverserviceprofile  
-ucs_manager ucsmanager  
-service_profile_dn org-root/ls-dev-boot-from-san-esx-2 -blade_dn "
```

dpmucs getblades Command--Get Blade Information

Use this command to get information for all blades or a specific blade for a UCS Manager.

This command has the following format:

```
dpmucs getblades  
[-sc sc_url]  
-ucs_manager ucsmanagername  
[-dn distinguishedname]  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-dn *distinguishedname*

Defines the full-qualified, unique name of a Cisco UCS chassis, blade, or switch.
Valid entry: Device name, or an empty string ("") to denote all chassis, blades, or switches. If not specified, device details are listed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get Blade Information

This example gets information for the blade, `-dn "sys/chassis-1/blade-1"` for the UCS Manager, `EngManager200`.

```
dpmucs getblades
-ucs_manager EngManager200
-dn sys/chassis-1/blade-1
-ws_user cba
-ws_password cba
```

dpmucs getchassis Command--Get Chassis Information

Use this command to get information for all chassis, or a specific chassis in a UCS Manager.

This command has the following format:

```
dpmucs getchassis
[-sc sc_url]
-ucs_manager ucsmanagername
[-dn distinguishedname]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-dn *distinguishedname*

Defines the full-qualified, unique name of a Cisco UCS chassis, blade, or switch.

Valid entry: Device name, or an empty string ("") to denote all chassis, blades, or switches. If not specified, device details are listed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Chassis Information

This example gets chassis information for `sys/chassis-1`.

```
dpmucs getchassis
-ucs_manager EngManager200
-dn sys/chassis-1
-ws_user cba
-ws_password cba
```


dpmucs getpolicy Command--Gets Policy from a UCS Manager

Use this command to get policy from a UCS Manager.

This command has the following format:

```
dpmucs getpolicy  
[-sc sc_url]  
-ucs_manager ucsmanagername  
[-policyType policytype|All]  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Policy

This example gets all policy for the specified UCS Manager.

```
dpmucs.exe getpolicy  
-ucs_manager Engmanager  
-policytype All
```

dpmucs getserviceprofilestatus Command--Get Service Profile Status

Use this command to get the status of a service profile.

This command has the following format:

```
dpmucs getserviceprofilestatus  
[-sc sc_url]  
-ucs_manager ucsmanagername  
-service_profile_dn serviceprofiledn  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-service_profile_dn *serviceprofiledn*

The fully qualified and unique name of a blade service profile. The blade service profile must be created before provisioning an OS. A service profile contains boot order, network settings, and storage (local or remote). Valid format: `org-root/serviceprofile`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get Service Profile Status

This example gets the service profile status.

```
dpmucs getserviceprofilestatus
-ucs_manager ucsmanager
-service_profile_dn org-root/ls-dev-boot-from-san-esx-11
```

dpmucs getswitches Command--Get Switch Information

Use this command to get information for all switches or a specific switch for a UCS Manager.

This command has the following format:

```
dpmucs getswitches
[-sc sc_url]
-ucs_manager ucsmanagername
[-dn distinguishedname]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-dn *distinguishedname*

Defines the full-qualified, unique name of a Cisco UCS chassis, blade, or switch. Valid entry: Device name, or an empty string (""), to denote all chassis, blades, or switches. If not specified, device details are listed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Switch Information

This example gets information for the switch, `syst/switch-A`, on the UCS Manager, `EngManager200`.

```
dpmucs getswitches
-ucs_manager EngManager200
-dn sys/switch-A
-ws_user cba
-ws_password cba
```

dpmucs gettopsystem Command--Get System Information

Use this command to get information from the top system for a UCS Manager information.

This command has the following format:

```
dpmucs gettopsystem
[-sc sc_url]
-ucs_manager ucsmanagername
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get System Information

This example gets the top system for the UCS manager, `EngManager200`.

```
dpmucs gettopsystem
-sc https://HOST/dpm/ucsws
-ucs_manager EngManager200
-ws_user cba
-ws_password cba
```

dpmucs listserviceprofiles Command--List Service Profiles

Use this command to list service profiles for a specified UCS Manager.

This command has the following format:

```
dpmucs listserviceprofiles
[-sc sc_url]
-ucs_manager ucsmanagername
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: List Service Profiles

This example lists service profiles for the UCS Manager, `EngManager200`.

```
dpmucs listserviceprofiles
-ucs_manager EngManager200
-ws_user cba
-ws_password cba
```

dpmucs powercycle Command--Power Cycle Blades

Use this command to perform power cycle operations on a blade.

This command has the following format:

```
dpmucs powercycle  
[-sc sc_url]  
-ucs_manager ucsmanagername  
[-dn distinguishedname]  
-operation operation  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-dn *distinguishedname*

Defines the full-qualified, unique name of a Cisco UCS chassis, blade, or switch.

Valid entry: Device name, or an empty string ("") to denote all chassis, blades, or switches. If not specified, device details are listed.

-operation *operations*

Defines the blade power cycle options. Valid entries are:

- 2 = Power Cycle. Immediately powers down the blade, then powers up the blade.
- 3 = Graceful Shut Down. Allows data to be saved, then powers down the blade, and powers up the blade.
- 4 = Reset Immediately. Allows you to unplug the power supply to the blade, and then plug it back without saving data.
- 5 = Reset Wait. Allows data to be saved before you unplug the power supply, and plug it back in.
- 6 = Soft Shut Down. Notifies all running applications of a shutdown so data is saved.
- 7 = Shut Down. Shuts down the blade without saving data.
- 8 = Boot Up. Boots the blade.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Power Cycle a Blade

This example immediately power cycles the blade, `sys/chassis-1/blade-3`.

```
dpmucs powercycle
-ucs_manager EngManager200
-dn "sys/chassis-1/blade-3"
-operation 2
-ws_user cba
-ws_password cba
```


dpmucs renamepool Command--Rename Pool

Use this command to rename a pool type.

This command has the following format:

```
dpmucs renamepool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_type pooltype
-pool_name poolname
-new_pool newpoolname
-org_path organizationpath
-description description
-ranges fromrange|torange fromrange|torange...
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-pool_type *pooltype*

Specifies the pool type. Valid pool types (case-sensitive): MacPool, UuidPool, ComputePool, WWNodeNamePool, WWPortNamePool.

-pool_name *poolname*

Specifies pool name. Valid entry: 1-16 alphanumeric characters.

-new_pool_name *newpoolname*

Defines the new pool name. Valid entry: 1-16 characters, excluding spaces and `^!@#%&*()+={}\|;"/?`~``

-org_path organizationpath

Defines the fully qualified path name to the organization. Valid format:
Org-root/org-suborg1/org-suborg2/org-suborg3...

-ranges fromrange | torange fromrange | torange...

Defines the slot ranges for pools. Valid entries per pool type: MAC addresses in the format: XX:XX:XX:XX:XX:XX, UUID in the format: XXXX-XXXXXXXXXXXX, and WWNN/WWPN in the format: XX:XX:XX:XX:XX:XX:XX. Delimit from-to range entries with |, and slot ranges with a single space.

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Rename a Pool

This example renames a pool.

```
dpmucs.exe renamepool
-ucs_manager ucsmanger
-pool_type UuidPool
-pool_name TEST_POOL
-new_pool_name NEW_TEST_POOL
-org_path ""
-description Windows pool
```

dpmucs serviceprofileupdate Command--Execute Service Profile Operations

Use this command to execute one of the following service profile operations: associate, disassociate, or failover.

This command has the following format:

```
dpmucs serviceprofileupdate
[-sc sc_url]
-ucs_manager ucsmangername
-update_operation operation
-service_profile_dn serviceprofiledn
[-blade_dn bladedn]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ucs_manager *ucsmanagername*

Defines the name of the UCS Manager for retrieving information or an operation.

-update_operation *operations*

Defines the operations to perform for the service profile. Valid entries:

associate = associates a service profile with the specified blade.

disassociate = disassociates a service profile from the specified blade.

failover = fails over from one service profile to another.

-service_profile_dn *serviceprofiledn*

The fully qualified and unique name of a blade service profile. The blade service profile must be created before provisioning an OS. A service profile contains boot order, network settings, and storage (local or remote). Valid format: `org-root/serviceprofile`.

-blade_dn *bladename*

The fully qualified and unique name of a blade. If a blade is not provided, an unassociated, available blade is selected. Valid format: `systemid/chassisid/bladeid`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Execute Operation on a Service Profile

This example associates a service profile for the specified blade.

```
dpmucs.exe serviceprofileupdate
-ucs_manager Engmanager
-update_operation associate
-service_profile_dn org-root/ls-dev-boot-from-san-esx-1 -blade_dn
sys/chassis-1/blade-1
```

CA Hyper-V CLI Commands

Use the CLI to script and automate CA Hyper-V commands and run actions based on the command results. Corresponding commands are also available in the AutoShell.

dpmhyperv AddVMNic Command--Add a Network Adapter Controller to a VM

The dpmhyperv AddVMNic command adds a network adapter to a VM.

This command has the following format:

```
dpmhyperv AddVMNic
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm VM_name|-vmguid vm_ID
[-mac mac_value]
[-switch switch_name]
[-legacy]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-mac *mac_value*

(Optional) Specifies the Media Access Control (MAC) address of the network adapter.

-switch *switch_name*

(Optional) Specifies the virtual network switch name to connect to the network adapter. If this parameter is not specified, then the network adapter does not have any connectivity.

-legacy

(Optional) Specifies whether to create a legacy network adapter instead of a Hyper-V synthetic adapter. This option must only be used when installing operating systems that do not have Hyper-V integration services available.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Add a Network Adapter Controller to a VM

This example adds a network adapter controller to the VM, "TestVM" and connects to it to the switch, "Local Area Connection - Virtual Network."

```
dpmhyperv
-addVMNic
-ws_user dcaadmin
-ws_password #test#
-hypervHost hvserver
-vm TestVM
-switch "Local Area Connection - Virtual Network"
```

dpmhyperv AddVMSCSIController Command--Create a SCSI Controller for a VM

The dpmhyperv AddVMSCSIController command creates a SCSI controller for a VM on a Hyper-V host.

This command has the following format:

```
dpmhyperv AddVMSCSIController  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-vm vm_name|-vguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a New SCSI Controller for a VM on a Hyper-V host

This example creates new SCSI controller for the VM with the GUID, "38F29B6A-8CE2-42D2-8269-BFED14644376" on the Hyper-V host, "hserver."

```
dpmhyperv AddVMSCSIController
-ws_user dcaadmin
-ws_password #test#
-hypervHost hserver
-vmguid 38F29B6A-8CE2-42D2-8269-BFED14644376
```

dpmhyperv AgentGet Command--Return an Agent Value

The dpmhyperv AgentGet command returns an agent value.

This command has the following format:

```
dpmhyperv AgentGet
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-modelPath modelpath_value attributeName attribute_name
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-jobRef *job_ref_identifier*

Specifies the reference identifier of the asynchronous job.

-modelPath *modelpath_value*

Specifies the model path of the object to get value from.

-attributeName *attribute_name*

Specifies the name of the attribute that you want to get.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Return an Agent Value

This example returns the agent value.

```
dpmhyperv AgentGet
-ws_user dcaadmin
-ws_password #test#
-hypervHost hserver -modelPath
"https://localhost/aip/AOM/root/cimv2:CA_ComputerSystem.CreationClassName=\"CA_Co
mputerSystem\",Name=\"8a2a30a0-a184-11df-84a8-00155d7c7405\" \" -attributeName
DPM.HYPERV.SERVER.CRITICALVMS.MONITOR
```

dpmhyperv AgentSet Command--Set an Agent Value

The dpmhyperv AgentSet command sets an agent value.

This command has the following format:

```
dpmhyperv AgentSet
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-modelPath model_path attributeName attribute_name
-snmPValue snmp_value
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-jobRef *job_ref_identifier*

Specifies the reference identifier of the asynchronous job.

-modelPath *modelpath_value*

Specifies the model path of the object to get value from.

-attributeName *attribute_name*

Specifies the name of the attribute that you want to get.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set an Agent Value

This example sets the agent value.

```
dpmhyperv AgentSet
-ws_user admin
-ws_password #test#
-hypervHost hserver
-modelPath
"https://localhost/aip/AOM/root/cimv2:CA_ComputerSystem.CreationClassName=\"CA_Co
mputerSystem\",Name=\"8a2a30a0-a184-11df-84a8-00155d7c7405\" \" -attributeName
DPM.HYPERV.SERVER.CRITICALVMS.MONITOR -snmpValue 1
```

dpmhyperv ChangeVMState Command--Change the State of a VM

The dpmhyperv ChangeVMState command changes the state of a VM on a Hyper-V host.

This command has the following format:

```
dpmhyperv ChangeVMState  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-vm vm_name|-vguid vm_ID  
-state {Start|Suspend|Stop|Pause|Shutdown|Reboot}  
-[pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-state {Start|Suspend|Stop|Pause|Shutdown|Reboot}

Specifies the state of the VM. Options include the following:

Start

Turns on the VM.

Suspend

Suspends the VM temporarily.

Stop

Stops the VM.

Pause

Stops the VM temporarily.

Shutdown

Shuts down the VM.

Reboot

Performs a hard reset of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Change the State of a VM

This example sets the state of the VM, "vmtest" to enabled.

```
dpmhyperv ChangeVMState -ws_user admin -ws_password #test# -hypervHost hvserver -vm vmtest -state start
```

dpmhyperv CreateTemplateFromVM Command--Create a Template from a VM

The dpmhyperv CreateTemplateFromVM command creates a template from an existing VM.

This command has the following format:

```
dpmhyperv CreateTemplateFromVM  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vmname vm_name|-vmguid vm_ID  
-template template_name  
[-description template_description]  
[-path template_location]  
[-async]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-template *template_name*

Defines the name of the new template.

-description *template_description*

(Optional) Specifies the template description.

-path *template_location*

(Optional) Defines the full path of the directory where you want to create the template.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a Template from a VM

This example creates the template, "MyTemplate" from the VM, "TestVM."

```
dpmhyperv -CreateTemplateFromVM -ws_user admin -ws_password #test#
-hypervHost hvserver -vm TestVM -template MyTemplate
```

dpmhyperv CreateVirtDisk Command--Create a Virtual Disk Image

The dpmhyperv CreateVirtDisk command lets you create a virtual disk image on a Hyper-V host.

This command has the following format:

```
dpmhyperv CreateVirtDisk
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-path location
-diskType {dynamic|fixed}
-size drive_size
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-path *location*

Defines the full path of the virtual disk image.

-diskType {*fixed*|*dynamic*}

Specifies the type of the virtual disk image. Options include the following:

fixed

Indicates that the disk is of fixed size and cannot be increased later.

dynamic

Indicates that the disk is of dynamic size and can be increased later on as required.

-size *drive_size*

Specifies the size of the drive in GB.

Note: The size of the disk can be changed later on.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a Virtual Disk Image

This example creates a virtual disk, "01.vhd" of 5 GB.

```
dpmhyperv CreateVirtDisk -hypervHost hvserver -path C:\data\Disks\01.vhd -type
dynamic -size 5 -async
```

dpmhyperv CreateVirtFloppy Command--Create a Virtual Floppy Disk Image

The dpmhyperv CreateVirtFloppy command creates a virtual floppy disk image on a Hyper-V host.

This command has the following format:

```
dpmhyperv CreateVirtFloppy  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-path location  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-path *location*

Defines the full path of the virtual floppy image

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Creates a Virtual Floppy Disk image

This example creates a virtual floppy disk image on the Hyper-V host, "hvserver."

```
dpmhyperv CreateVirtFloppy -ws_user dcaadmin -ws_password #test# -hypervHost hvserver -path C:\data\Floppy\1.vfd
```

dpmhyperv CreateVM Command--Create a VM

The dpmhyperv CreateVM command creates a VM on a Hyper-V host.

This command has the following format:

```
dpmhyperv CreateVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
[-templatevm vm_template]
[-templatevmguid template_ID]
[-path location]
[-memory memory_size]
[-cpuSocketCount total_CPU_socket_count]
[-cpuidLimit]
[-cpufeatlimit]
[-cpuReserve cpu_reserve]
[-cpulimit max_CPU_resources]
[-cpuWeight cpu_weight]
[-startAction {none|auto|always}]
[-startDelay start_delay_secs]
[-stopAction {save|off|shutdown}]
[-recoveryAction {none|restart|revert}]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-templatevm *vm_template*

(Optional) Specifies the name of the VM from which you want to copy the configuration. This parameter identifies an existing VM in the Hyper-V environment and not a template from the VM catalog.

Note: We do not recommend creating VMs from the existing VMs because the shared resources results in potential conflicts.

-templatevmguid *template_ID*

(Optional) Specifies the GUID of the VM from which you want to copy the configuration. This parameter identifies an existing VM in the Hyper-V environment and not a template from the VM catalog.

Note: We do not recommend creating VMs from the existing VMs because the shared resources results in potential conflicts.

-path *location*

(Optional) Defines the full path of the new VM.

-memory *memory_size*

(Optional) Defines the memory size of the VM in MB.

-cpuSocketCount *total_CPU_socket_count*

(Optional) Defines the number of CPU cores in the VM. The number of the CPU cores of the VM cannot be more than the total number of the CPU cores available in the Hyper-V host.

-cpuidLimit

(Optional) Indicates whether the processor must limit the maximum CPUID value.

-cpufeatlimit

(Optional) Indicates whether the VM must limit the CPU features exposed to the operating system.

-cpuReserve *cpu_reserve*

(Optional) Specifies the amount of CPU resources that are reserved for use by the VM.

-cpuLimit *max_CPU_resources*

(Optional) Specifies the maximum amount of CPU resources that the VM can consume.

-cpuWeight *cpu_weight*

(Optional) Specifies the relative weight of the virtual machine from 1 to 10000.

Default: 100

-startAction {none|auto|always}

(Optional) Specifies the action to perform after the VM startup. Options include the following:

none

Performs no action on the VM.

auto

Starts the VM automatically if it was running before the Hyper-V host was shut down.

always

Starts the VM every time Hyper-V starts.

-startDelay *start_delay_secs*

(Optional) Specifies the delay in seconds to start the VM after the Hyper-V host starts up.

-stopAction {save|off|shutdown}

(Optional) Specifies the action to perform on the VM before the Hyper-V host stops. Options include the following:

save

Suspends the VM.

off

Turns off the VM.

shutdown

Shuts down the VM. To use this option, the Hyper-V tools must be installed.

-recoveryAction {none|restart|revert}

(Optional) Specifies the action to perform on the VM after the VM worker process terminates abnormally and recovers. Options include the following:

none

Performs no action.

restart

Restarts the VM.

revert

Returns to the last snapshot. To use this option, the VM should have at least one snapshot.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a VM

This example creates the VM, "TestVM" in the "VM" folder.

```
dpmhyperv CreateVM -ws_user dcaadmin -ws_password #test# -hypervHost hvserver -name  
TestVM -path c:\VM\TestVM
```

dpmhyperv CreateVMFromTemplate Command--Create a VM from a SCVMM Server Template

The dpmhyperv CreateVMFromTemplate command creates a VM from an SCVMM server template.

This command has the following format:

```
dpmhyperv CreateTemplateFromVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
[-prompt {yes|no}] -hypervHost hostname
-vmname vm_name
-template template_name
[-dest destination]
[-computerName computer_Name]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcp]
[-ip4dhcpInt ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-adminPass admin_password]
[-productKey product_key]
[-userName user_name]
[-organization org_name]
[-domain domain_name]
[-domainAdmin domain_admin_login]
[-domainAdminPass domain_admin_password]
[-adminUser admin_username]
[-startVM start_vm]
{-memory memory_size}
[cpuSocketCount total_CPU_socket]
[-scvmmHost scvmm_host]
[-scvmmHardwareProf scvmm_hardware_profile]
[-scvmmGuestOSProf scvmm_guest_os_profile]
[-auto_deploy autodeploy]
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the user name and password.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-template *template_name*

Defines the name of the new template.

-dest *destination_path*

(Optional) Specifies the destination path where the template is created.

-computerName *computer_name*

(Optional) Specifies the computer name of the VM.

-ip4addr *ip4addr*

(Optional) Specifies the static IPv4 address that you want to assign to the VM interface.

-ip4dhcp *ip4dhcp*

(Optional) Specifies whether to use DHCP.

Default: local

-ip4dhcpInt *ip4dhcpInt*

(Optional) Specifies the DHCP address.

-ip4mask *network_mask*

(Optional) Specifies the subnet mask that you want to assign to the VM. Use this option with the *-ip4addr* option.

-ip4gw *gateway_address*

(Optional) Specifies the option to set the gateway for the VM. Use this option with the *-ip4addr* option.

-ip4metric *ip_metric*

(Optional) Specifies the interface metric that you want to set for the VM. Use this option with the *-ip4addr* option.

-ip4dns *dnserver_ip*

(Optional) Specifies the DNS server for the VM. Use this option with the *-ip4addr* option.

-adminPass *admin_password*

(Optional) Specifies the default administrator password for the VM.

-productKey *product_key*

(Optional) Specifies the Windows product activation key for the VM.

-userName *user_name*

(Optional) Specifies the user name of the Windows in the VM.

-organization *org_name*

(Optional) Specifies the organization name of the Windows OS in the VM.

-domain *domain_name*

(Optional) Specifies the domain name of the VM.

-domainAdmin *domain_admin_login*

(Optional) Specifies the domain administrator login.

-domainAdminPass *domain_admin_password*

(Optional) Specifies the password for the domain administrator account. This option is invalid for the asynchronous mode.

-adminUser *admin_username*

(Optional) Specifies the user name that is the member of the default Administrators group.

-startVM *start_vm*

(Optional) Indicates whether to start the VM.

-memory *memory_size*

(Optional) Defines the size of the memory in bytes.

-cpuSocketCount *total_CPU_socket*

(Optional) Specifies the number of processor sockets in the VM.

-scvmmHost *scvmm_hostname*

(Optional) Specifies the name of the SCVMM server host.

-scvmmHardwareProf *scvmm_hardware_profile*

(Optional) Specifies the name of the SCVMM defined hardware profile.

-scvmmGuestOSProf *scvmm_guest_os_profile*

(Optional) Specifies the name of the SCVMM defined guest operating system profile.

-autodeploy *autodeploy*

(Optional) Indicates automatic deployment.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

Note: If you are not using the -async mode, the following parameters are ignored: memory, cpuSocketCount, cpuidLimit, cpufeatlimit, cpuReserve, cpuLimit, cpuWeight, startAction, startDelay, stopAction, and recoveryAction.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a VM from a Standalone Server Template

This example creates the VM, "NewVM" using the standalone template, "Win2kSysPrepped."

```
dpmhyperv createVMFromTemplate -ws_user dcaadmin -ws_password #test# -hypervHost  
hvserver -vm NewVM -dest c:\VMs\NewBox -template Win2k3SysPrepped -ip4addr 127.0.0.1  
-ip4mask 255.255.255.0 -ip4gw 127.0.0.1 -ip4dns 127.0.0.1 -computerName NewBox
```

dpmhyperv CreateVMFromTemplate Command--Create a VM from a Template

The dpmhyperv CreateVMFromTemplateEx command creates a VM from a standalone Hyper-V server template.

This command has the following format:


```
dpmhyperv CreateTemplateFromVMEx
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
[-prompt {yes|no}]
-hypervHost hostname
-vmname vm_name
-template template_name
[-dest destination]
[-computerName computer_Name]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcp]
[-ip4dhcpInt ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-disableAdmin]
[-adminPass admin_password]
[-autoLogon autologon]
[-duplicatorString duplicate_String]
[-timeZone timezone]
[-productKey product_key]
[-userName user_name]
[-organization org_name]
[-domain domain_name]
[-domainAdmin domain_admin_login]
[-domainAdminPass domain_admin_password]
[-workgroup workgroup_name]
[-adminUser admin_username]
[-adminUserPass admin_password] [adminGroup admin_group_name]
[-startVM start_vm]
[-custom custom]
[-memory memory_size] [cpuSocketCount total_CPU_socket]
[-cpuidLimit cpu_id_limit]
[-cpufeatlimit cpu_feat_limit]
[-cpuReserve cpu_reserve]
[-cpuLimit max_CPU_usage]
[-cpuWeight cpu_wieight]
[-startAction {none|auto|always}]
[-startDelay start_delay_secs]
[-stopAction {save|off|shutdown}]
[-recoveryAction {none,restart,revert}]
[-scvmmHost scvmm_host]
[-scvmmHardwareProf scvmm_hardware_profile]
[-scvmmGuestOSProf scvmm_guest_os_profile]
[-async]
[-pre]
```

[-post]

[-locale *iso639value*]

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the user name and password.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-template *template_name*

Defines the name of the new template.

-dest *destination_path*

(Optional) Specifies the destination path where the template is created.

-computerName *computer_name*

(Optional) Specifies the name of the computer.

-ip4addr *ip4addr*

(Optional) Specifies the static IPv4 address that you want to assign to the VM interface.

-ip4dhcp *ip4dhcp*

(Optional) Specifies whether to use DHCP.

Default: local

-ip4dhcpInt *ip4dhcpInt*

(Optional) Specifies the DHCP address.

-ip4mask *network_mask*

(Optional) Specifies the subnet mask that you want to assign to the VM. Use this option with the -ip4addr option.

-ip4gw gateway_address

(Optional) Specifies the option to set the gateway for the VM. Use this option with the *-ip4addr* option.

-ip4metric ip_metric

(Optional) Specifies the interface metric that you want to set for the VM. Use this option with the *-ip4addr* option.

-ip4dns dnsserver_ip

(Optional) Specifies the DNS server for the VM. Use this option with the *-ip4addr* option.

-disableAdmin

(Optional) Specifies an option to disable default administrator account for the VM.

-adminPass admin_password

(Optional) Specifies the default administrator password for the VM.

-autoLogon autologon

(Optional) Specifies the option to set the number of accounts that automatically log in with the default administrator account Sysprep process is complete.

-duplicatorString duplicate_string

(Optional) Specifies the name of the system duplicator to set in the VM registry.

-timeZone timezone

(Optional) Specifies the time zone used by the VM that are created using the template.

-productKey product_key

(Optional) Specifies the Windows product activation key for the VM.

-userName user_name

(Optional) Specifies the user name of the Windows in the VM.

-organization org_name

(Optional) Specifies the organization name of the Windows in the VM.

-domain domain_name

(Optional) Specifies the domain name of the VM.

-domainAdmin domain_admin_login

(Optional) Specifies the domain administrator login.

-domainAdminPass domain_admin_password

(Optional) Specifies the password for the domain administrator account. This option is invalid for the asynchronous mode.

-workgroup *workgroup_name*

(Optional) Specifies the workgroup of the VM. This option is invalid for the asynchronous mode.

-adminUser *admin_username*

(Optional) Specifies the user name that is the member of the default Administrators group.

-adminUserPass *admin_password*

(Optional) Specifies the password of the default Administrators group.

-adminGroup *admin_group_name*

(Optional) Specifies the group name of the administrator.

-startVM *start_vm*

(Optional) Indicates whether to start the VM.

-custom *custom*

(Optional) Specifies the list of comma-separated custom commands that are executed at the end of the Sysprep process.

-memory *memory_size*

(Optional) Defines the size of the memory in bytes.

-cpuSocketCount *total_CPU_socket*

(Optional) Specifies the number of processor sockets in the VM.

-cpuidLimit *cpu_id_limit*

(Optional) Indicates whether the processor must limit the maximum CPUID value.

cpufeatlimit *cpu_feat_limit*

(Optional) Indicates whether the VM must limit the CPU features exposed to the operating system.

-cpuReserve *cpu_reserve*

(Optional) Specifies the amount of CPU resources that are reserved for use by the VM.

-cpuLimit *max_CPU_usage*

(Optional) Specifies the maximum amount of CPU resources that the VM can consume.

-cpuWeight *cpu_weight*

(Optional) Specifies the relative weight of the virtual machine from 1 to 10000.

-startAction {none|auto|always}

(Optional) Specifies the action to perform on the VM after the Hyper-V host starts up. Options include the following:

none

Performs no action.

auto

Starts the VM automatically if it was running before the Hyper-V host was shut down.

always

Starts the VM every time Hyper-V starts.

-startDelay *start_delay_secs*

(Optional) Specifies the delay in seconds to start the VM after the Hyper-V host starts up.

-stopAction {save|off|shutdown}

(Optional) Specifies the action to perform on the VM before the Hyper-V host shuts down. Options include the following:

save

Suspends the VM.

off

Turns off the VM.

shutdown

Shuts down the VM.

-recoveryAction {none,restart,revert}

(Optional) Specifies the action to perform on the VM after the Hyper-V host restarts after an unexpected shutdown. Options include the following:

none

Performs no action.

restart

Restarts the VM.

revert

Returns to the last snapshot.

-scvmmHost *scvmm_hostname*

(Optional) Specifies the name of the SCVMM server host.

-scvmmHardwareProf *scvmm_hardware_profile*

(Optional) Specifies the name of the SCVMM defined hardware profile.

-scvmmGuestOSProf *scvmm_guest_os_profile*

(Optional) Specifies the name of the SCVMM defined guest operating system profile.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

Note: If you are not using the -async mode, the following parameters are ignored: memory, cpuSocketCount, cpuidLimit, cpufeatlimit, cpuReserve, cpuLimit, cpuWeight, startAction, startDelay, stopAction, and recoveryAction.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a VM from a Standalone Hyper-V Server Template

This example creates the VM, "NewVM" using the template, "Win2kSysPrepped."

```
dpmhyperv -createVMFromTemplateEx -ws_user dcaadmin -ws_password #test# -hypervHost hvserver -vm NewVM -dest c:\VMs\NewBox -template Win2k3SysPrepped -ip4addr 127.0.0.1 -ip4mask 255.255.255.0 -ip4gw 127.0.0.1 -ip4dns 127.0.0.1 -computerName NewBox
```

dpmhyperv DeleteVM Command--Delete a VM

The dpmhyperv DeleteVM command deletes a VM on a Hyper-V host. This command also deletes virtual disks, virtual floppy drives, and ISO images assigned to a VM.

This command has the following format:

```
dpmhyperv DeleteVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password] -hypervHost hostname
-vm vm_name|-vguid vm_ID
[-delvhd]
[-delvfd]
[-deliso]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-delvhd

(Optional) Indicates whether to delete virtual hard drives.

-delvfd

(Optional) Indicates whether to delete virtual floppy disk images associated with the VM.

-deliso

(Optional) Indicates whether to delete virtual CD/DVD images associated with the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Delete a VM from a Hyper-V Host

This example deletes the VM with the GUID, "346597794004894538F29B6A-8CE2-42" from the Hyper-V host, "hserver". This command also deletes virtual disks images, virtual floppy drives, and CD/DVD ISO images assigned to the VM.

```
dpmhyperv DeleteVM -ws_user admin -ws_password #test# -hypervHost hserver -vmguid 38F29B6A-8CE2-42D2-8269-BFED14644376 -delvhd -delvfd -deliso
```

dpmhyperv ExpandVirtDisk Command--Expand the Size of a Virtual Disk

The dpmhyperv ExpandVirtDisk command expands the size of a virtual disk image on a Hyper-V host.

This command has the following format:

```
dpmhyperv ExpandVirtDisk
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-path location
-size disk_size
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-path *location*

Defines the location of the virtual disk.

-size *drive_size*

Specifies the size of the drive in GB.

Note: The size of the disk can be changed later on.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Expand the Size of a Virtual Disk

This example expands the size of the virtual disk image on the Hyper-V host, "hserver" to 2 GB in the asynchronous mode. This command returns the Hyper-V job ID of the created task.

```
dpmhyperv ExpandVirtDisk -ws_user admin -ws_password #test# -hypervHost hserver -path C:\data\Disks\1.vhd -size 2 -async
```

dpmhyperv ExportVM Command--Export a VM

The dpmhyperv ExportVM command stores files having information of a VM to a directory on a Hyper-V host. This command is deprecated, use dpmhv ExportVMEx instead.

This command has the following format:

```
dpmhyperv ExportVM  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vm_name|-vmguid vm_ID  
-dest dest_location  
[-nostate]  
[-async]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-dest *dest_location*

(Optional) Defines the destination location of the imported file.

-nostate

(Optional) Specifies that only the VM specification is exported. This option prevents the exported VM from trying to refer to the original VM disk images.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Export a VM

This example stores files having information of the VM, "TestVM" from the Hyper-V host, "hserver."

```
dpmhv ExportVM -hypervHost hserver -vm TestVM -dest c:\DATA\Export
```

dpmhyperv ExportVMEx Command--Export a VM

The dpmhyperv ExportVM command lets you export a virtual machine.

This command has the following format:

```
dpmhyperv ExportVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm vm_name|-vmguid vm_ID
-dest dest_folder
[-noruntimeInfo]
[-nosnapshots]
[-nostorage]
[-nosubdir]
[-snapshot snapshot_name][[-snapshotid snapshot_ID]
[-nostate]
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-dest *dest_location*

(Optional) Defines the destination location of the imported file.

-noruntimeInfo

(Optional) Specifies not to export the run time-related information.

-nosnapshots

(Optional) Specifies not to export snapshots.

-nostorage

(Optional) Specifies not to export virtual disk image files.

-nosubdir

(Optional) Specifies not to create a sub directory with the VM display name in the directory specified as the export path.

-snapshot *snapshot_name*

(Optional) Specifies the name of the snapshot that you want to export.

-snapshot *snapshot_ID*

(Optional) Specifies the GUID of the snapshot that you want to export.

-nostate

(Optional) Specifies that only the VM specification is exported. This option prevents the exported VM from trying to refer to the original VM disk images.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Export a VM

This example exports the VM, "TestVM" without snapshots and creates the exported VM, "textvmExp."

```
dpmhyperv ExportVMEx -ws_user admin -ws_password #test# -hypervHost hserver -vm
TestVM -dest c:\Ex\TestVMExp -nosubdir -nosnapshots
```

dpmhyperv GetHostSwitches Command--Get Switches of a Host

The dpmhyperv GetHostSwitches command returns switches of a Hyper-V host. You can specify a filter condition to get specific switches.

This command has the following format:

```
dpmhyperv GetHostSwitches
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname [filter filter_condition]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-jobRef *job_ref_identifier*

Specifies the reference identifier of the asynchronous job.

-filter *condition*

(Optional) Specifies a filter condition such as "*".

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Switches of a Host

This example returns switches of the host, "hvserver."

```
dpmhyperv GetHostSwitches -ws_user admin -ws_password #test# -hypervHost hvserver
```

Example: Get Switches of a Host Using Filters

This example returns all the switches having "n" somewhere in the switch name.

```
dpmhyperv GetHostSwitches -ws_user admin -ws_password #test# -hypervHost hvserver  
-filter *n*
```

dpmhyperv GetJobErrorInfo Command--Get the Error Information of a Job

The dpmhyperv GetJobErrorInfo command returns the error details of an asynchronous job running on a Hyper-V host.

This command has the following format:

```
dpmhyperv GetJobErrorInfo  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-jobRef job_ref_identifier  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-jobRef *job_ref_identifier*

Specifies the reference identifier of the asynchronous job.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get the Error Information of a Job

This example returns the error information of the job,

```
"\\hserver\root\virtualization:Msvm_ConcreteJob.InstanceID="86C9700348-4117-ABA8-445456"."
```

```
dpmhyperv GetJobErrorInfo -ws_user admin -ws_password admin -hypervHost hvserver
-jobRef
\\hserver\root\virtualization:Msvm_ConcreteJob.InstanceID="86C9700348-4117-ABA8-445456"
```

dpmhyperv GetJobInfo Command--Get Information of a Job

The dpmhyperv GetJobInfo command returns the details of an asynchronous job on a Hyper-V host. The details include the percentage completed, error information, and the start time.

This command has the following format:

```
dpmhyperv GetJobInfo
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-jobRef job_ref_identifier
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-jobRef *job_ref_identifier*

Specifies the reference identifier of the asynchronous job.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Information of a Job

This example returns the information about the Hyper-V job,

```
"\\hserver\root\virtualization:Msvm_ConcreteJob.InstanceID="86TSTC4-ABA8-C82734036"."
```

```
dpmhyperv GetJobInfo -ws_user admin -ws_password #test# -hypervHost hvserver -jobRef \\hserver\root\virtualization:Msvm_ConcreteJob.InstanceID="86TSTC4-ABA8-C82734036"
```


dpmhyperv GetJobState Command--Get the State of a Job

The dpmhyperv GetJobState command returns the state of an asynchronous job running on a Hyper-V host.

This command has the following format:

```
dpmhyperv GetJobState  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-jobRef job_ref-identifier  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-jobRef *job_ref_identifier*

Specifies the reference identifier of the asynchronous job.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get the State of a Job

This example returns the state of the job,

```
"\\hvserver\root\virtualization:Msvm_ConcreteJob.InstanceID="86CD04TEST82795D6F036"."
```

```
dpmhyperv GetJobState -ws_user admin -ws_password #test# -hypervHost hserver -jobRef \\hserver\root\virtualization:Msvm_ConcreteJob.InstanceID="86CD04TEST82795D6F036"
```

dpmhyperv GetServerInfo Command--Get all Managed Servers

The dpmhyperv GetServerInfo command returns all Hyper-V host with its properties in the CA Server Automation system.

This command has the following format:

```
dpmhyperv GetServerInfo  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

This example returns the properties of each Hyper-V host registered in the CA Server Automation system.

```
dpmhyperv GetServerInfo -ws_user admin -ws_password #test#
```

dpmhyperv GetVersion Command--Get the PMM Version

The dpmhyperv GetVersion command returns the version of the Platform Management Module (PMM).

This command has the following format:

```
dpmhyperv GetVersion  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get the PMM Version

This example displays the version of the current Hyper-V PMM.

```
dpmhyperv GetVersion -ws_user admin -ws_password #test#
```

dpmhyperv GetVMProperties Command--Get VM Properties

The dpmhyperv GetVMProperties command lets you view VM properties.

This command has the following format:

```
dpmhyperv GetVMProperties  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname| -vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get VM Properties

This example displays the properties of the VM, "vm-01" on the host, "hserver."

```
dpmhyperv GetVMProperties -ws_user admin -ws_password #test# -hypervHost hserver -vm
vm01
```

dpmhyperv GetVMState Command--Get a VM State

The dpmhyperv GetVMState command returns the current state of a VM. The following list describes the four states:

- disabled: VM is turned off
- enabled: VM is up and running
- paused: VM is temporarily stopped
- suspended: VM is stopped

This command has the following format:

```
GetVMState
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm vmname| -vmguid vm_ID
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get a VM State Using the VM Name

This example returns the state of the VM, "vm01" on the host, "hserver."

```
dpmhyperv GetVMState -ws_user admin -ws_password #test# -hypervHost hserver -vm test-vm01
```

Example: Get a VM State Using the VM GUID

This example returns the state of the virtual machine having the GUID, "38FTESTED14644376" on the host, "hserver."

```
dpmhyperv GetVMState -ws_user admin -ws_password #test# -hypervHost hserver -vmguid 38FTESTED14644376
```

dpmhyperv ImportVM Command--Import a VM

The dpmhyperv ImportVM command lets you import a VM to a Hyper-V hypervHost.

This command has the following format:

```
dpmhyperv ImportVM  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-path location  
[-name vm_name]  
[-noNewID]  
[-async]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-path *location*

Defines the location of the imported VM.

-vm *vm_name*

Specifies the VM name.

-noNewID

(Optional) Indicates not to generate a new VM GUID. In this case, the imported VM and the imported from VM have the same GUID. We recommend not to use this option to avoid potential conflicts with other VMs.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Import a VM to a Hyper-V Host

This example imports the VM, "test-vm01-imported" to the Hyper-V host, "hserver."
This command does not generate a new ID.

```
dpmhyperv ImportVM -ws_user admin -ws_password #test# -hypervHost hserver -path  
C:\data\Export\test-vm01 -name test-vm01-imported -noNewID.
```

dpmhyperv ImportVMEx Command--Copy and Import a VM

The dpmhyperv ImportVMEx command copies and imports an already exported VM to a Hyper-V host.

This command has the following format:

```
dpmhyperv ImportVMEx  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-path location [dest dest_location]  
[-name vm_name]  
[-noNewID]  
[-async]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-path *location*

Defines the location of the server.

-dest *dest_location*

(Optional) Defines the destination location of the imported file.

-vm *vm_name*

Specifies the VM name.

-noNewID

(Optional) Indicates not to generate a new VM GUID. In this case, the imported VM and the imported from VM have the same GUID. We recommend not to use this option to avoid potential conflicts with other VMs.

-async

(Optional) Runs the command in the asynchronous mode.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Copy and Import a VM to a Hyper-V host

This example imports an exported VM to a Hyper-V host and does not generate a new ID for it.

```
dpmhyperv ImportVMEx -ws_user admin -ws_password #test# -hypervHost hserver -path
C:\data\Export\test -name test-vm01-imported -noNewID
```

dpmhyperv SetClustered Command--Add or Remove a VM from a Cluster

The dpmhyperv SetClustered command either adds or removes a VM from the Hyper-V cluster.

This command has the following format:

```
dpmhyperv SetClustered
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm VM_name|-vmguid vm_ID
-clustered {on|off}
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-clustered {on|off}

Indicates whether to add or remove the VM from the cluster.

on

Indicates to add the VM to the cluster.

off

Indicates to remove the VM from the cluster.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Append a VM to a Hyper-V Cluster

This example appends the VM with the GUID, "38F29B6A-TEST-BFED14644376" to the Hyper-V cluster.

```
dpmhyperv SetClustered -ws_user admin -ws_password #test# -hypervHost hserver -vmguid 38F29B6A-TEST-BFED14644376 -clustered on
```

Example: Remove a VM from a Hyper-V Cluster

This example removes the VM, "test-vm01" from the Hyper-V cluster.

```
dpmhyperv SetClustered -ws_user admin -ws_password #test# -hypervHost hserver -vm test-vm01 -clustered off
```

dpmhyperv SetSysPrepProperties Command--Sets Sysprep Properties

The dpmhyperv SetSysPrepProperties command lets you set sysprep properties on a Hyper-V server.

This command has the following format:

```
dpmhyperv SetSysPrepProperties
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm VM_name|-vmguid vm_ID
[-jobRef job_reference]
[-computerName computer_name]
[-ip4addr ip4_address]
[-ip4dhcp]
[-ip4dhcpint ip4dhcpint]
[-ip4mask network_mask]
[-ip4gw gateway_address]
[-ip4metric ip_metric]
[-ip4dns dnsserver_ip]
[-disableAdmin]
[-adminPass administrator_password]
[-autoLogon autologon]
[-duplicatorString duplicator_string]
[-timezone timezone]
[-productKey product_key]
[-userName user_name]
[-organization org_name]
[-domain domain_name]
[-domainAdmin domain_admin_login]
[-domainAdminPass domain_admin_password]
[-workgroup workgroup_name]
[-adminUser admin_username]
[-adminUserPass admin_password]
[-adminGroup admin_groupname]
[-custom custom]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-jobRef *job_reference*

Specifies the job identifier.

-computerName *computer_name*

(Optional) Specifies the name of the computer.

-ip4addr *ip4_address*

(Optional) Specifies the static IPv4 address that you want to assign to the VM interface.

-ip4dhcp *ip4dhcplnt*

(Optional) Specifies the DHCP address.

-ip4mask *network_mask*

(Optional) Specifies the subnet mask that you want to assign to the VM. Use this option with the *-ip4addr* option.

-ip4gw *gateway_address*

(Optional) Specifies the option to set the gateway for the VM. Use this option with the *-ip4addr* option.

-ip4metric *ip_metric*

(Optional) Specifies the interface metric that you want to set for the VM. Use this option with the *-ip4addr* option.

-ip4dns *dnsserver_ip*

(Optional) Specifies the DNS server for the VM. Use this option with the *-ip4addr* option.

-disableAdmin

(Optional) Specifies the option to disable default administrator account for the VM. Support for this parameter requires a Windows image created using Sysprep tool. This option is invalid in the asynchronous mode.

-adminPass *administrator_password*

(Optional) Specifies the default administrator password for the VM.

-autoLogon *autologon*

(Optional) Specifies the option to set the number of accounts that automatically log in with the default administrator account after the Sysprep process completes.

-duplicatorString *duplicate_string*

(Optional) Specifies the name of the system duplicator to set in the VM registry.

-timeZone *timezone*

(Optional) Specifies the time zone used by the VM that are created using the template.

Default:

-1

-productKey *product_key*

(Optional) Specifies the Windows product activation key for the VM.

-userName *User_name*

(Optional) Specifies the user name of the Windows in the VM.

-organization *org_name*

(Optional) Specifies the user name of the Windows in the VM. This option is invalid in the asynchronous mode.

-domain *domain_name*

(Optional) Specifies the domain name of the VM.

-domainAdmin *domain_admin_login*

(Optional) Specifies the domain administrator login.

-domainAdminPass *domain_admin_password*

(Optional) Specifies the password for the domain administrator account. This option is invalid for the asynchronous mode.

-workgroup *workgroup_name*

(Optional) Specifies the workgroup of the VM. This option is invalid for the asynchronous mode.

-adminUser *admin_username*

(Optional) Specifies the user name that is the member of the default Administrators group.

-adminUserPass *admin_password*

(Optional) Specifies the password of the default Administrators group.

-custom *custom*

(Optional) Specifies the list of comma-separated custom commands that are executed at the end of the Sysprep process.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Sysprep Properties on the Hyper-V Server

This example sets the sysprep properties on the host, "hserver."

```
dpmhyperv SetSysPrepProperties -hypervHost hserver -vm VM_05 -ip4addr 130.119.0.40  
-ip4mask 255.255.0.0 -ip4gw 130.119.0.100
```

dpmhyperv SetVMDisk Command--Assign a Drive or Disk to a Drive Controller

The SetVMDisk command lets you assign a virtual drive or disk to a drive controller.

This command has the following format:

```
dpmhyperv SetVMDisk  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-vm vm_name|-vmguid vm_ID  
-drive {HDD|DVD} -driveType {scsi|ide}  
[-path location]  
-id drive_ID_number  
[-lun logical_unit_num]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-drive {HDD|DVD}

Specifies the type of drive that the image is assigned to. Options include the following:

HDD

Indicates that the drive type is hard drive.

DVD

Indicates that the drive type is CD/DVD drive.

-driveType {scsi|ide}

Specifies the type of drive controller that the image is assigned to. Options include the following:

scsi

Indicates that the drive controller is SCSI.

ide

Indicates that the drive controller is IDE. For CD/DVD images the controller type must IDE.

-path *location*

(Optional) Defines the location of the virtual disk or drive.

-id *drive_ID_number*

Specifies the ID number of the drive. Specify 0 for the IDE and 0,1, or 2 for the SCSI drive.

-lun *logical_unit_num*

(Optional) Specifies the Logical Unit Number (LUN) number of the SCSI drive. For IDE controllers, the LUN must be 0 or 1 and for SCSI controllers, the LUN must be from 0 to 63. If no number is specified, the available channel is assigned automatically.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Assign a Drive on a Drive Controller

This example assigns the virtual hard drive image file, "disk.vhd" to the first channel on the first SCSI controller:

```
dpmhv-setVMDisk -hypervHost hvserver -vm TestVM -type scsi -id 0 -lun 0 -drive HDD  
-path c:\VHDs\disk.vhd
```

dpmhyperv SetVMProperties Command--Set Properties of a VM

The dpmhyperv SetVMProperties command lets you change the properties of a VM.

This command has the following format:

```
dpmhyperv SetVMProperties  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vm_name|-vguid vm_ID  
[-memory memory_size]  
[-cpuSocketCount total_CPU_socket_count]  
[-cpuidLimit CPU_ID_Limit]  
[-cpufeatlimit CPU_Feature_Limit]  
[-cpuReserve cpu_reserver]  
[-cpuReserve cpu_reserve]  
[-cpuLimit max_CPU_usage]  
[-cpuWeight cpu_wieight]  
[-startAction {none|auto|always}]  
[-startDelay start_delay_secs]  
[-stopAction {save|off|shutdown}]  
[-recoveryAction {none|restart|revert}]  
[-pre]  
[-post]  
[-locale iso639value]
```


-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-memory *memory_size*

(Optional) Defines the size of the memory size of the VM in MB.

-cpuSocketCount *total_CPU_socket_count*

(Optional) Defines the number of CPU cores in the VM. The CPU cores cannot be more than the total number of the CPU cores available in the Hyper-V host.

-cpuidLimit *CPU_ID_Limit*

(Optional) Indicates whether the VM must lower the CPU identifier for better compatibility with legacy operating systems like Windows NT.

cpufeatlimit *CPU_Feature_Limit*

(Optional) Indicates whether the VM must limit the CPU features exposed to the operating system.

-cpuReserve *cpu_reserve*

(Optional) Specifies the amount of CPU resources that are reserved for use by the VM.

-cpuLimit *max_CPU_usage*

(Optional) Specifies the maximum amount of CPU resources that the VM can consume.

-cpuWeight *cpu_weight*

(Optional) Specifies the relative weight of the virtual machine from 1 and 10000.

Default: 100

-startAction {none|auto|always}

(Optional) Specifies the time interval to wait before performing the start-up action. Options include the following:

none

Performs no action.

auto

Starts the VM automatically if it was running before the Hyper-V host was shut down.

always

Starts the VM every time Hyper-V starts.

-startDelay *start_delay_secs*

(Optional) Specifies the delay in seconds to start the VM after the Hyper-V host is started.

-stopAction {save|off|shutdown}

(Optional) Specifies the action to perform on the VM before the Hyper-V host shuts down. Options include the following:

save

Suspends the VM.

off

Turns off the VM.

shutdown

Shuts down the VM.

-recoveryAction {none|restart|revert}

(Optional) Specifies the action to perform on the VM after the Hyper-V host restarts after an unexpected shutdown. Options include the following:

none

Performs no action.

restart

Restarts the VM.

revert

Returns to the last snapshot.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Properties of a VM

This example changes the number of assigned CPU cores to two.

```
dpmhyperv -setVMProperties -ws_user dcaadmin -ws_password #test# -hypervHost hserver
-vm TestVM -cpus 2
```

dpmhyperv SetVMVirtFloppy Command--Assign a Virtual Floppy Image to a VM Floppy Drive

The dpmhyperv SetVMVirtFloppy command assigns a virtual floppy image to a VM floppy drive on the Hyper-V host.

This command has the following format:

```
dpmhyperv SetVMVirtFloppy
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password] -hypervHost hostname
-vm vm_name|-vguid vm_ID
[-path location]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-path *location*

Defines the location of the virtual floppy image.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Assign a Virtual Floppy Image to the VM Floppy Drive on a Hyper-V Host

This example connects the virtual floppy image, "1.vfd" to the VM floppy drive on the Hyper-V host, "hserver."

```
dpmhyperv SetVMVirtFloppy -ws_user admin -ws_password test -hypervHost hserver  
-vm test-vm01 -path C:\data\Floppy\1.vfd
```

dpmhyperv ShowClusterSharedVolumes Command--Show all Cluster Shared Volumes

The dpmhyperv ShowClusterSharedVolumes command lists all cluster shared volumes on a Hyper-V server.

This command has the following format:

```
dpmhyperv ShowClusterSharedVolumes  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all Cluster Shared Volumes on a Hyper-V Host

This example lists all shared volumes on the host, "hserver."

```
dpmhyperv ShowClusterSharedVolumes -ws_user admin -ws_password #test# -hypervHost
hserver
```

dpmhyperv ShowDirectories Command--Show Directory Properties on a Hyper-V Server

The dpmhyperv ShowDirectories command lists all the properties of a directory on a Hyper-V host.

This command has the following format:

```
dpmhyperv ShowDirectories
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-directory directory_path
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-directory *location*

Defines the location of the directory on the Hyper-V host.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show Properties of a Directory on a Hyper-V Host

This example lists all the properties of the directory, "c:\test" on the host, "hserver."

```
dpmhyperv ShowDirectories -ws_user admin -ws_password #test# -hypervHost hserver  
-directory c:\test
```

dpmhyperv ShowLogicalDisks Command--Show all Logical Disks

The dpmhyperv ShowLogicalDisks command lists all logical disks and its properties on a Hyper-V host.

This command has the following format:

```
dpmhyperv ShowLogicalDisks  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all Logical Disks on a Hyper-V Host

This example lists all logical disks on the Hyper-V host, "hserver."

```
dpmhyperv ShowLogicalDisks -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowPhysDisks Command--Show Physical Disks of a Host

The dpmhyperv ShowPhysDisks command all physical disks on a Hyper-V host.

This command has the following format:

```
dpmhyperv ShowPhysDisks
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
[-prompt]
-hypervHost hostname
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the user ID and password.

Default: Yes

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show Physical Disks of a Host

This example shows all physical disks on the Hyper-V host, "hserver".

```
dpmhyperv ShowPhysDisks -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowSCVMMHardwareProfiles Command--Show all SCVMM Hardware Profiles

The dpmhyperv ShowSCVMMHardwareProfiles command lists all hardware profiles on the SCVMM host.

This command has the following format:

```
dpmhyperv ShowSCVMMHardwareProfiles  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -scvmmHost scvmm_hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-scvmmHost *scvmm_hostname*

(Optional) Specifies the name of the SCVMM server host.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Hardware Profiles on an SCVMM Host

This example shows all hardware profiles on the SCVMM host, "ms01."

```
dpmhyperv ShowSCVMMHardwareProfiles -ws_user admin -ws_password #test# -scvmmHost ms01
```

dpmhyperv ShowSCVMMGuestOSProfiles Command--Show Guest Operating System Profiles

The dpmhyperv ShowSCVMMGuestOSProfiles command lists all profiles of the guest operating system on the SCVMM host.

This command has the following format:

```
dpmhyperv ShowSCVMMGuestOSProfiles
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-scvmmHost scvmm_hostname
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-scvmmHost *scvmm_hostname*

(Optional) Specifies the name of the SCVMM server host.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show Guest Operating System Profiles

This example lists all guest operating system profiles on the SCVMM host, "ms01."

```
dpmhyperv ShowSCVMMGuestOSProfiles -ws_user admin -ws_password #test# -scvmmHost ms01
```

dpmhyperv ShowTemplates Command--Show all VM Templates

The dpmhyperv ShowTemplates command lists all VM templates on a Hyper-V host.

This command has the following format:

```
dpmhyperv ShowTemplates  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-filter condition]  
[-scvmmHost scvmm_hostname]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-filter *condition*

(Optional) Specifies a filter condition such as "*".

-scvmmHost *scvmm_hostname*

(Optional) Specifies the name of the SCVMM server host.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all VM Templates

This example lists all the VM templates on the host, "hserver."

```
dpmhyperv ShowTemplates -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowVirtFloppy Command--Show the List of Virtual Floppy Drives

The dpmhyperv ShowVirtFloppy command lists all virtual floppy disks assigned to a VM.

This command has the following format:

```
dpmhyperv GetVMState
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm vmname| -vmguid vm_ID
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all Floppy Disks

This example lists all the virtual floppy disks assigned to the VM having the GUID, "38900009000000yy."

```
dpmhyperv ShowVirtFloppy -ws_user admin -ws_password #test# -hypervHost hserver  
-vmguid 38900009000000yy
```

dpmhyperv ShowVMDisks Command--Show all Virtual Disks

The dpmhyperv ShowVMDisks command lists all virtual disks in a VM on a Hyper-V host.

This command has the following format:

```
dpmhyperv ShowVMDisks  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname|-vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all Virtual Disks on a Hyper-V Host

This example lists all virtual disks with its properties in the VM, "vm01" on the host "hserver."

```
dpmhyperv ShowVMDisks -ws_user admin -ws_password #test# -hypervHost hserver -vm test-vm01
```

dpmhyperv ShowVMNics Command--Show all Network Interface Cards

The dpmhyperv ShowVMNics command lists all network interface cards in a VM.

This command has the following format:

```
dpmhyperv ShowVMNics  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname| -vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all Network Interface Cards

This example lists all network interface cards with its properties in the VM, "vm01" on the host, "hserver."

```
dpmhyperv ShowVMDisks -ws_user admin -ws_password #test# -hypervHost hserver -vm
test-vm01
```

dpmhyperv ShowVMs Command--Show all VMs

The dpmhyperv ShowVMs command lists all VMs on a Hyper-V host. The command gets VM names and their corresponding GUIDs.

This command has the following format:

```
dpmhyperv ShowVMs
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all VMs

This example lists all VMs and their corresponding GUIDs on the host, "hserver."

```
dpmhyperv ShowVMs -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowVMSCSIControllers Command--Show all SCSI Controllers

The dpmhyperv ShowVMSCSIControllers command lists all SCSI controllers in a VM.

This command has the following format:

```
dpmhyperv ShowVMSCSIControllers  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname | -vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-vm *vm_name*

Specifies the VM name.

-vmguid *vm_ID*

Specifies the unique GUID of the VM.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Show all SCSI Controllers on the SCVMM host

This example lists all the SCSI controllers with their properties in the virtual machine, "vm01" on the host, "hserver."

```
dpmhyperv ShowVMSCSIControllers -ws_user admin -ws_password #test# -hypervHost
hserver -vm vm01
```

dpmhyperv ValidateAgentSNMPAccess Command--Validate SNMP Access to an Agent

The dpmhyperv ValidateAgentSNMPAccess command validates whether an agent can be accessed using SNMP community string.

This command has the following format:

```
dpmhyperv ValidateAgentSNMPAccess
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-snmAccessName snmp_name -checkWriteAccess write_access
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-host *hostname*

Specifies the host name of the Hyper-V server.

-snmpAccessName *snmp_name*

Specifies the SNMP community string.

-checkWriteAccess *write_access*

Specifies whether to select write access.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Validate SNMP Access to an Agent

This example validates SNMP access to the host, "hserver."

```
dpmhyperv ValidateAgentsSNMPAccess -hypervHost hserver -snmpAccessName public  
-checkWriteAccess -ws_user admin -ws_password #test#
```

dpmhyperv ValidateServerAccess Command--Check Server Access

The dpmhyperv ValidateServerAccess command checks whether you can access a host server.

This command has the following format:

```
dpmhyperv ValidateServerAccess  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-user username]  
[-pass password]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-user *username*

(Optional) Specifies the name of the user.

-pass *password*

(Optional) Specifies the password of the user.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Verify Server Access

This example verifies whether you can access the Hyper-V host, "hserver."

```
dpmhyperv ValidateServerAccess -ws_user admin -ws_password #test# -hypervHost
hserver
```

dpmhyperv DeleteTemplate Command--Deletes a Template

The dpmhyperv DeleteTemplate command deletes a template on a Hyper-V host.

This command has the following format:

```
dpmhyperv Deletetemplate
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
[-prompt]
-hypervHost hostname
-template template_name
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

-ws_encrypted_password

(Optional) Indicates whether the CA Server Automation password is encrypted.

-hypervHost *hostname*

Specifies the name of the Hyper-V host.

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the user ID and password.

Default: Yes

-template *template_name*

Specifies the template name to delete.

-pre

(Optional) Generates an event before the operation is complete.

-post

(Optional) Generates an event after the operation is complete.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deletes a Template

This example deletes the template, "template_5" on the Hyper-V host, "hserver".

```
dpmhyperv ShowPhysDisks -ws_user admin -ws_password #test# -hypervHost hserver  
-template template_5
```

CA IBM LPAR CLI Commands

You can use the CLI to script and automate CA IBM LPAR commands and run actions based on the command results. Corresponding commands are also available in the AutoShell.

dpmlpar cycle Command--Cycle a Logical Partition

The dpmlpar cycle command powers on, powers off, resets, or suspends a logical partition.

This command has the following format:

```
dpmlpar cycle
[-sc sc_url]
-powerop {activate|restart|shutdown}
-hmc name
-managed_system managementsystemname
-partition_name partitionname
[-type {delayed|immediate|os_shutdown|immediate_os_shutdown}]
[-profile_name profilename]
[-activate_bootmode {normal|open_firmware}]
[-activate_keylock {normal|manual}]
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-powerop {activate|restart|shutdown}

Specifies the power operation to perform on the LPAR. Options include the following:

activate

Turns on the LPAR.

restart

Turns off the LPAR, if necessary, and then turns it on.

shutdown

Turns off the LPAR.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {immediate|os_shutdown|immediate_os_shutdown}

Specifies to use the imaging operation type resource group (*res_group*) or individual resources (*individual_res*). Options include the following:

immediate

Shuts down the partition immediately. This option can cause undesirable results if the data has only been partially updated.

os_shutdown

Shuts down the partition by issuing the command for a typical shutdown. The partition must be imaged for this option to succeed.

immediate_os_shutdown

Shuts down the partition by issuing the operating system command to shut down the system as soon as possible. This command bypasses typical shutdown activities including sending messages to other users. The partition must be imaged for this option to succeed.

-profile_name *lparprofile* (HMC only)

[-profile_name *lparname*] (IVM only)

Specifies the partition profile to use when you activate the logical partition. Required for HMC. Optional for IVM, and if specified the name must match the partition name.

-activate_bootmode {normal|open_firmware}

(Optional) Specifies the keylock mode for the activate operation. Options include the following:

normal

Starts the partition in the typical manner.

open_firmware

Starts the partition and opens the open firmware prompt.

-activate_keylock {normal|manual|bypass}

(Optional) Specifies the keylock mode for the activate operation. Options include the following:

normal

Starts the partition in unattended mode and requires no user interaction during activation.

manual

Starts the partition in attended mode and requires user interaction during activation.

bypass

Does not activate the keylock mode.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user username

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password password

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value***Example: Activate a Logical Partition with the Default Profile**

This example activates the logical partition, "testlpar," using the default profile.

```
dpmlpar cycle -powerop activate -hmc uslihmc  
-managed_system testComputer -partition_name testlpar
```

Example: Activate a Logical Partition with a Specified Profile

This example activates the logical partition, "testlpar," using the profile testlparprofile.

```
dpmlpar cycle -powerop activate -hmc usliivm  
-managed_system testComputer -partition_name testlpar  
-profile_name testlparprofile
```

Example: Activate a Logical Partition to Open Firmware Prompt

This example activates the logical partition "testlpar" using the profile testlparprofile and opens to the open firmware prompt.

```
dpmlpar cycle -powerop activate -hmc uslihmc  
-managed_system testMS -partition_name testlpar  
-activate_bootmode open_firmware
```

Example: Delayed Shut Down of a Logical Partition

This example performs a delayed shutdown of the logical partition, "testlpar."

```
dpmlpar cycle -hmc testivm -powerop shutdown  
-managed_system managedsys1 -partition testlpar -type delayed
```


dpmlpar delete Command--Delete a Logical Partition

The dpmlpar delete command deletes a logical partition that is deactivated.

Important! Verify that you back up any important data *before* you issue this command. This command deletes the LPAR and the data files for the LPAR. The LPAR must be powered off when you issue this command.

This command has the following format:

```
dpmlpar delete
[-sc sc_url]
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <false|true>]
[-ws_remote_user username]
[-ws_remote_password password]
[-delete_logical_value <false|true>]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the name of a logical partition to delete. This partition must exist on the managed system.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value*

Example: Delete a Logical Partition

This example deletes the logical partition, "lpar01."

```
dpm\lpar delete -hmc hmc02 -managed_system system05
-partition_name lpar01
```

dpmlpar getresources Command--Get LPAR Resources

The dpmlpar getresources command retrieves memory units, processor units, or processors for IBM AIX LPARs.

This command has the following format:

```
dpmlpar getresources
[-sc sc_url]
-resource {all|memory|processors|processor_units}
-hmc name
-managed_system managementsystemname
-partition_name partitionname
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <false|true>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-resource {all|memory|processors|processor_units}

Specifies whether a specific resource is retrieved or all resources. Options include the following:

all

Retrieves and displays all partition resources.

memory

Retrieves and displays only the memory resources for the partition.

processors

Retrieves and displays only the processor resources for the partition.

processor units

Retrieves and displays only the processor unit resources for the partition.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition for which resources are being listed. The partition must exist on the managed system.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value*

Example: Get all Resources for an LPAR

This example retrieves all resources for lpar06.

```
dpm\lpar getresources -resource all -partition_name lpar06  
-hmc ivm_02 -managed_system abcsystem4
```

Example: Get the Memory Resources for an LPAR

This example retrieves the memory resources for lpartest01.

```
dpmlpar getresources -resource memory -partition_name lpartest01
-hmc hmc_02 -managed_system abcsystem2
```

dpmlpar image Command--Create an IBM AIX LPAR

The dpmlpar image command creates an IBM AIX LPAR using the HMC/IVM and a target IBM AIX managed system. (Optional) You can deploy the IBM AIX operating system after you create the LPAR.

This command has the following format:

```
dpmlpar image [-sc sc_url]
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-profile_name lparprofile (HMC only)
[-profile_name lparname] (IVM only)

-min_mem size
-desired_mem size
-max_mem size

-proc_mode {shared|dedicated}
-min_proc_units units
-desired_proc_units units
-max_proc_units units

-min_procs number
-desired_procs number
-max_procs number

-max_virtual_slots number
-share_mode {capped|uncapped}
[-uncap_weight weight]
[-io_slot DRC index, {true|false}]
[-lpar_io_pool_id id {id, ...}]
```

```
[-virtual_serial_adapter slot_num, remote_lpar_name, remote_slot_num, is_required]
[-virtual_scsi_client_adapter slot_num, remote_lpar_name, remote_slot_num,
is_required] |
[-virtual_scsi_server_client_adapter virtual_io_server_name, backing_device_name,
client_slot_num, is_required]
[-virtual_eth_adapter slot_num, is_IEEE, port_vlan_id, (additional_vlan_id,
additional_vlan_id, ...), trunk_priority, is_required]
[-virtual_fc_server_client_adapter virtual_io_server_name, physical_fc_port_name,
client_slot_num, is_required
[virtual_io_server_name,physical_fc_port_name,client_slot_num,is_required;...]]

[-pre] [-post]
[-ws_user username]
[-ws_password password]

[-encrypted_password {no|yes}]
[-prompt <no|yes>]

[-ws_remote_user username]
[-ws_remote_password password]

[-create_logical_volume <false|true>]
[-logical_volume_size <value>]
[-volume_group_names <value>]
[-logical_volume_name <value>]
[-use_logical_volume_name_as_prefix {no|yes}]
[-default_vio_server_name vioserver]
[-itcm_server itcmservername]
[-locale iso639value]
```

Parameters for imaging AIX using the NIM MKSYSB utility

```
[-install_type {rte|mksysb}]

-mksysb mksysb_resource

-lpp lpp_resource

-bosinst_data bos_install_data_resource

-image_data image_data_resource
```

Parameters for provisioning AIX using a resource group

```

-provision_aix true
-type res_group
-res_group_name resourcegroupname
-machine_res_name machineresourcenam
-target_username targetusername
[-target_password targetpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
-nim_master_host_name nimmasterhostname
[-scalability_server servername]
[-deploy_template templatename]
-auto_deploy {yes|no}
[-wait [timeout]]

```

Parameters for provisioning AIX using individual resources

```

-provision_aix true
-type individual_res
-machine_res_name machineresourcenam
-lpp lppresource
-spot spotresource
-bosinst_data bosdata
-resolv_conf resolveconf
-fb_script fbscript
-post_inst_scripts script1,script2,script3
-target_username targetusername
[-target_password targetpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
-nim_master_host_name nimmasterhostname
[-scalability_server servername]
[-deploy_template templatename]
-auto_deploy {yes|no}
[-wait [timeout]]

```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the name of a logical partition to create.

-profile_name *lparprofile* (HMC only)

[-profile_name *lparname*] (IVM only)

Specifies the partition profile to use when you activate the logical partition. Required for HMC. Optional for IVM, and if specified the name must match the partition name.

-min_mem *size*

Defines the minimum amount of memory for the partition.

-desired_mem *size*

Defines the desired amount of memory for the partition.

-max_mem *size*

Defines the maximum amount of memory for the partition.

-proc_mode {*shared* | *dedicated*}

Specifies the type of processor mode for the partition. Options include the following:

shared

Shares processor resources with other partitions.

dedicated

Specifies that the partition has dedicated processor resources.

-min_proc_units *units*

Defines the minimum number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-desired_proc_units *units*

Defines the assigned number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-max_proc_units *units*

Defines the maximum number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-min_procs *number*

Defines the minimum number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-desired_procs *number*

Defines the assigned number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-max_procs *number*

Defines the maximum number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-shared_mode {*capped|uncapped*}

Specifies whether the managed system allows the logical partition to use idle processing units that are not committed to another partition from the shared processor pool.

Note: This option can only be used with shared processors.

Options include the following:

capped

Specifies that the partition can only use the number of processing units that are committed to it.

uncapped

Specifies that the partition can use idle processor units from the shared processor pool when they are available.

-uncap_weight *weight*

(Optional) Defines a weighted average of processing priority when you select uncapped sharing mode.

Limits: 0 - 255

-max_virtual_slots *number*

Defines the maximum number of virtual adapters for this partition.

Default: 2

Limits: 2 - 65,536

-io_slot *DRC-Index*,{true|false}

(Optional) Specifies the I/O slot for a physical component. You can specify this value multiple times for different I/O devices.

DRC-Index

Specifies the slot dynamic reconfiguration connector (DRC) index.

true

Specifies that a DRC index is required for the I/O slot.

false

Specifies that a DRC index is not required for the I/O slot.

-io_pool_id *id* {*id* ...}

(Optional) Defines the group of I/O adapters that can be taken over and used by any of a specified group of logical partitions without any active intervention from the HMC. The group of partitions can be a comma-separated list of I/O pool IDs.

-virtual_serial_adapter *slot_num*, *remote_lpar_name*, *remote_slot_num*, *is_required*

(Optional) Defines the virtual serial adapters for this logical partition. The variable *is_required* can be set to true or false. Only client serial adapter is supported, but can be specified multiple times.

Default: Two server serial adapters created in slots 1 and 2.

Note: Not supported for creating an LPAR on IVM.

-virtual_eth_adapter_client *slot_num*, *is_IEEE*, *port_vlan_id1* (*vlanid2*,*vlanid3*,...), *trunk_priority*, *is_required*

(Optional) Defines the virtual ethernet adapters for this logical partition and can be specified multiple times. Optional values are *vlan_idn* and *trunk_priority*. The variables *is_IEEE* and *is_required* can be set to true or false. Additional *vlan_ids* must be comma-separated and enclosed in parentheses.

Note: When creating an LPAR on IVM, *slot_num* must be greater than or equal to 4.

Example: 4,false,1,,,true

-virtual_fc_server_client_adapter *virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required* [*virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required*;...]

(Optional) Defines a virtual Fibre Channel (FC) client adapter, FC server adapter on VIO server, and associates the FC server adapter to the physical FC port. This option is not valid when the `-provision_aix` option is set to true. This parameter can be specified multiple times. The variable `is_required` can be set to true or false.

-virtual_scsi_client_adapter *slot_num, remote_lpar_name, remote_slot_num, is_required*

(Optional) Defines the virtual client SCSI adapters for this logical partition and can be specified multiple times. The variable `is_required` can be set to true or false.

-virtual_scsi_server_client_adapter *virtual_io_server_name, backing_device_name, client_slot_num, is_required*

(Optional) Defines a virtual SCSI server device and virtual SCSI client device, which is automatically linked after the server device is created and the server slot number is identified. This parameter can be specified multiple times. The variable `is_required` can be set to true or false.

Example: CUST-VIOSERVER,hdisk5,3,true

Note: If you create Logical Volume using the `dpmlpar` image command and set the `-create_logical_volume` option to 'yes', the `-virtual_scsi_server_client_adapter` option is not required and shall be omitted.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* **-ws_password** *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

-encrypted_password {no|yes}

(Optional) Specifies whether the user password is encrypted.

Default: No

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user username

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password password

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-create_logical_volume {no|yes}

(Optional) Specifies whether to create Logical Volume.

Default: no

Note: If you create Logical Volume using the `dpmlpar image` command and set the `-create_logical_volume` option to 'yes', the `-virtual_scsi_server_client_adapter` option is not required and shall be omitted.

-logical_volume_size *logical_volume_size*

(Optional) Defines the size of the Logical Volume in MB.

-volume_group_names *volume_group_1, volume_group_2, volume_group_3*

(Optional) Specifies the Logical Volume group names as a comma-separated list.

-logical_volume_name *logical_volume_name*

(Optional) Defines the name of the Logical Volume.

-use_logical_volume_name_as_prefix {no|yes}

(Optional) Specifies whether to use the Logical Volume Name as a prefix.

Default: no

-default_vio_server_name *vioserver*

(Optional) Specifies the VIO server that is used for creation of Logical Volumes.

-itcm_server *itcm_servername*

(Optional) Specifies the name of the CA ITCM Server.

Parameters used for imaging with NIM mksysb

-install_type {rte | mksysb}

(Optional) NIM installation type. Accepted values are `rte` or `mksysb`. `rte` is the default if `-install_type` is not specified.

-mksysb *mksysbResource*

`mksysb` resource. Only valid if `-install_type` is `mksysb`.

-lpp *lpp_resource*

Specifies the licensed program product files to use for an imaging request. Optional if `-install_type` is `mksysb`. Required if `-install_type` is `rte`.

-bosinst_data *bos_install_data_resource*

Specifies a file that contains information for the base operating system (BOS) installation program. Optional if `-install_type` is `mksysb`. Required if `-install_type` is `rte`.

-image_data *imageDataResource*

(Optional) Specifies the image data resource file that describes how physical disks, volume groups, logical volumes, file systems, and paging space are configured on the root volume.

Parameters used for provisioning

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-auto_deploy {yes | no}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-bosinst_data *bos_install_data_resource*

Specifies a file that contains information for the base operating system (BOS) installation program. Optional if `-install_type` is `mksysb`. Required if `-install_type` is `rte`.

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-fb_script *fbscript*

(Optional) Defines the name of the file to use to configure devices when a NIM client is initially booting after the BOS installation process is complete.

-lpp *lpp_resource*

Specifies the licensed program product files to use for an imaging request. Optional if `-install_type` is `mksysb`. Required if `-install_type` is `rte`.

-machine_res_name *machineresourcenname*

Defines the name of the machine resource. This name must be predefined at the NIM Master.

-nim_master_host_name *nimmasterhostname*

Defines the NIM master host name to perform the image deployment.

-post_inst_scripts *script1,script2,script3*

(Optional) Specifies a comma-separated list of scripts to run after installation.

-provision_aix {true|false}

Indicates whether the partition is imaged using NIM, after you create it.

true

Uses NIM to image the partition that you created. If you set to `true`, the NIM resource group or NIM individual resource parameters are used. See the `dpmnim image Command|Deploy an IBM AIX Image Using a Resource Group` and `dpmnim image Command|Deploy an IBM AIX Image Using an Individual Resource`. The job ID is returned when NIM provisioning starts.

false

Does not use NIM to image the partition that you created. No job ID is returned.

-res_group_name *resource group name*

Defines the name of the resource group.

-resolve_conf *resolveconf*

(Optional) Defines a file that contains valid */etc/resolv.conf* entries that define Domain Name Protocol name-server information for local resolver routines.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-spot *spotResource [mksysbResource]*

Defines the shared product object tree to use for an imaging request.

-target_password *targetpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying the image. If you do not specify the password, it is retrieved from the authorization file.

Note: Use the *dpmutil* CLI to set up the authorization file.

-target_username *targetusername*

Defines the user name used for deploying agents to the target host server to which you are deploying the image.

-type {*res_group* | *individual_res*}

Specifies to use the imaging operation type resource group or individual resources.

res_group

Specifies to use the resource group for imaging.

individual_res

Specifies to use individual resources for imaging.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the *caimgconf.cfg* file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

Example: Create an IBM AIX Logical Partition

This example creates a logical partition, but does not provision it.

```
dpmlpar image -hmc ivm01 -managed_system testMS
-partition_name lpartest01 -profile_name lpartest01
-max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5
-proc_mode shared -share_mode uncapped -uncap_weight 123
-min_proc_units .5 -max_proc_units 3.25 -desired_proc_units 1.75
-virtual_serial_adapter 2,serial_partition,1,true
-virtual_scsi_client_adapter 3,scsi_partition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true
-io_slot 21020003,false
```

Example: Create an IBM AIX Logical Partition and Image it Using Individual Resources

This example creates a logical partition and provisions it using individual resources.

```
dpmlpar image -hmc hmc01 -managed_system testMS
-partition_name IRTEST -profile_name Default
-max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 1
-max_procs 1 -proc_mode dedicated
-virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),0,true
-io_slot 21020003,false -provision_aix true
-type individual_res -lpp 530lpp_res -spot 530spot_res
-bosinst_data 530_bid_ow -resolv_conf master_net_conf
-post_inst_scripts piScript -machine_res_name MachineName
-nim_master_host_name machine.mydomain.com -auto_deploy no
-target_username root -target_password password
```

Example: Create an IBM AIX Logical Partition and Image it Using a Resource Group

This example creates a logical partition and provisions it using a resource group.

```
dpmlpar image -hmc hmc02 -managed_system testMS
-partition_name RGTEST -profile_name Default
-max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5
-proc_mode dedicated -virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true
-io_slot 21020003,false -provision_aix true -type res_group
-res_group_name Res_grp -machine_res_name Machine
-nim_master_host_name machine.mydomain.com -auto_deploy no
-target_username root -target_password password
```


dpmlpar imgjobcheck Command--Retrieve Status of IBM AIX Imaging Job

The dpmlpar imgjobcheck command retrieves the status of the IBM AIX logical partition imaging job for a specific CA Server Automation job ID.

This command has the following format:

```
dpmlpar imgjobcheck
[-sc sc_url]
-status jobID
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-status *jobID*

Specifies the CA Server Automation job ID used to obtain the job status.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user username

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password password

(Optional) Specifies the CA Server Automation remote password.

-locale iso639value

Example: Retrieve the Status of the IBM AIX Imaging Job Using the Job ID

This example obtains the job status of the IBM AIX logical partition imaging job using the CA Server Automation job ID 42.

```
dpmlpar imgjobcheck -status 42
```

dpmlpar list Command--List HMC Resources

The dpmlpar list command lists information from the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM).

This command has the following format:

```
dpmlpar list [-sc sc_url]
-hmc name
[-managed_system managedsystemname]
[-partition_name partitionname]
-display {managed_systems | managed_system_details | partitions | io_components |
backing_devices | scsi_adapters | partition_details | profiles | wwpns}
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-display

{managed_systems|managed_system_details|partitions|io_components|backing_devices|scsi_adapters|partition_details|profiles|wwpns}

Specifies which resources to list. Options include the following:

managed_systems

Lists the managed systems controlled by the management hardware console.

managed_system_details

Provides detailed information about a managed system.

partitions

Lists the partitions on a managed system.

io_components

Lists the I/O components of a managed system.

backing_devices

Lists the devices that are available to be attached as backing devices to the Server Virtual SCSI Adapter.

scsi_adapters

Lists the SCSI adapters of a managed system.

partition_details

Lists the details for a partition.

profiles

Lists the list of profiles for a partition.

wwpns

Displays the worldwide port names for a partition.

-hmc *name*

Specifies the HMC/IVM for which to list information.

-managed_system *managedsystemname*

(Optional) Specifies the managed system on which the logical partition resides for which you want to list information.

Note: Do not use with managed_systems option.

-partition_name *partitionname*

(Optional) Specifies the logical partition for which you want to list information.

Note: Do not use with managed_systems option.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value*

Example: Display the List of Managed Systems

This example displays the managed systems for the hardware management console "labhmc."

```
dplmpar list -display managed_systems -hmc labhmc
```

Example: Display the SCSI Adapters

This example displays the SCSI adapters for the managed system "testMS."

```
dplmpar list -display scsi_adapters -hmc uslihmc  
-managed_system testMS
```

Example: Display Partition Details

This example displays the details for the logical partition "testlpar."

```
dpmlpar list -display partition_details -hmc uslihmc
-managed_system testMS -partition_name testlpar
```

dpmlpar setresources Command--Add Memory Resources

The dpmlpar setresources command adjusts memory units for IBM AIX LPARs.

This command has the following format:

```
dpmlpar setresources
[-sc sc_url]
-add_memory value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-add_memory *value*

Defines the amount of memory to add to the partition.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic*|*all*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

-prompt {*yes*|*no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password password

(Optional) Specifies the CA Server Automation remote password.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native". Example: Add Memory to a Logical Partition

This example adds 128 MB of memory to the logical partition "lptest01."

```
dpmlpar setresources -add_memory 128 -partition_name lptest01
-hmc uni02 -managed_system usil01system1 -type dynamic
```

Example: Add Memory to a Logical Partition and Update a Profile

This example adds 128 MB of memory to the logical partition "lptest01" and updates the partition profile file.

```
dpmlpar setresources -add_memory 128 -partition_name lptest01
-hmc uni02 -managed_system usil01system1 -type all
-profile_name lptest01
```

dpmlpar setresources Command--Add Processor Resources

The dpmlpar setresources command adds processors and processor units for IBM AIX LPARs.

This command has the following format:

```
dpmlpar setresources
[-sc sc_url]
{-add_processors|add_processor_units} value
-hmc name
-managed_system managementsystemname
-partition_name partitionname
-type {dynamic|all}
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-add_processors *value*

Defines the number of processors to add to the partition.

-add_processor_units *value*

Defines the processor units to add to the partition.

Limits: .01 increments

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic*|*all*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value***-prompt {*yes|no*}**

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value***Example: Add Processors for a Partition and Update Profile**

This example adds a processor to the partition "lpartest01" and updates the partition profile.

```
dpm_lpar setresources -add_processors 1 -partition_name lpartest01
-hmc uni02 -managed_system MSystem1 -type all -profile_name lpartest01
```

Example: Add Processors and Processor Units for a Partition

This example temporarily adds the processor and processor unit resources for the partition "lpartest01."

```
dpm_lpar setresources -add_processors 1 -add_processor_units 4.5
-partition_name lpartest01 -hmc uni02 -managed_system MSystem1
-type dynamic
```

dpmlpar setresources Command--Subtract Memory Resources

The dpmlpar setresources command removes memory units for IBM AIX LPARs.

This command has the following format:

```
dpmlpar setresources
[-sc sc_url]
-subtract_memory value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subtract_memory *value*

Defines the amount of memory to remove from the partition.

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic* | *all*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

-prompt {*yes* | *no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value*

Example: Subtract Memory from a Logical Partition

This example subtracts 128 MB of memory from the logical partition "lpartest01."

```
dpmlpar setresources -subtract_memory 128 -partition_name lpartest01  
-hmc con02 -managed_system MSsystem1 -type dynamic
```

Example: Subtract Memory from a Logical Partition and Update a Profile

This example subtracts 128 MB of memory from the logical partition "lpartest01" and updates the partition profile file.

```
dpmlpar setresources -subtract_memory 128 -partition_name lpartest01  
-hmc con02 -managed_system MSsystem1 -type all
```

dpmlpar setresources Command--Subtract Processor Resources

The dpmlpar setresources command removes processors and processor units from IBM AIX LPARs.

This command has the following format:

```
dpmlpar setresources  
[-sc sc_url]  
{-subtract_processors|subtract_processor_units} value  
-hmc name  
-managed_system managedsystemname  
-partition_name partitionname  
-type {dynamic|all}  
[-pre] [-post]  
[-ws_user username]  
[-ws_password password]  
[-prompt <no|yes>]  
[-ws_remote_user username]  
[-ws_remote_password password]  
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subtract_processors *value*

Defines the number of processors to remove from the partition.

-subtract_processor_units *value*

Defines the processor units to remove from the partition.

Limits: .01 increments

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic*|*all*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the current profile value. The adjusted values are saved for the partition.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_remote_user *username*

(Optional) Specifies the CA Server Automation remote user name.

-ws_remote_password *password*

(Optional) Specifies the CA Server Automation remote password.

-locale *iso639value*

Example: Subtract Processors From a Partition and Update Profile

This example subtracts a processor from the partition "lpartest01" and updates the partition profile

```
dpm_lpar setresources -subtract_processors 1 -partition_name lpartest01  
-hmc hmc02 -managed_system MSystem1 -type all
```

Example: Subtract Processors and Processor Units From a Partition

This example temporarily adjusts the processor and processor unit resources for the partition "lpartest01."

```
dpm_lpar setresources -subtract_processors 1 -subtract_processor_units 4.5  
-partition_name lpartest01 -hmc hmc02 -managed_system MSystem1 -type dynamic
```

dpmlpar setresources Command--Update Memory Resources

The dpmlpar setresources command adjusts memory units for IBM AIX LPARs.

This command has the following format:

```
dpmlpar setresources
[-sc sc_url]
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all|profile}
-min_mem size
-desired_mem size
-max_mem size
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {dynamic|all|profile}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the profile value. The adjusted values are saved for the partition.

profile

Adjusts the profile value.

-min_mem size

Defines the minimum amount of memory for the partition.

-desired_mem size

Defines the desired amount of memory for the partition.

-max_mem size

Defines the maximum amount of memory for the partition.

dpmlpar setresources Command--Update Processor Resources

The dpmlpar setresources command updates processors for IBM AIX LPARs.

This command has the following format:

```
dpmlpar setresources
[-sc sc_url]
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all|profile}
[-proc_mode {shared |dedicated}]
[-min_proc_units units]
-desired_proc_units units
[-max_proc_units units]
[-min_procs number]
-desired_procs number
[-max_procs number]
[-share_mode {capped |uncapped} [-uncap_weight weight]]
```


-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-type {*dynamic*|*all*|*profile*}

Specifies whether the adjustment is temporary or permanent. Options include the following:

dynamic

Adjusts the current resources only and the adjusted values are lost when you shut down the partition.

all

Adjusts the current resources and the profile value. The adjusted values are saved for the partition.

profile

Adjusts the profile value.

-proc_mode {shared|dedicated}

(Optional) Specifies the type of processor mode for the partition. Options include the following:

shared

Shares processor resources with other partitions.

dedicated

Specifies that the partition has dedicated processor resources.

-min_proc_units *units*

(Optional) Defines the minimum number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-desired_proc_units *units*

Defines the assigned number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-max_proc_units *units*

(Optional) Defines the maximum number of processing units for this partition.

Limits: .01 increments

Note: This option can only be used with shared processors.

-min_procs *number*

(Optional) Defines the minimum number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-desired_procs *number*

Defines the assigned number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-max_procs *number*

(Optional) Defines the maximum number of virtual processors for this partition.

Note: This option can only be used with shared processors.

-shared_mode {capped|uncapped}

Specifies whether the managed system allows the logical partition to use idle processing units that are not committed to another partition from the shared processor pool.

Note: This option can only be used with shared processors.

Options include the following:

capped

Specifies that the partition can only use the number of processing units that are committed to it.

uncapped

Specifies that the partition can use idle processor units from the shared processor pool when they are available.

-uncap_weight *weight*

(Optional) Defines a weighted average of processing priority when you select uncapped sharing mode.

Limits: 0 - 255

CA Microsoft Cluster Server CLI Commands

You can use the CLI to script and automate CA MSCS commands and run actions based on the command results. Corresponding commands are also available in the AutoShell.

dpmmscs getclusterinfo Command--Get Information of a Cluster

The `getclusterinfo` command returns general information of a cluster.

This command has the following format:

```
dpmmscs getclusterinfo
[-sc <scUrl>]
-cluster <cluster>
[-locale iso639value]
[-ws_user username]
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Information of a Cluster

```
dpmmscs.exe getclusterinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs getgroupresourcesinfo Command--Get Resource Information of a Cluster group

The `getgroupresourcesinfo` command returns resource information for a cluster group

This command has the following format:

```
dpmmscs getgroupresourcesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
-resgroup <resource group>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-resgroup

Specifies the resource group.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Resource Information of a Cluster group

```
dpmmscs.exe getgroupresourcesinfo -cluster MYCLUSTER -resgroup "Group 0" -ws_user  
admin -ws_password admin
```

dpmmcs getnetworkinterfacesinfo Command--Get Network Interface Information

The getnetworkinterfaces info command returns cluster network interface information.

This command has the following format:

```
dpmmcs getnetworkinterfacesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
-net <network name>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-net

Specifies the network name.

-cluster

Specifies the cluster name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Cluster Network Interface Information.

```
dpmmscs.exe getnetworkinterfacesinfo -cluster MYCLUSTER -net "Local Area Connection 2" -ws_user admin -ws_password admin
```

dpmmscs getnetworksinfo Command--Get Network Information

The getnetworksinfo command returns cluster network information.

This command has the following format:

```
dpmmscs getnetworksinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Cluster Network Information.

```
dpmmscs.exe getnetworksinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmcs getnodesinfo Command--Get Cluster Nodes Information

The getnodesinfo command returns cluster nodes information.

This command has the following format:

```
dpmmscs getnodesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Cluster Nodes Information.

```
dpmmscs.exe getnodesinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```


dpmmcs getresourcegroupsinfo Command--Get Cluster Resource Groups Information

The getresourcegroupsinfo command returns cluster resource groups information.

This command has the following format:

```
dpmmcs getresourcegroupsinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password password]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user username

(Optional) Specifies the CA Server Automation user name.

-ws_password password

(Optional) Specifies the CA Server Automation user password.

Example: Get Network Cluster Resource groups Information.

```
dpmmcs.exe getresourcegroupsinfo -cluster MYCLUSTER -ws_user admin -ws_password  
admin
```

dpmmcs getresourcesdependenciesinfo Command--Get Dependency Information of Resources

The `getresourcesdependenciesinfo` command returns dependency information for a cluster resource.

This command has the following format:

```
dpmmcs getresourcesdependenciesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
-res <resource name>  
-restype <resource type>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-res

Specifies the resource name.

-restype

Specifies the resource type.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Dependency Information of Cluster Resources

```
dpmmscs.exe getresourcedependenciesinfo -cluster MYCLUSTER -res "SQL Server
Fulltext" -restype "Generic Service" -ws_user admin -ws_password admin
```

dpmmscs getresourcetypesinfo Command--Get Cluster Resource Types Information

The getresourcetypes info command returns cluster resource types information.

This command has the following format:

```
dpmmscs getresourcetypesinfo
[-sc <scUrl>]
-cluster <cluster>
[-locale iso639value]
[-ws_user username]
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Cluster Resource Types Information.

```
dpmmscs.exe getresourcetypesinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs getservicesinfo Command--Get Cluster Services Information

The getservicesinfo command returns cluster services information.

This command has the following format:

```
dpmmscs getservicesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password password]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Get Cluster Service Information.

```
dpmmscs.exe getservicesinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs movegroup Command--Move a Resource Group to another Node

The movegroup command lets you move a resource group to another node.

This command has the following format:

```
dpmmscs movegroup  
[ -sc <scUrl> ]  
-cluster <cluster>  
-resgroup <resource group>  
-node <node name>  
[ -locale <locale> ]  
[ -ws_user <username> ]  
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-resgroup

Specifies the resource group.

-node

Specifies the node name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Move a Resource Group to another Node

```
dpmmscs.exe movegroup -cluster MYCLUSTER -resgroup "Group 0" -node ATS-281-W2k3CL  
-ws_user admin -ws_password admin
```

dpmmcs pauseservice Command--Pause a Cluster Service

The pauseservice command lets you pause a cluster service on a node.

This command has the following format:

```
dpmmcs pauseservice  
[ -sc <scUrl> ]  
-cluster <cluster>  
-node <node name>  
[ -locale <locale> ]  
[ -ws_user <username> ]  
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-node

Specifies the node name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Pause a cluster service on a node:

```
dpmmscs.exe pauseservice -cluster MYCLUSTER -node ATS-281-W2k3CL -ws_user admin
-ws_password admin
```

dpmmscs bringgrouponline Command--Bring a Resource Group Online

The bringgrouponline command lets you bring a resource group online.

This command has the following format:

```
dpmmscs bringgrouponline
[ -sc <scUrl> ]
-cluster <cluster>
-resgroup <resource group>
[ -locale <locale> ]
[ -ws_user <username> ]
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-resgroup

Specifies the resource group.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Bring a Resource Group Online

```
dpmmscs.exe bringgrouponline -cluster MYCLUSTER -resgroup "Group 0" -ws_user admin  
-ws_password admin
```

dpmmscs bringresourceonline Command--Bring a Resource Online

The bringresourceonline command lets you bring a resource online.

This command has the following format:

```
dpmmscs bringresourceonline  
[ -sc <scUrl> ]  
-cluster <cluster>  
-res <resource name>  
[ -locale <locale> ]  
[ -ws_user <username> ]  
[ -ws_password <password> ]
```


-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-res

Specifies the resource name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Bring a Resource Online

```
dpmmscs.exe bringresourceonline -cluster MYCLUSTER -res "SQL Server Fulltext"  
-ws_user admin -ws_password admin
```

dpmmscs resumeservice Command--Resume a Cluster Service

The resumeservice command lets you resume a cluster service on a node.

This command has the following format:

```
dpmmscs resumeservice
```

```
[ -sc <scUrl> ]
```

```
-cluster <cluster>
```

```
-node <node name>
```

```
[ -locale <locale> ]
```

```
[ -ws_user <username> ]
```

```
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-node

Specifies the node name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Resume a cluster service on a node

```
dpmmscs.exe resumeservice -cluster MYCLUSTER -node ATS-281-W2k3CL -ws_user admin  
-ws_password admin
```

dpmmscs startservice Command--Start a Cluster Service

The startservice command lets you start a cluster service on a node.

This command has the following format:

```
dpmmscs startservice
```

```
[ -sc <scUrl> ]
```

```
-cluster <cluster>
```

```
-node <node name>
```

```
[ -locale <locale> ]
```

```
[ -ws_user <username> ]
```

```
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-node

Specifies the node name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Start a cluster service on a node

```
dpmmscs.exe startservice -cluster MYCLUSTER -node ATS-281-W2k3CL -ws_user admin  
-ws_password admin
```

dpmmscs stopservice Command--Stop a Cluster Service

The stopservice command lets you stop a cluster service on a node.

This command has the following format:

```
dpmmscs stopservice
```

```
[ -sc <scUrl> ]
```

```
-cluster <cluster>
```

```
-node <node name>
```

```
[ -locale <locale> ]
```

```
[ -ws_user <username> ]
```

```
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cluster

Specifies the cluster name.

-node

Specifies the node name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Stop a cluster service on a node

```
dpmmscs.exe stopservice -cluster MYCLUSTER -node ATS-281-W2K3CL -ws_user admin  
-ws_password admin
```

dpmmscs takegroupoffline Command--Take a Resource Group Offline

The takegroupoffline command lets you take a resource group offline.

This command has the following format:

```
dpmmscs takegroupoffline
[ -sc <scUrl> ]
-cluster <cluster>
-resgroup <resource group>
[ -locale <locale> ]
[ -ws_user <username> ]
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-resgroup

Specifies the resource group.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Take a Resource Group Offline

```
dpmmscs.exe takegroupoffline -cluster MYCLUSTER -resgroup "Group 0" -ws_user admin
-ws_password admin
```

dpmmscs takeresourceoffline Command--Take a Resource Offline

The takeresourceoffline command lets you take a resource offline.

This command has the following format:

```
dpmmscs takeresourceoffline  
[ -sc <scUrl> ]  
-cluster <cluster>  
-res <resource name>  
[ -locale <locale> ]  
[ -ws_user <username> ]  
[ -ws_password <password> ]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-cluster

Specifies the cluster name.

-res

Specifies the resource name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-ws_user *username*

(Optional) Specifies the CA Server Automation user name.

-ws_password *password*

(Optional) Specifies the CA Server Automation user password.

Example: Take a Resource Offline

```
dpmmscs.exe takeresourceoffline -cluster MYCLUSTER -res "SQL Server Fulltext"  
-ws_user admin -ws_password admin
```

CA Solaris Zones CLI Commands

You can use the CLI to script and automate CA Solaris Zones commands and run actions based on the command results. Corresponding commands are also available in the AutoShell.

dpmzone associateproject Command--Associate a Task with a Project

The `associateproject` command associates a task with a project. A task is a collection of processes that represent a set of work over time. Each task is associated with one project.

This command has the following format:

```
dpmzone associateproject
[-sc sc_host]
-host hostname
-name zonename
-proj_name pname
-task_id tid
[-pre]
[-post]
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone that provides the project.

-proj_name *pname*

Specifies the project.

-task_id *tid*

Specifies the ID number of the task.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Associate a task with a project:

```
dpmzone associateproject -host SolarisServer2 -name myzone1 -project_name myproject1  
-task_id 1954
```

dpmzone clonezone Command--Clone a Zone

The clonezone command creates a duplicate zone from an existing zone. The source zone must be halted to start the cloning process. Cloning occurs on the Solaris 10 server and may take some time.

This command has the following format:

```
dpmzone clonezone [-sc sc_host]  
-host hostname  
-name zonename  
-new_name nzonename  
-new_path npath  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone that you want to clone.

-new_name *nzonename*

Defines the new zone name.

-new_path *npath*

Defines the path of the new zone.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Clone a zone on host SolarisServer2.

```
dpmzone clonezone -host SolarisServer2 -name myzone1 -new_name myzone2
-new_path /opt/zones/myzone2
```

dpmzone createandinstallzone Command--Create and Install a Zone

The createandinstallzone command creates and installs a zone with custom parameters on the Solaris host.

This command has the following format:

```
dpmzone createandinstallzone
[-sc sc_host]
-host hostname
-name zonename
-path zonepath
-type type
[-archive_path path]
[-autoboot]
[-if_type name]
[-ip ip]
[-pool_name pool]
[-sched_type sched]
[-phy_mem pmem]
[-swap_mem smem]
[-lock_mem lmem]
[-desc desc]
[-pre]
[-post]
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Defines the name of the new zone.

-path *zonepath*

Defines the path of the new zone.

-type *type*

Specifies the type of the new zone. Options include the following:

native

Creates a non-global zone with a Solaris 10 operating environment for running applications.

whole-root

Creates a whole root zone that does not inherit packages.

branded

Creates a non-global zone that contains a non-native operating environment for running applications.

Default: native

-archive_path *path*

(Optional) Specifies the path of the operating environment installer for a branded zone.

Limits: This argument is required for branded zones, but not required for native zones.

-autoboot

(Optional) Specifies that a zone boots automatically at system boot.

Note: If the zones service is disabled on the server, the zone does not autoboot, regardless of the setting of this property. You can enable the service by using the following command:

```
svcadm enable svc:/system/zones:default
```

-if_type *name*

(Optional) Specifies the type of the network interface used by the zone, for example, eri0.

-ip *ip*

(Optional) Defines the IP address of the zone.

-pool_name *pool*

(Optional) Specifies the name of the pool with which you associate the zone.

-sched_type *shed*

(Optional) Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Valid values include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. ts is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

-phy_mem *pmem*

(Optional) Defines the physical memory that is assigned to the zone. A scale (K, M, G, T) can be applied to the value of this number, for example, 1M is one megabyte.

-swap_mem *smem*

(Optional) Defines the swap memory that is assigned to the zone.

-lock_mem *lmem*

(Optional) Defines the locked memory that is assigned to the zone. Locked memory cannot be paged.

-desc *description*

(Optional) Defines a description for the zone.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmzone createpool Command--Create Resource Pool

The createpool command creates a resource pool on a Solaris 10 host.

This command has the following format:

```
dpmzone createpool  
[-sc sc_host]  
-host hostname  
-pset_name pset  
-cpu_min mincpu  
-cpu_max maxcpu  
-pool_name pool  
-sched_type sched  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-pset_name *pset*

Defines the name of the processor set. Each processor set (grouping of CPUs) can contain zero or more CPUs.

-cpu_min *mincpu*

Defines the minimum number of CPUs in the processor set.

-cpu_max *maxcpu*

Defines the maximum number of CPUs in the processor set.

-pool_name *pool*

Specifies the name of the pool with which you associate the zone.

-sched_type *shed*

Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Options include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. ts is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Create the resource pool POOL1 on host SolarisServer2:

```
dpmzone createpool -host SolarisServer2 -pset_name PSET1 -cpu_min 1 -cpu_max 20  
-pool_name POOL1 -sched_type FSS
```

dpmzone createproject Command--Create a Project

The createproject command creates a project in a zone.

This command has the following format:

```
dpmzone createproject  
[-sc sc_host]  
-host hostname  
-name zonename  
-user_id uid  
[-project_id pid]  
-proj_name pname  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-user_id *uid*

Specify a user.

Example: root

-project_id *pid*

(Optional) Defines the project ID. You can assign the ID or let the system automatically generate one.

-proj_name *pname*

Defines the name of the new project.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Create a project in a zone:

```
dpmzone createproject -host SolarisServer2 -name myzone1 -user_id root
-proj_name myProject
```

dpmzone createzone Command--Create a Zone

The createzone command creates a zone with custom parameters on the Solaris host.

This command has the following format:

```
dpmzone createzone
[-sc sc_host]
-host hostname
-name zonename
-path zonepath
-type type
[-autoboot]
[-if_type name]
[-ip ip]
[-pool_name pool]
[-sched_type sched]
[-phy_mem pmem]
[-swap_mem smem]
[-lock_mem lmem]
[-desc desc]
[-pre]
[-post]
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Defines the name of the new zone.

-path *zonepath*

Defines the path of the new zone.

-type *type*

Specifies the type of the new zone. Options include the following:

native

Creates a non-global zone with a Solaris 10 operating environment for running applications.

whole-root

Creates a whole root zone that does not inherit packages.

branded

Creates a non-global zone that contains a non-native operating environment for running applications.

Default: native

-autoboot

(Optional) Specifies that a zone boots automatically at system boot.

Note: If the zones service is disabled on the server, the zone does not autoboot, regardless of the setting of this property. You can enable the service by using the following command:

```
svcadm enable svc:/system/zones:default
```

-if_type *name*

(Optional) Specifies the type of the network interface used by the zone, for example, eri0.

-ip *ip*

(Optional) Defines the IP address of the zone.

-pool_name *pool*

(Optional) Specifies the name of the pool with which you associate the zone.

-sched_type *shed*

(Optional) Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Valid values include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. ts is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

-phy_mem *pmem*

(Optional) Defines the physical memory that is assigned to the zone. A scale (K, M, G, T) can be applied to the value of this number, for example, 1M is one megabyte.

-swap_mem *smem*

(Optional) Defines the swap memory that is assigned to the zone.

-lock_mem *lmem*

(Optional) Defines the locked memory that is assigned to the zone. Locked memory cannot be paged.

-desc *description*

(Optional) Defines a description for the zone.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Create a zone called WebServer4:

```
dpmzone createzone -host MySolarisServer -name WebServer4 -path /Zones/Apache -type
native
-autoboot -if_type eri0 -ip 192.168.100.100 -pool_name WebServer -sched_type fss
-phy_mem 1024M -swap_mem 2048M
```

dpmzone customcommand Command--Run a Custom Command

The customcommand command lets you run external commands or scripts on the Solaris server.

Note: For information about restricted external commands, see the Custom Commands appendix in this guide.

This command has the following format:

```
dpmzone customcommand  
[-sc sc_host]  
-host hostname  
-cmd command  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-cmd *command*

Specifies the name of the program or script to run.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Run a command on host SolarisServer2:

```
dpmzone customcommand -host SolarisServer2  
-cmd /usr/local/private/scripts/test_connections.pl
```

dpmzone deleteproject Command--Delete a Project

The deleteproject command deletes a project from a zone.

This command has the following format:

```
dpmzone deleteproject  
[-sc sc_host]  
-host hostname  
-name zonename  
-proj_name pname  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone that is associated with the project.

-proj_name *pname*

Specifies the project to delete.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Delete a project:

```
dpmzone deleteproject -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone deletezone Command--Delete a Zone

The deletezone command lets you delete a zone on the Solaris server.

This command has the following format:

```
dpmzone deletezone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to delete.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Delete a zone on host SolarisServer2:

```
dpmzone deletezone -host SolarisServer2 -name myzone
```

dpmzone installzone Command--Install a Zone

The `installzone` command starts the installation command on the Solaris server. The installation process may take some time. Verify the status of the zone with the `showprocesses` command to confirm that the installation process is running.

This command has the following format:

```
dpmzone installzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-archive_path path]  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to install.

-archive_path *path*

(Optional) Specifies the path of the operating environment installer for a branded zone.

Limits: This argument is required for branded zones, but not required for native zones.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Install a Native Zone

This example installs MyZone3 on SolarisServer.

```
dpmzone installzone  
-host SolarisServer  
-name MyZone3
```

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example

Install MyZone4 with Solaris 8 on SolarisServer.

```
dpmzone installzone -host SolarisServer -name MyZone4  
-archive_path /opt/zoneos/branded/Solaris8Installer
```

dpmzone movezone Command--Move a Zone

The movezone command lets you move a zone from one path to a new path.

This command has the following format:

```
dpmzone movezone  
[-sc sc_host]  
-host hostname  
-name zonename  
-new_path path  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to move to a new location.

-new_path *path*

Defines the new path of the zone.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Move a zone on host SolarisServer2:

```
dpmzone movezone -host SolarisServer2 -name myzone1
-new_path /opt/zones/myzone1
```

dpmzone rebootzone Command--Reboot a Zone

The rebootzone command lets you reboot a zone on a Solaris server.

This command has the following format:

```
dpmzone rebootzone
[-sc sc_host]
-host hostname
-name zonename
[-pre]
[-post]
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to reboot.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Reboot a zone on host SolarisServer2:

```
dpmzone rebootzone -host SolarisServer2 -name myzone
```

dpmzone renamezone Command--Rename a Zone

The renamezone command lets you rename a zone on the Solaris 10 server.

This command has the following format:

```
dpmzone renamezone  
[-sc sc_host]  
-host hostname  
-name czonename  
-new_name nzonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-old_name *czonename*

Specifies the current zone name.

-new_name *nzonename*

Defines the new zone name.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Rename a zone on host SolarisServer2:

```
dpmzone renamezone -host SolarisServer2 -name myzone -new_name yourzone
```

dpmzone setpoolscheduler Command--Set the Pool Scheduler

The setpoolscheduler command sets the type of scheduler to use for the resource pool. Resource pools can have two scheduler types, the Fair Share Scheduler (FSS) and the Time Share Scheduler (TS).

This command has the following format:

```
dpmzone setpoolscheduler  
[-sc sc_host]  
-host hostname  
-pool_name poolname  
-sched_type sched  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-pool_name *poolname*

Specifies an associated group of resources that can be partitioned.

-sched_type *shed*

Specifies the type of scheduler used to allocate CPU time based on shares. Shares are the portion of the system CPU resources allocated to a project. Options include the following:

ts

Specifies the Time Share Scheduler which fairly allots CPU resources to every process and does not concentrate CPU resources on a particular process. ts is the default scheduler for the Solaris operating environment.

fss

Specifies the Fair Share Scheduler which allows you to allocate CPU time based on shares.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set the Pool Scheduler to FSS

This example sets the scheduling for the Resource Pool using the Fair Share Scheduler.

```
dpmzone setpoolscheduler -host SolarisServer -pool_name test_pool  
-sched_type FSS
```

Example: Set the Pool Scheduler to TS

This example sets the scheduling for the Resource Pool using the Time Share Scheduler.

```
dpmzone setpoolscheduler -host SolarisServer -pool_name test_pool  
-sched_type TS
```

dpmzone setprocessorset Command--Set the Maximum Number of CPUs for a Processor Set

The setprocessorset command sets the maximum number of CPUs for a processor set.

This command has the following format:

```
dpmzone setprocessorset  
[-sc sc_host]  
-host hostname  
-processor_set_name psetname  
-cpu_max maxcpu  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-processor_set_name *psetname*

Specifies the processor set. Each pset (grouping of CPUs) can contain zero or more processors.

-cpu_max *maxcpu*

Defines the maximum number of CPUs in the processor set.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Set the maximum number of CPUs for the processor set:

```
dpmzone setprocessorset -host SolarisServer -processor_set_name test_pset -cpu_max 64
```

dpmzone showinterfaces Command--Show Network Interfaces

The showinterfaces command lists all the network interfaces present on the Solaris 10 host.

This command has the following format:

```
dpmzone showinterfaces  
[-sc sc_host]  
-host hostname  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List all network interfaces on host SolarisServer2:

```
dpmzone showinterfaces -host SolarisServer2
```

dpmzone showpools Command--Show Resource Pools

The showpools command lists all the resource pools present on the Solaris host.

This command has the following format:

```
dpmzone showpools  
[-sc sc_host]  
-host hostname  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List all resource pools on host SolarisServer2:

```
dpmzone showpools -host SolarisServer2
```

dpmzone showprojects Command--List the Properties of a Project

The showprojects command lists the properties of a project.

This command has the following format:

```
dpmzone showprojects  
[-sc sc_host]  
-host hostname  
-name zonename  
-proj_name pname  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-proj_name *pname*

Specifies the project that is associated with the zone.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List the properties of a project:

```
dpmzone showprojects -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone showresourceset Command--Show Resource Sets

The showresourceset command lists the resources of a resource pool.

This command has the following format:

```
dpmzone showresourceset  
[-sc sc_host]  
-host hostname  
-pool_name poolname  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-pool_name *poolname*

Specifies an associated group of resources that can be partitioned.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List the resource sets in MyZonesResourcePool:

```
dpmzone showresourceset -host SolarisServer2 -pool_name MyZonesResourcePool
```

dpmzone showtasks Command--List the Tasks of a Project

The showtasks command lists the tasks of a project.

This command has the following format:

```
dpmzone showtasks  
[-sc sc_host]  
-host hostname  
-name zonename  
-proj_name pname  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-proj_name *pname*

Specifies the project that is associated with the zone.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List the tasks of a project:

```
dpmzone showtasks -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone showzoneinterfaces Command--Show Network Interfaces of a Zone

The showzoneinterfaces command lists all the network interfaces of a zone.

This command has the following format:

```
dpmzone showzoneinterfaces  
[-sc sc_host]  
-host hostname  
-name zonename  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List all network interfaces on zone MyZone3 on SolarisServer2:

```
dpmzone showzoneinterfaces -host SolarisServer2 -name MyZone3
```

dpmzone showzoneprocess Command--Show Processes of a Zone

The showzoneprocess command lists all processes on a zone.

This command has the following format:

```
dpmzone showzoneprocess  
[-sc sc_host]  
-host hostname  
-name zonename  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List the processes on zone MyZone3 on SolarisServer2:

```
dpmzone showzoneprocess -host SolarisServer2 -name MyZone3
```

dpmzone showzoneprojects Command--Show Projects of a Zone

The showzoneprojects command lists all the projects on a zone.

This command has the following format:

```
dpmzone showzoneprojects  
[-sc sc_host]  
-host hostname  
-name zonename  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List all projects on zone MyZone3 on SolarisServer2:

```
dpmzone showzoneprojects -host SolarisServer2 -name MyZone3
```

dpmzone showzones Command--Show Zones

The showzones command lists all the zones present on the Solaris host.

This command has the following format:

```
dpmzone showzones  
[-sc sc_host]  
-host hostname  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

List all zones on host SolarisServer2:

```
dpmzone showzones -host SolarisServer2
```

dpmzone startzone Command--Start a Zone

The startzone command lets you start a zone on the Solaris 10 server.

This command has the following format:

```
dpmzone startzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to start.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Start a zone on host SolarisServer2:

```
dpmzone startzone -host SolarisServer2 -name myzone
```

dpmzone stopzone Command--Stop a Zone

The stopzone command lets you stop a zone on the Solaris 10 server.

This command has the following format:

```
dpmzone stopzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to stop.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

Stop a zone on host SolarisServer2:

```
dpmzone stopzone -host SolarisServer2 -name myzone
```

dpmzone uninstallzone Command--Uninstall a Zone

The `uninstallzone` command lets you uninstall a zone on the Solaris 10 server.

This command has the following format:

```
dpmzone uninstallzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(Optional) Specifies the host name of the Service Controller.

-host *hostname*

Specifies the Solaris Zones server. This parameter is case-sensitive.

-name *zonename*

Specifies the zone to uninstall.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example

Uninstall a zone on host SolarisServer2:

```
dpmzone uninstallzone -host SolarisServer2 -name myzone
```

CA VMware vCenter Server CLI Commands

You can use the CLI to script and automate CA VMware vCenter Server commands and run actions based on the command results. Corresponding commands are also available in the AutoShell.

dpmvc addesxhost Command--Add an ESX Host

The dpmvc addesxhost command adds an ESX host to a vCenter Server.

This command has the following format:

```
dpmvc addesxhost
-esx_host_name esxhostname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-esx_host_user esxhostuser
-esx_host_password esxhostpassword
[-management_ip managementip]
[-host_connect_port hostconnectport]
[-folder_name foldername]
[-datacenter_name datacentername]
[-cluster_name clustername]
[-vim_account_user vimaccountuser]
[-vim_account_password vimaccountpassword]
[-locale iso639value]
```

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-esx_host_user *esxhostuser*

(Optional) Specifies the user name to access the ESX host.

-esx_host_password *esxhostpassword*

Specifies the password used to access the ESX host.

-management_ip *managementip*

(Optional) Specifies the management IP.

-host_connect_port *hostconnectport*

(Optional) Specifies the port used by ESX host to communicate with vCenter Server.

-folder_name *foldername*

(Optional) Specifies the name of the folder where you can add the ESX host.

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-cluster_name *clustername*

(Optional) Specifies the name of the cluster where you can add the ESX host.

-vim_account_user *vimaccountuser*

(Optional) Specifies the username of the VIM account.

-vim_account_password *vimaccountpassword*

(Optional) Specifies the password to access the VIM account.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Add an ESX Host

This example adds an ESX host, "esx1" to the vCenter Server "myvcenter."

```
dpmvc addesx -esx_host_name esx1 -vc_server myvcenter -esx_host_user admin
-esx_host_password topsecret
```

dpmvc addvmvdisk Command--Add Virtual Disk

The dpmvc addvmvdisk command adds a virtual disk to a VM.

This command has the following format:

```
dpmvc addvmvdisk
-datastore_name datastorename
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-capacity_in_mb disk_capacity
[-controller_key controllerkey]
[-disk_mode
{append|independent_nonpersistent|independent_persistent|nonpersistent|persistent
|undoable}]
-thin_provisioning {no|yes}
[-unit_number unitnumber]
[-locale iso639value]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-capacity_in_mb *disk_capacity*

Specifies the capacity of the virtual disk in MB.

-controller_key *controllerkey*

(Optional) Specifies the controller key of the disk.

-disk_mode {append | independent_nonpersistent | independent_persistent | nonpersistent | persistent | undoable}

(Optional) Specifies the mode of the disk.

-thin_provisioning {no | yes}

(Optional) Indicates whether the disk must be thin provisioned.

-unit *disk_unitnumber*

(Optional) Specifies the unit number of the disk.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Add a Virtual Disk

This example adds a virtual disk to the VM, "myvm."

```
dpmvc addvmvdisk -datastore_name disk1 -vm_name myvm -vc_server mycenterserver
-capacity_in_mb 20000 -thin_provisioning no
```

dpmvc addvmvnic Command--Add Virtual NIC

The dpmvc addvmvnic command adds a virtual NIC to a VM.

This command has the following format:

```
dpmvc addvmvnic
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-devicetype {e1000 | vmxnet}
-network_name networkname
[-mac_address macaddress]
-wake_on_lan_enabled {no | yes}
[-locale iso639value]
```

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-devicetype {*e1000* | *vmxnet*}

Indicates the type of network device.

-network_name *networkname*

Specifies the virtual network the NIC must be connected to. You can distinguish the names of Standard Switches and Distributed Virtual Switches based on the following naming convention:

- For Standard Switches, the name is the network name.
- For Distributed Virtual Switches, the name is a concatenation of the dvPort group name followed by the Distributed Virtual Switch name enclosed in parentheses: `dvPortGroupName (dvSwitchName)`

-mac_address *macaddress*

(Optional) Specifies the MAC (Media Access Control) address of the network device.

-wake_on_lan_enabled {no | yes}

Indicates whether to start the VM when LAN is enabled.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Add a vNIC

This example adds a virtual NIC to a VM.

```
dpmvc addvmnic -vm_name myvm -vc_server mycenterserver -devicetype vmxnet  
-network_name net1 -wake_on_lan_enabled no
```

dpmvc clone Command--Clone a VM

The dpmvc clone command lets you clone a VM.

This command has the following format:

```
dpmvc clone
[-sc sc_url]
-datacenter_name datacentername
-compute_resource_name name
-esx_host_name esxhostname
-datastore_name datastorename
-vm_name vmname
-resource_pool_name resourcepool
-spec_name specificationname
[-auto_deploy value]
[-deploy_template templatename]
[-network_connection
"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_
gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask[,win_dns_server=wins_dns_
server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_sec
ondary"]]
[-global_dns_search_suffix suffix1[,suffix2,suffix3,...]]
[-linux_domain_name domain_name]
[-linux_dns_servers primary_dns=value,secondary_dns=value,tertiary_dns=value]
[-create_disk MB,datastorename,controller]
[-modify_disk key, datastorename]
[-set_memory number]
[-set_cpu number]
-vm_os_username username
[-vm_os_password password]
[-vc_server vcservername]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID] {-template_name templatename |
-vm_name_cloned_from -vmname}
[-scalability_server scalabilityservername]
[-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-compute_resource_name *name*

Specifies the cluster or VMware ESX host where the VM is created.

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-vm_name *vmname*

Specifies the VM.

-resource_pool_name *resourcepool*

Specifies the name of the resource pool from which you want to select the VM for cloning.

-spec_name *specificationname*

(Optional) Specifies the name of the specification you want to use for the cloned virtual machine.

-auto_deploy {yes|no}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask,win_dns_server=wins_dns_server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_secondary"

Windows:

(Optional) Creates network connections for a VM. This parameter can be specified multiple times. Sequence number is the integer value starting with one (1) that identifies the network connection. The full set of parameters must be enclosed in parentheses if it includes any spaces. All values except nic, network_name, ip_address, def_gateway and subnet_mask; can be reset by entering the name without a value (for example, "nic=123,alt_gateway=").

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask"

Linux:

(Optional) Creates one or more network connections. This parameter can be specified multiple times. The sequence number is the integer value starting with one (1) that identifies the network connection. The full set of parameters must be enclosed in parentheses if it contains any spaces. Only alt_gateway can be reset by entering the name without a value (for example, "nic=123,alt_gateway=").

-global_dns_search_suffix *suffix1{,suffix2,suffix3,...}*

Linux:

Specifies one or more DNS search suffixes.

Windows:

(Optional) Specifies one or more DNS search suffixes.

-linux_domain_name *domain_name*

Linux:

(Optional) Specifies the domain name for a Linux VM.

-linux_dns_servers *primary_dns=value{,secondary_dns=value,tertiary_dns=value}*

Linux:

(Optional) Specifies the IP addresses of DNS servers for a Linux VM. One or more of the values can be omitted (for example, `primary_dns=123.123.123.13,secondary_dns=,tertiary_dns=123.123.123.15`) to unset the value.

-create_disk *MB,datastore_name,controller*

(Optional) Creates one or more additional hard drives. This parameter can be specified multiple times.

MB

Defines the size in megabytes of the hard drive.

Datastore_name

Specifies the name of the data store for this hard drive. Currently only one data store is supported per VM.

Controller

Specifies the controller key for this hard drive.

-modify_disk *{key,datastore_name}*

(Optional) Specifies the data store where the hard drive resides. Can be specified multiple times.

key

Specifies an existing disk.

datastore_name

Specifies the name of the data store that this disk moves to.

-set_memory *number*

(Optional) Defines the amount of memory in megabytes (MB) for the cloned VM. This value overrides the value specified in the template.

-set_cpu *number*

(Optional) Define the number of CPUs for the cloned VM. This value overrides the value specified in the template.

-vm_os_username *user name*

Specifies the user for the cloned VM. This user name is also used for authentication when you auto-deploy the image.

Windows: Must be the user name defined in the customization specification.

Linux: Must be the user name defined in the template.

-vm_os_password *password*

(Optional) Specifies the password for the user for the cloned VM. This password is also used for authentication when you auto-deploy the image.

Windows: Must be the same password defined in the customization specification.

Linux: Must be the same password defined in the template.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpassword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-template_name *templatename*

Specifies the name of the template you want to use for the cloned virtual machine. This parameter replaces -vm_name_cloned_from and you cannot specify both together.

-vm_name_cloned_from *vmname*

Specifies the name of the virtual machine to use as a template for the cloned virtual machine. This parameter replaces -template_name and you cannot specify both together.

-resource_pool_name *resourcepool*

Specifies the name of the resource pool from which you want to select the VM for cloning.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-spec_name *specificationname*

(Optional) Specifies the name of the specification you want to use for the cloned virtual machine.

-compute_resource_name *name*

Specifies the cluster or VMware ESX host where the VM is created.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {*yes|no*}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Clone a VM using Global Credentials

This example creates a new VM, "testvm01" using the template "BaseW2k3" on the data center, DCA/MyCity. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes  
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1  
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator  
-vm_os_password testpassword -vc_server vc_server_1
```

Example: Clone a VM using Authorization File and Component

This example creates a new VM, "testvm01" using the template "BaseW2k3" on the data center, DCA/MyCity. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes  
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1  
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator  
-vm_os_password testpassword -vc_server vc_server_1 -vc_user VCAdmin1  
-auth_file c:\localauth.dat -auth_comp Imaging
```

Example: Clone a VM using Global Credentials and Create Two CPUs

This example creates a new VM, "testvm01" using the template "BaseW2k3" on the data center, DCA/MyCity and creates two CPUs for the VM. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes  
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1  
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator  
-vm_os_password testpassword -set_cpu 2 -vc_server vc_server_1
```

Example: Clone a VM using Default Authorization File and Set Memory to 4096 MB

This example creates a new VM, "testvm01" using the template "BaseW2k3" on the data center, DCA/MyCity and sets the memory to 4096. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -set_memory 4096 -vm_server vm_server_1
-vm_user VCAdmin1
```

Example: Clone a VM using Default Authorization File and Create Two Hard Disks

This example creates a new VM, "testvm01" using the template "BaseW2k3" on the data center, DCA/MyCity and creates 2 hard disks for data store storage1. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000
-create_disk 10000,storage1,2001 -vm_server_1 -vm_user VCAdmin1
```

Example: Clone a VM and Create a Network Connection

This example creates a new VM, "testvm01" and network connection using the template "BaseW2k3" on the data center, DCA/MyCity and creates a network connection and sets the initial values. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -network_connection "nic=1,network_name=VM
Network,ip_address=123.321.5.22,def_gatewat=123.321.1.108,subnet_mask=255.255.255
.0,win_dns_server=123.321.42.1,win_alt_dns=123.321.3.101,wins_primary=123.321.3.3
"
```

Example: Clone a VM using Default Authorization File from a Powered off VM

This example creates a new VM, "testvm01" using VM "testvm02" in place of a template on the data center, DCA/MyCity. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -datacenter_name "DCA/MyCity" -vm_name testvm01 -datastore_name storage1
-esx_host_name ussd-dpmvc.ca.com -resource_pool_name Resources/DPMTTest
-compute_resource_name DPMCluster/ussdCluster -vm_name_cloned_from testvm02
-auto_deploy yes -spec_name w2k3_spec_01 -vm_os_username administrator
-vm_os_password mypassword -vc_server vm_server_1.myco.com -vc_user VCAdmin1
```

Example: Clone a VM using Default Authorization File and Modify an Existing Hard Disk

This example creates a new VM, "testvm01" using the template "BaseW2k3" on the data center, DCA/MyCity and creates 2 hard disks for data store storage1. When the clone operation is complete, CA Server Automation agents are automatically deployed.

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000 -modify_disk
2000,storage2
-vm_server_1 -vm_user VCAdmin1
```

dpmvc createdatastore Command--Create a Datastore

The dpmvc createdatastore command creates a new datastore.

This command has the following format:

```
dpmvc createdatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
-datastore_type {local|nas|vmfs}
[-vmfs_device_path path]
[-vmfs_block_size_mb blocksize]
[-local_path lpath]
[-nas_user_name nasuser]
[-nas_password naspaswd]
[-nas_remote_hostname nasremhost]
[-nas_remote_path nsarempath]
[-nas_access_mode {read-only|read-write}]
[-sc sc_url]
[-locale iso639value]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-datastore_type {*vmfs|nas|local*}

Specifies the type of datastore to be created, VMFS, NAS, or local.

-vmfs_device_path *path*

(Optional) Specifies the device path of the VMFS disk to use when creating a new datastore. You can obtain the path using the "getavailablescsidisks" command.

-vmfs_block_size_mb *blocksize*

(Optional) Specifies the block size to use when creating a new datastore, in MB. Block sizes available are 1, 2, 4 and 8. They enabled VM disk files to reach a maximum of 256GB, 512GB, 1024GB, and 2048GB respectively.

-local_path *lpath*

(Optional) When creating a local datastore, this path specifies the ESX host's local file system path that will be used to create the datastore.

-nas_user_name *nasuser*

(Optional) Specifies the remote host user name for NAS datastore creation.

-nas_password *naspaswd*

(Optional) Specifies the remote host password for NAS datastore creation.

-nas_remote_hostname *nasremhost*

(Optional) Specifies the hostname of the server hosting the network-based storage.

-nas_remote_path *nasrempath*

(Optional) Specifies the file system path on the remote server which is to be used for the NAS datastore

-nas_access_mode *nasaccmode*

(Optional) Specifies the desired kind of access to the network-based storage which will be used for the NAS datastore. Available modes are "read-only" and "read-write". If not specified, the default mode is read-write.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

dpmvc cycle Command--Cycle a VM

The dpmvc cycle command powers on, powers off, resets, or suspends a VM.

This command has the following format:

```
dpmvc cycle
[-sc sc_url]
-powerop {poweron|poweroff|reset|suspend|shutdown guest}
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcserver]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-powerop {poweron|poweroff|reset|suspend|shutdown guest}

Specifies the power operation to perform on the VM. Options include the following:

poweron

Powers on the VM.

poweroff

Powers off the VM.

reset

Resets the VM.

suspend

Temporarily suspends the VM.

shutdown guest

Shuts down the guest OS gracefully.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpassword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {yes|no}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Turn off a VM when there are Multiple vCenter Servers in the Data Center

This example turns off a VM from a specific vCenter Server.

```
dpmvc cycle --powerop poweroff --vc_server my_server1 -datacenter_name my_dc --vm_name my_vm
```

Example: Turn off a VM using Global Credentials in Single vCenter Server Environment

This example turns off the VM, "testvm" that belongs to the data center "DCA/MyCitySan Diego."

```
dpmvc cycle -vm_name testvm -powerop poweroff -datacenter_name DCA/MyCity
```

Example: Turn off a VM using Default Authorization File

This example turns off the VM, "testvm" that belongs to the data center "DCA/MyCitySan Diego."

```
dpmvc cycle -vm_name testvm -powerop poweroff -datacenter_name DCA/MyCity -vc_server vc_server_1
```

Example: Turn on a VM using Specified Authorization File and Component

This example turns on the VM, "testvm" that belongs to the data center "DCA/MyCitySan Diego."

```
dpmvc cycle -vm_name testvm -powerop poweron -datacenter_name DPM/San Diego -vc_server vc_server_1 -vc_user VCAdmin1 -auth_file c:\localauth.dat -auth_component Imaging
```

Example: Turn on a VM using Default Authorization File and Specified Component

This example turns on the VM, "testvm" that belongs to the data center "DCA/MyCitySan Diego."

```
dpmvc cycle -vm_name testvm -powerop poweron -datacenter_name DPM/San Diego  
-vc_server vc_server_1 -vc_user VCAdmin1 -auth_component Imaging
```

dpmvc datastore Command--Get Data Store Properties

The dpmvc datastore command retrieves the free space and capacity settings for a specific VMware vCenter data store.

This command has the following format:

```
dpmvc datastore  
[-sc sc_url]  
-getproperty {all|capacity|freespace}  
-datacenter_name datacentername  
-datastore_name datastorename  
[-vc_server vcserver]  
[-pre]  
[-post]  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]  
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-getproperty {all|capacity|freespace}

(Optional) Specifies which property to retrieve. Options include the following:

all

Retrieves capacity and free space in the data store.

capacity

Retrieves the capacity in the data store.

freespace

Retrieves the free space in the data store.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get all Properties for the Data Store using Global Credentials

This example retrieves both free space and capacity for the data store.

```
dpmvc datastore -getproperty all datacenter_name DCA/MyCity  
-datastore_name storage1 -vc_server vc_server_1
```

Example: Get all Properties for the Data Store

This example retrieves both free space and capacity for the data store.

```
dpmvc datastore -getproperty all datacenter_name DCA/MyCity  
-datastore_name storage1 -vc_server vc_server_1
```

Example: Get all Properties for the Data Store

This example obtains both free space and capacity for the data store.

```
dpmvc datastore -getproperty all datacenter_name DCA/MyCity  
-datastore_name storage1 -vc_server vc_server_1
```

dpmvc delete Command--Destroy a VM

The dpmvc delete command deletes a VM that is in a powered off state. Use this command to clean up and free unused resources.

Important! Verify that you back up any important data *before* you issue this command. This command deletes the VM data store and data files for the VM, including the disk image.

This command has the following format:

```
dpmvc delete  
[-sc sc_url]  
-datacenter_name datacentername  
-vm_name vmname  
[-vc_server vcservername]  
[-vc_user vcuser]  
[-vc_password vcpassword]  
[-auth_file authorizationfilename]  
[-auth_comp componentID] [-wait [timeout]]  
[-pre]  
[-post]  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]  
[-encrypted_password {yes|no}]  
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpasssword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {*yes|no*}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Destroy a VM using Global Credentials in Single vCenter Server Environment

This example destroys the VM "vm11."

```
dpmvc delete -vm_name vm11 -datacenter_name "lab 444"
```

Example: Destroy a VM using Global Credentials in Multiple vCenter Server Environment

This example destroys the VM "vm11."

```
dpmvc delete -vm_name vm11 -datacenter_name "lab 444" -vc_server vc_server_1
```

dpmvc distributedswitch Command--Manage Virtual Distributed Switches

The dpmvc distributedswitch command lets you manage virtual distributed switches.

- Add a new virtual distributed switch to the datacenter
- Update the properties of a virtual distributed switch
- Delete a virtual distributed switch
- Create a new distributed port group to a virtual distributed switch
- Update the port group properties of a virtual distributed switch
- Remove a distributed port group from a virtual distributed switch
- Rename a distributed port group of a virtual distributed switch

The command has the following formats:

```
dpmvc distributedswitch {-vds_add | -vds_update}
-vc_server vcservername
[-datacenter_name datacentername]
-vds_folder vdsfolder
-switch_name switchname
[-hostnics hostname1:nic1,nic2,...nicn [;hostname1:nic2,...nicn ]]
[-uplink_port_names uplink1[,uplink2,...,uplinkn]]
[-maxports maxports]
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc distributedswitch -vds_remove  
-vc_server vcservername  
-switch_name switchname  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```

```
dpmvc distributedswitch -add_portgroup  
-vc_server vcservername  
-switch_name switchname  
-portgroup_name portgroupname  
[-bindtype earlyBinding | ephemeral | lateBinding]  
[-vlan vlanID]  
[-numports numberofports]  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```

```
dpmvc distributedswitch -update_portgroup  
-vc_server vcservername  
-portgroup_name portgroupname  
[-portgroup_newname portgroupnewname]  
[-bindtype earlyBinding | ephemeral | lateBinding]  
[-vlan vlanID]  
[-numports numberofports]  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```

```
dpmvc distributedswitch -remove_portgroup  
-vc_server vcservername  
-portgroup_name portgroupname  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```



```
dpmvc distributedswitch -rename_portgroup  
-vc_server vcservername  
-portgroup_name portgroupname  
-portgroup_newname portgroupnewname  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vds_folder *vdsfolder*

Specifies the folder of the virtual distributed switch in the CA Server Automation Explorer pane.

-switch_name *switchname*

Specifies the switch name to perform the operation on.

-hostnics *hostname1:nic1,nic2,...nicn* [*;hostname1:nic2,...nicn*]

(Optional) Specifies lists of NICs associated with the ESX host members.

-uplink_port_names *uplink1[,uplink2,...,uplinkn]*

(Optional) Specifies a comma-separated list of uplink port names to use.

-maxports *maxports*

(Optional) Specifies the maximum number of ports.

-bindtype *earlyBinding* | *ephemeral* | *lateBinding*

(Optional) Specifies the bind type of the port group. Valid values are:

earlyBinding

Assigns the ports when the VM binds to the portgroup. This type of binding ensures connectivity at all times, but permanently reserves the port. This binding type is the default.

lateBinding

Assigns a port to a VM if the VM is powered on and its NIC is in connected state. This binding type reassigns the port when the VM is powered off or its NIC is disconnected. LateBinding is configurable through vCenter.

ephemeral

Assigns a port to a VM if the VM is powered on and its NIC is in connected state. This binding type reassigns the port when the VM is powered off or its NIC is disconnected. Ephemeral binding is configurable through the ESX Host and vCenter.

-numports *numberofports*

(Optional) Specifies the number of ports of the port group.

-portgroup_name *portgroupname*

Specifies the port group name.

-portgroup_newname *portgroupnewname*

Specifies the new port group name.

-vlan *vlanid*

(Optional) Specifies an Integer value (*vlanid*) to use for the virtual portgroup operations.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a New Virtual Distributed Switch

This example creates a new virtual distributed switch.

```
dpmvc distributedswitch -vds_add -vc_server vc5master -datacenter_name dc3  
-vds_folder vds -switch_name vds1
```

Example: Updates an Existing Virtual Distributed Switch

This example updates a virtual distributed switch. It specifies NICs for MYSERVER1 and removes MYSERVER2 from the virtual distributed switch. When you want to delete a host from a virtual distributed switch, specify the servername without NICs assigned to it (server name followed by colon, for example, MYSERVER2:).

```
dpmvc distributedswitch -vds_update -vc_server VAS-VC5 -datacenter_name VC5  
-switch_name vdistSwitch -hostnics MYSERVER1:nxmg2,tmp2;MYSERVER2:  
-ws_user admin -ws_password ca_admin -post
```

The command updates vdistSwitch:

- Host MYSERVER1 is using NICs nxmg2 and tmp2
- Host MYSERVER2 is removed from the switch

dpmvc entermaintenancemode Command--Set to the Maintenance Mode

The dpmvc entermaintenancemode command sets an ESX host to the maintenance mode

This command has the following format:

```
dpmvc entermaintenancemode  
-esx_host_name esxhostname  
-vc_server vcservername  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
-timeout timeout  
[-evacuate_powered_off_vm {no | yes}]  
[-locale iso639value]
```

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-timeout *timeout*

Specifies the timeout time to enter or exit the maintenance mode in seconds.

-evacuate_powered_off_vm {no | yes}

(Optional) Specifies whether to exit after the VM stops.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Enter Maintenance Mode

This example sets an ESX host into maintenance mode.

```
dpmvc entermaintenancemode -esx_host_name esx1 -vc_server myvcenter -timeout 60
```

dpmvc exitmaintenancemode Command--Exit the Maintenance Mode

The dpmvc exitmaintenancemode command sets the ESX host out of the maintenance mode.

This command has the following format:

```
dpmvc exitmaintenancemode  
-esx_host_name esxhostname  
-vc_server vcservername  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
-timeout timeout  
[-locale iso639value]
```

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-timeout *timeout*

Specifies the timeout time to enter or exit the maintenance mode in seconds.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Exit the Maintenance Mode

This example sets the esx1 ESX host out of maintenance mode.

```
dpmvc exitmaintenancemode -esx_host_name esx1 -vc_server myvcserver  
-timeout 60
```

dpmvc faulttolerance Command--Specify Fault Tolerant Operations

The dpmvc faulttolerance command turns on, turns off, enables, disables, tests fault tolerance, or migrates the secondary VM.

This command has the following format:

```
dpmvc faulttolerance  
{-turn_on|-turn_off|-enable|-disable|-test_ft|-migrate_secondary }  
[-datacenter_name datacentername]  
-vm_name vmname  
[-esx_host_name esxhostname]  
-vc_server vcservername  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
[-locale iso639value]
```

-turn_on

Turns on fault tolerance for the specified VM.

(Optional) Specifies the ESX host name of the secondary VM.

-turn_off

Turns off fault tolerance for the specified VM.

Note: This operation deletes the secondary VM.

-enable

Enables fault tolerance for the specified VM.

-disable

Disables fault tolerance for the specified VM.

-test_ft

Tests fault tolerance on the specified VM.

-migrate_secondary

Migrates the fault tolerance secondary VM to another ESX server. The

-esx_host_name option is required.

-vm_name *vmname*

Specifies the VM.

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-esx_host_name *esxhostname*

(Optional) Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmvc getavailablescsidisks Command--Get Available SCSI Disks

The dpmvc getavailablescsidisks command lists the available SCSI disks.

This command has the following format:

```
dpmvc getavailablescsidisks  
[-datastore_name datastorename]  
-esx_host_name esxhostname  
-vc_server vcservername  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]  
[-sc sc_url]  
[-locale iso639value]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmvc gethosthba Command--Get Host Bus Adapters

The `dpmvc gethosthba` command retrieves the list of host bus adapters configured on the specified ESX host.

This command has the following format:

```
dpmvc gethosthba
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-sc sc_url]
[-locale iso639value]
```

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://*hostname*:*port*/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmvc getresources Command--Get VM Resources

The `dpmvc getresources` command retrieves CPU or memory share limits or reservations.

This command has the following format:

```
dpmvc getresources
[-sc sc_url]
-vm_name vmname
-datacenter_name datacentername
-resource {all|cpulimit|cpureserv|memlimit|memresrv}
[-vc_server vcserver]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-vm_name vmname

Specifies the VM.

-datacenter_name datacentername

Specifies the data center where the VM is located.

-resource {all|vm_cpulimit|vm_cpureserv|vm_memlimit|vm_memresrv}

Specifies whether a specific resource is retrieved or all resources. Options include the following:

all

Retrieves all CPU and memory share limits and reservations for the VM.

vm_cpulimit

Retrieves the limit for the number of shares of CPU for the VM.

vm_cpureserv

Retrieves the number of shares of CPU reserved for the VM.

vm_memlimit

Retrieves the limit for the number of shares of memory for the VM.

vm_memresrv

Retrieves the number of shares of memory reserved for the VM.

-vc_server vcservername

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get all Resources for a VM

This example retrieves all resources for usa-vm2.

```
dpmvc getresources -resource all -datacenter_name MyCity -vm_name usa-vm2
```

Example: Get the CPU Limit for a VM

This example retrieves the individual resource cpulimit for vm uss-vm3.

```
dpmvc getresources -resource cpulimit -datacenter_name MyCity  
-vm_name uss-vm3 -vc_server vc_server_1
```

dpmvc getshares Command--Get VM Shares

The `dpmvc getshares` command lets you view how many shares of CPU or memory are allocated to a VM.

This command has the following format:

```
dpmvc getshares
[-sc sc_url] {-cpu|-memory}
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcserver]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cpu|-memory

Displays how many shares of CPU or memory are allocated to a VM. Shares are used to determine which VMs are given more of the available physical resources proportionally to other running VMs. If you allocate more shares to a VM, more physical resources are given to that VM. If you allocate fewer shares to a VM, less physical resources are given to that VM.

-cpu

Specifies that CPU values are being retrieved. Do not use with `-memory`.

-memory

Specifies that memory values are being retrieved. Do not use with `-cpu`.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get the CPU Values for a VM

This example retrieves CPU values for usa-vm2.

```
dpmvc getshares -resource all -datacenter_name MyCity -vm_name usa-vm2  
-vc_server vc_server_1
```

Example: Get the CPU values for a VM in a Single vCenter Server Environment

This example retrieves CPU values for usa-vm2.

```
dpmvc getshares -resource all -datacenter_name MyCity -vm_name usa-vm2
```

Example: Get the Memory Values for a VM using Global Credentials in a Multiple vCenter Server Environment

This example shows getting the individual resource cpulimit for vm uss-vm3.

```
dpmvc getshares -resource cpulimit -datacenter_name MyCity -vm_name uss-vm3  
-vc_server vc_server_1
```

dpmvc getsnapshots Command--Get VM Snapshots

The dpmvc getsnapshots command lists the snapshots of a VM.

This command has the following format:

```
dpmvc getsnapshots  
[-sc sc_url]  
-datacenter_name datacentername  
-vm_name vmname  
[-vc_server vcserver]  
[-pre]  
[-post]  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]  
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: List Snapshots

This example lists the snapshots of a VM.

```
dpmvc getsnapshots -vc_server my_server1 -datacenter_name my_dc  
-vm_name my_vm
```

dpmvc imgjobcheck Command--Get Clone Job Status

The dpmvc imgjobcheck command obtains the cloning job status for a specific CA Server Automation job ID or a specific VC task ID. The job ID is provided for certain dpmvc commands. Use this command to verify the status on VC.

This command has the following format:

```
dpmvc imgjobcheck  
[-sc sc_url]  
-status {job ID|vc task ID}  
[-vc_server vcserver]  
[-vc_user vcuser]  
[-vc_password vcpassword]  
[-auth_file authorizationfilename]  
[-auth_comp componentID]  
[-pre]  
[-post]  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]  
[-encrypted_password {yes|no}]  
[-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-status jobID

Specifies the job ID used to obtain the job status.

-vc_server vcservername

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user vcuser

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password vcpasssword

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file authorizationfilename

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp componentID

(Optional) Specifies a component ID that you can use to group hosts and users.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {yes|no}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get the Cloning Job Status Using the Job ID in a Single vCenter Server Environment

This example obtains the cloning job status using a CA Server Automation job ID.

```
dpmvc imgjobcheck -status 42
```

Example: Get the Cloning Job Status Using the VC Task ID in a Single vCenter Server Environment

This example obtains the cloning job status using a VC task ID.

```
dpmvc imgjobcheck -status task -21099
```

dpmvc migrate Command--Migrate a VM

The `dpmvc migrate` command migrates a VM from one host server to another host server.

This command has the following format:

```
dpmvc migrate
[-sc sc_url]
-datacenter_name datacentername
-vm_name vm -migrate_to_target_host name
-migrate_to_target_resource_pool_name name
-migrate_to_compute_resource_name name
[-vc_server vcserver]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-migrate_to_target_host *name*

Specifies the name of the host to which you are migrating the VM.

-migrate_to_target_resource_pool_name *name*

Specifies the target resource pool.

-migrate_to_compute_resource_name *name*

Specifies the name of the cluster or VMware ESX host to which you are migrating the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpassword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {yes|no}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Migrate a VM using the Default Authorization File

This example migrates the VM, "MyVM1", from the data center DCA/MyCity to ServerC-dcavc.

```
dpmvc migrate -vm_name MyVM1 -datacenter_name DCA/MyCity
-migrate_to_compute_resource_name DCACluster/ClusterA
-migrate_to_target_resource_pool_name Resources/DCATest
-migrate_to_target_host ServerC-dcavc.MyCompany.com
-vm_server VM_Server1 -vc_user VCAdmin1
```


dpmvc removedatastore Command--Remove a Datastore

The dpmvc removedatastore command deletes a datastore.

This command has the following format:

```
dpmvc removedatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-sc sc_url]
[-locale iso639value]
```

-datastore_name *datastorename*

Specifies the name of the data store where the new VM is located.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmvc removevmvdisk Command--Remove Virtual Disk

The `dpmvc removevmvdisk` command removes a virtual disk from a VM.

This command has the following format:

```
dpmvc removevmvdisk  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
-disk_device_key diskdevicekey  
-delete_data {yes | no}  
[-locale iso639value]
```

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-disk_device_key *diskdevicekey*

Integer that specifies the device key of the disk, for example, 2000.

-delete_data {yes | no}

Indicates whether to delete data while deleting a disk.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Remove a Virtual Disk

This example removes virtual disk with the device key of 2000 from the myvm VM.

```
dpmvc removevmdisk -vm_name myvm -vc_server myvcsrvr -disk_device_key 2000
-delete_data yes
```

dpmvc removevmvnic Command--Remove Virtual NIC

The dpmvc removevmvnic command removes a virtual NIC from a VM.

This command has the following format:

```
dpmvc removevmvnic
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-device_key devicekey
[-locale iso639value]
```

-datacenter_name *datacentername*

(Optional) Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-device_key *devicekey*

Integer that specifies the device key of the network interface, for example, 3000.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Remove a vNIC

This example removes a vNIC from a VM.

```
dpmvc removevmnic -vm_name myvm -vc_server myvcserver -device_key 3000
```

dpmvc setcpu Command-- Set CPUs for a VM

The dpmvc setcpu command sets the number of CPUs for a VM.

This command has the following format:

```
dpmvc setcpu  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-set_cpu number  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
[-locale iso639value]
```

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-set_cpu *number*

(Optional) Define the number of CPUs for the cloned VM. This value overrides the value specified in the template.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set CPUs for a VM

This example sets the number of four CPUs for the myvm VM.

```
dpmvc setcpu -vm_name myvm -vc_server myvcserver -set_cpu 4
```

dpmvc setmemory Command-- Set memory size for a VM

The dpmvc setmemory command sets the memory size for a VM.

This command has the following format:

```
dpmvc setmemory  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-set_memory number  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
[-locale iso639value]
```

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-set_memory *number*

(Optional) Defines the amount of memory in megabytes (MB) for the cloned VM.
This value overrides the value specified in the template.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set memory size for a VM

This example sets the memory size of the myvm VM to 4 GB.

```
dpmvc setmemory -vm_name myvm -vc_server myvcserver -set_memory 4000
```

dpmvc setresources Command--Set VM Resources

The dpmvc setresources command lets you adjust CPU or memory share limits or reservations.

This command has the following format:

```
dpmvc setresources
[-sc sc_url] {-setcpulimit|-setcpureserve|-setmemlimit|-setmemresrv} value
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcservername]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

setcpulimit

Defines the limit for the number of shares of CPU for the VM.

setcpureserv

Defines the number of shares of CPU reserved for the VM.

setmemlimit

Defines the limit for the number of shares of memory for the VM.

setmemresrv

Defines the number of shares of memory reserved for the VM.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpassword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {yes|no}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Resource CPU Limit to 1000 Shares using Global Credentials in a Single vCenter Server Environment

This example adjusts the CPU share limit for the virtual machine "MyVM1" to 1000.

```
dpmvc setresources -setcpulimit 1000 -vm_name MyVM1  
-datacenter_name DCA/MyCity
```

Example: Set Resource CPU Limit to 1000 using a Specific Component ID

This example adjusts the CPU share limit for the virtual machine "MyVM1" to 1000.

```
dpmvc setresources -setcpulimit 1000 -vm_name MyVM1 -datacenter_name DCA/MyCity  
-vc_server vc_server_1 -vc_user VCAAdmin1 -auth_comp Imaging
```

dpmvc setshares Command--Set VM Shares

The dpmvc setshares command lets you set how many shares of CPU or memory are allocated to a VM.

This command has the following format:

```
dpmvc setshares
[-sc sc_url] {-cpu_add_prop_value value|-cpu subtract_prop_value
value|-cpu_overwrite_prop_value value|-mem_add_prop_value
value|-mem_subtract_prop_value value|-mem_overwrite_prop_value value}
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcservername]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-wait [timeout]][-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-cpu_add_prop_value value | -cpu subtract prop value value | -cpu_overwrite_prop_value value | -mem_add_prop_value value | -mem_subtract_prop_value value | -mem_overwrite_prop_value value

Specifies how many shares of CPU or memory to add, subtract, or change. Shares are used to determine which VMs are given more of the available physical resources proportionally to other running VMs. If you allocate more shares to a VM, more physical resources are given to that VM. If you allocate less shares to a VM, less physical resources are given to that VM.

-cpu_add_prop_value value

Specifies the number of shares of CPU to add to the VM.

-cpu_subtract_prop_value value

Specifies the number of shares of CPU to subtract from the VM.

-cpu_overwrite_prop_value value

Specifies the number of shares of CPU to use for the VM.

-mem_add_prop_value value

Specifies the number of shares of memory to add to the VM.

-mem_subtract_prop_value value

Specifies the number of shares of memory to subtract from the VM.

-mem_overwrite_prop_value value

Specifies the number of shares of CPU to use for the VM.

-datacenter_name datacentername

Specifies the data center where the VM is located.

-vm_name vmname

Specifies the VM.

-vc_server vcservername

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user vcuser

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password vcpasssword

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {yes|no}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Increase CPU Shares for a VM using Global Credentials in a Single vCenter Server Environment

This example increases the CPU shares for the virtual machine "MyVM01" by 1000 in the data center "DCA/MyCity."

```
dpmvc setshares -vm_name MyVM01 -datacenter_name DCA/MyCity  
-cpu_add_prop_value 1000
```

Example: Increase CPU Shares for a VM using Global Credentials in a Multiple vCenter Server Environment

This example increases the CPU shares for the virtual machine "MyVM01" by 1000 in the data center "DCA/MyCity."

```
dpmvc setshares -vm_name MyVM01 -datacenter_name DCA/MyCity  
-cpu_add_prop_value 1000 -vc_server vc_server_1
```

Example: Overwrite CPU Shares for a VM using Specified Authorization File and Component ID

This example set the CPU shares for the virtual machine "MyVM01" to 1200 in the data center "DCA/MyCity."

```
dpmvc setshares -vm_name MyVM01 -datacenter_name DCA/MyCity  
-cpu_overwrite_prop_value 1200 -vc_server vc_server_1  
-vc_user VCAdmin1 -auth_file c:\localauth.dat -auth_comp Imaging
```


dpmvc snapshot Command--Manage Snapshots

The dpmvc snapshot command lets you manage snapshots for a VM. You can create snapshots, delete one or all snapshots, or revert to a snapshot using this command.

This command has the following format:

```
dpmvc snapshot
[-sc sc_url]
-operation create
-datacenter_name datacentername -vm_name vmname
-vm_snapshot_name vmsnapshotname
[-vm_snapshot_desc description]
[-withmemory {yes|no}]
[-quiesce {yes|no}][-poweron]
[-vc_server vcservername]
[-vc_user vcusername]
[-vc_password vcpassword]
[-auth_file authorizationfile]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

```
dpmvc snapshot
[-sc sc_url]
-operation remove
-datacenter_name datacentername -vm_name vmname
[-vm_snapshot_name vmsnapshotname]
[-vm_snapshot_id snapid]
[-withchildren {yes|no}]
[-vc_server vcservername]
[-vc_user vcusername]
[-vc_password vcpassword]
[-auth_file authorizationfile]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

```
dpmvc snapshot
[-sc sc_url]
-operation revert
-datacenter_name datacentername -vm_name vmname
-vm_snapshot_name vmsnapshotname
[-vm_snapshot_id snapid]
[-vc_server vcservername]
[-vc_user vcusername]
[-vc_password vcpassword]
[-auth_file authorizationfile]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-operation {create|remove|revert}

Specifies to create, remove, or revert a VM snapshot.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vm_name *vmname*

Specifies the VM.

-vm_snapshot_name *vmsnapshotname*

Defines a name for the VM snapshot.

-vm_snapshot_desc *description*

(Optional) Defines a description for the VM snapshot.

-withmemory {true | false}

(Optional) Specifies that the snapshot includes memory. This parameter is invalid for -remove or -revert.

-quiesce {yes|no}

(Optional) If set to yes and the virtual machine is powered on when the snapshot is taken, VMware Tools quiesce the file system in the virtual machine. This assures that a disk snapshot represents a consistent state of the guest file systems. If the virtual machine is powered off or VMware Tools are not available, the command ignores the quiesce flag.

-vm_snapshot_id *snapid*

(Optional) Defines the snapshot identifier.

-withchildren

(Optional) Specifies that you want to remove all children of the snapshot. This parameter is invalid for -create or -revert.

-poweron

(Optional) Specifies that you want to power on the VM. This parameter is invalid for -remove or -revert.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpassword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {*yes|no*}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a VM Snapshot using Global Credentials in a Single vCenter Server Environment

This example creates a snapshot named "Hello world" with a description of "This is my first snapshot" for the VM system named MyVM01.

```
dpmvc snapshot -create -vm_snapshot_name "Hello world" -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity
```

Example: Revert to a VM Snapshot using Global Credentials in a Single vCenter Server Environment

This example reverts MyVM01 to the snapshot named "Hello world".

```
dpmvc snapshot -revert -vm_snapshot_name "Hello world" -vm_name MyVM01
-datacenter_name VAS/MyCity
```

Example: Remove a VM Snapshot using Global Credentials in a Single vCenter Server Environment

This example deletes the snapshot named "Hello world" from a VM system named MyVM01.

```
dpmvc snapshot -remove -vm_snapshot_name "Hello world" -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity
```

Example: Create a Snapshot for a VM using Global Credentials in a Multiple vCenter Server Environment

This example creates a snapshot named PayrollSnapshot on the VM system named PayrollVM.

```
dpmvc snapshot -create -vm_snapshot_name PayrollSnapshot -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity -vc_server vc_server_1
```

Example: Remove a VM Snapshot using Specified Authorization File and Component ID

This example set the CPU shares for the virtual machine "MyVM01" to 1200 in the data center "DCA/MyCity."

```
dpmvc snapshot -remove -vm_snapshot_name "Snapshot 1" -vm_name MyVM01
-datacenter_name VAS/MyCity -vc_server vc_server_1 -vc_user VCAdmin1
-auth_file c:\localauth.dat -auth_comp Imaging
```

dpmvc templatetovm Command--Convert Template to VM

The dpmvc templatetovm command converts virtual machine templates back to VMs. This capability allows you to apply patches or software updates to the converted VM and then convert it back to a template, for example.

This command has the following format:

```
dpmvc templatetovm
[-sc sc_url]
-template_name templatename
-datacenter_name datacentername
-compute_resource_name hostname
-resource_pool_name resourcepool
-esx_host_name hostname
[-vc_server vmserver]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-template_name *templatename*

Defines the name of the template.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-compute_resource_name *name*

Specifies the cluster or VMware ESX host where the VM is created.

-resource_pool_name *resourcepool*

Specifies the name of the resource pool from which you want to select the VM for cloning.

-esx_host_name *esxhostname*

Specifies the VMware ESX server where the VM resides.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpassword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {yes|no}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Convert a Template into a VM using Global Credentials in a Single vCenter Server Environment

This example converts the template named convertTest into a VM.

```
dpmvc templatetovm -datacenter_name VAS/MyCity -template_name convertTest
-compute_resource_name onDemand/myhost.myco.com
-esx_host_name myhost.myco.com -resource_pool_name Resources/QA
```

Example: Convert a Template into a VM using Global Credentials in a Multiple vCenter Server Environment

This example converts the template named payrollSystem1 into a VM.

```
dpmvc templatetovm -datacenter_name VAS/MyCity -template_name payrollSystem1
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com
-resource_pool_name Resources/QA -vc_server vc_server_1
```

Example: Convert a Template into a VM using Specified Authorization File and Component ID

This example converts the template named Win2K3 into a VM.

```
dpmvc templatetovm -datacenter_name VAS/MyCity -template_name Win2K3
-compute_resource_name onDemand/myhost.myco.com
-esx_host_name myhost.myco.com -resource_pool_name Resources/QA
-vc_server vc_server_1 -vc_user VCAdmin1 -auth_file c:\localauth.dat
-auth_comp Imaging
```


dpmvc vapp Command--Manage vApp

The dpmvc vapp command supports the following operations on vApps:

- Create New vApp
- Clone
- Power On
- Power Off
- Suspend
- Delete vApp from VMware vCenter
- Unregister from VMware vCenter
- Add VMs to vApp
- Add resource pool to vApp
- Add vApps to a vApp
- Update vApp configuration

The command has the following formats:

```
dpmvc vapp -create
-vc_server vcservername
-vapp_path vapppath
[-vapp_name vappname]
[-cpu_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
[-mem_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
[-locale iso639value]
```

```
dpmvc vapp -clone
-vc_server vcservername
-vapp_path vapppath
[-target_vapp_path targetvapppath]
[-vapp_name vappname]
[-target_vapp_host targetvapphost]
```

```
dpmvc vapp [-power_on | -power_off | -suspend | -delete | -unregister ]
-vc_server vcservername
-vapp_path vapppath
```

```
dpmvc vapp -add_vms_to_vapp
-vc_server vcservername
-vapp_path vapppath
[-vapp_name vappname]
[-vms vms]
```

```
dpmvc vapp -add_rps_to_vapp
-vc_server vcservername
-vapp_path vapppath
[-vapp_name vappname]
[-rps resourcepool]
```

```
dpmvc vapp -add_vapps_to_vapp
-vc_server vcservername
-vapp_path vapppath
[-vapp_name vappname]
[-vapps vapps]
[-locale iso639value]
```

```
dpmvc vapp -update_vapp_config
-vc_server vcservername
-vapp_path vapppath
[-vapp_name vappname]
[-config_settings
vappName, objectType, startAction, startDelay, startOrder, stopAction, stopDelay, waitFor
Guest]
[-cpu_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
[-mem_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
[-locale iso639value]
```

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-vapp_path *vapppath*

Specifies the vApp path.

-vapp_name *vappname*

(Optional) Specifies the vApp name.

-cpu_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

(Optional) Specifies the CPU allocation settings when updating a vApp configuration.

isExpandableReservation

Specifies if the allocation is expandable. Possible values: true or false.

limit

Specifies the allocation limit.

reservation

Specifies the reservation allocation.

sharesLevel

Specifies the shares level that is 0 for low, 1 for normal, 2 for high and 3 for custom. e.g. true,-1,1000,1,4000

shares

Specifies the allocation shares.

-mem_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

Specifies the memory allocation settings when updating a vApp configuration.

isExpandableReservation

Specifies if the allocation is expandable. Possible values: true or false.

limit

Specifies the allocation limit.

reservation

Specifies the reservation allocation.

sharesLevel

Specifies the shares level that is 0 for low, 1 for normal, 2 for high and 3 for custom. e.g. true,-1,1000,1,4000

shares

Specifies the allocation shares.

-vms *vms*

(Optional) Specifies the VMs you want to add to vApp.

-rps *resourcepool*

(Optional) Specifies the name of the resource pool which you want to add to a vApp.

-vapps *vapps*

(Optional) Specifies the vApps you want to add to a vApp.

-config_settings

vappName, objectType, startAction, startDelay, startOrder, stopAction, stopDelay, waitForGuest

(Optional) Specifies the configuration settings of the vApp. Proper format is "

startAction none | powerOn

Specifies the start action. The available options are none or powerOn

stopAction none | powerOff | guestShutdown | suspend

Specifies the start action. The available options are none or powerOff or guestShutdown or suspend.

-target_vapp_path *targetvapppath*

Specifies the target vApp path of the vApp you want to create.

-target_vapp_host *targetvapphost*

(Optional) Specifies the host where the new vApp is to reside on.

-target_vapp_datastore *targetvappdatastore*

(Optional) Specifies the name of the data store where the vApp is to reside on.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmvc virtualswitch Command--Manage Virtual Switches

The dpmvc virtualswitch command lets you manage virtual switches.

- Create a new virtual switch
- Update the properties of a virtual switch
- Delete a virtual switch
- Create a new port group for a virtual switch
- Update the port group properties of a virtual switch
- Remove a port group from a virtual switch
- Rename a port group of a virtual switch

Note: This operation does not run asynchronously, and the result gets back immediately. However, the PMM treats the operation as a tasked operation. Therefore the response contains a task ID, but it is always an empty string ("").

The command has the following formats:

```
dpmvc virtualswitch {-vs_add | -vs_update}
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-nic_names nicname1[,nicname2, ...]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc virtualswitch -vs_remove
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc virtualswitch -add_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-portgroup_name portgroupname
[-vlan vlanID]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc virtualswitch -update_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-portgroup_name portgroupname
[-portgroup_newname portgroupnewname]
[-vlan vlanID]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc virtualswitch -remove_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-portgroup_name portgroupname
[-vlan vlanID]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc virtualswitch -rename_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-portgroup_name portgroupname
-portgroup_newname portgroupnewname
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers.
Optional for single vCenter Server environments.

-esx_host_name *esxhostname*

Specifies the name of the ESX server that hosts a VM.

-switch_name *switchname*

Specifies the switch name to perform the operation on.

-nic_names *nicname1 [,nicname2, ...]*

Specifies a list of physical NICs separated by a comma. If you specify multiple NICs, use double quotes to escape the argument, for example: "sc1,nfs1,mnic1,mnic2".

-portgroup_name *portgroupname*

Specifies the port group name.

-portgroup_newname *portgroupnewname*

Specifies the new port group name.

-vlan *vlanID*

(Optional) Specifies an Integer value (vlan ID) used for the virtual portgroup operations.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

This example creates a new virtual switch.

```
dpmvc virtualswitch -vs_add -vc_server vc5master -esx_host_name esx5  
-switch_name switch1 -nic_names "sc1,nfs1,mnic1,mnic2"
```


dpmvc vmtotemplate Command--Convert VM to Template

The dpmvc vmtotemplate command converts virtual machines to templates. This capability helps you to enforce consistency across multiple servers in the data center.

This command has the following format:

```
dpmvc vmtotemplate
[-sc sc_url]
-vm_name vmname
-datacenter_name datacentername
[-vc_server vmserver]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-vm_name *vmname*

Specifies the VM.

-datacenter_name *datacentername*

Specifies the data center where the VM is located.

-vc_server *vcservername*

Specifies the vCenter Server to access when you have multiple vCenter Servers. Optional for single vCenter Server environments.

-vc_user *vcuser*

(Optional) Specifies the vCenter Server user ID. It is optional depending on whether global credentials are enabled.

-vc_password *vcpassword*

(Optional) Specifies the password for the vCenter server user ID. When this option is omitted an encrypted password is retrieved for the user (-vc_user option) and component (-auth_comp option), if specified from either the default authorization file or a specific authorization file (-auth_file).

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *wsuser*

(Optional) Specifies the web service user name to connect to the VC Platform Management Module (PMM).

-ws_password *wspassword*

(Optional) Specifies the password for the web service user to connect to the VC Performance Management Module (PMM).

-prompt {*yes|no*}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-encrypted_password {*yes|no*}

(Optional) Specifies whether you want to encrypt the VC user password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Convert a VM into a Template using Global Credentials in a Single vCenter Server Environment

This example converts the VM named convertTest test into a template.

```
dpmvc vmtotemplate -datacenter_name VAS/MyCity -template_name convertTest  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA
```

Example: Convert a VM into a Template using Global Credentials in a Multiple vCenter Server Environment

This example converts the template named payrollSystem1 into a VM.

```
dpmvc vmtotemplate -datacenter_name VAS/MyCity -template_name payrollSystem1  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA -vc_server vc_server_1
```

Example: Convert VM into a Template using Specified Authorization File and Component ID

This example converts the template named Win2K3 into a VM.

```
dpmvc vmtotemplate -datacenter_name VAS/MyCity -template_name Win2K3  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA -vc_server vc_server_1 -vc_user VCAdmin1  
-auth_file c:\localauth.dat -auth_comp Imaging
```

dpmovf import Command--Import an OVF Package

The `dpmovf import` command imports the OVF package and creates VMs or vApps. You can provide a custom properties file by using the `-properties` attribute. A *custom properties file* allows you to specify custom properties that are defined in the OVF package. The custom properties file contains a list of property keys and the corresponding property values.

Note: If you do not have a custom properties file, the `properties.txt` file is created in your working directory. The default directory is `CA\ProductName\bin`.

This command has the following format:

```
dpmovf import
-host vCenter_server
-user user_name
-password user_password
-name VM_VApp_name
-path OVF_file_path
-datacenter data_center
-datastore data_store
-resourcepool resource_pool
[-locale iso639value]
[-properties properties_file]
```

-host vCenter_server

Specifies the name of the vCenter server host.

-user user_name

Specifies the user name to log in.

-password user_password

Specifies the user password to log in.

-name VM_VApp_name

Specifies the name of the VM or the vApp.

-path OVF_file_path

Specifies the OVF file path.

-datacenter data_center

Specifies the data center name.

-datastore data_store

Specifies the data store.

-resourcepool resource_pool

Specifies the resource pool.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-properties *properties_file*

(Optional) Specifies the custom properties file path.

Example: Import the OVF file for CA Platform using CA Server Automation

This example imports CA Platform OVF package and creates a vApp and VMs. The CA Platform OVF file is *CA Platform_v1_0_0_92c.ovf* and the file location is *D:\OVF\CA_Platform*. The username is *user123*. The following attributes for the vApp are specified: *my_datastore*, *my_datacenter*, and *my_resourcepool*. The custom properties are provided in the *custom_properties.txt* file.

```
dpmovf import -path "D:\OVF\CA_Platform\CA Platform_v1_0_0_92c.ovf" -name
"My_CA_Platform" -host my_host.company.com -user user123 -locale en-US -datastore
"my_datastore" -datacenter "my_datacenter" -resourcepool "my_resourcepool"
-properties "custom_properties.txt"
```

General Shell Commands

You can use the CLI to script and automate miscellaneous commands and run actions based on the command results.

caaipsecurity Command--Specifies Credentials for Web Service Security Check

The caaipsecurity command lets you specify the credentials for your own web service session. When you use caaipsecurity in a web service session, other CLI commands do not prompt you for credentials.

This command has the following format:

```
caaipsecurity setcurrentsession  
[-prompt {yes|no}]  
[-ws_user username -ws_password password -locale iso639value]
```

```
caaipsecurity deletecurrentsession  
[-prompt {yes|no}]  
[-ws_user username -ws_password password -locale iso639value]
```

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmcmd run Command--Run Scripts

dpmcmd run is the command-line interface command for running scripts.

This command has the following format:

```
dpmcmd run -cmdline commandline [-pre] [-post][-locale iso639value]
```

-cmdline *commandline*

Defines the shell command or path where the script file that you want to run is located.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Run a Script and Create Events

This example runs the script named MyScript and creates an event before and after the script is run.

```
dpmcmd run -cmdline C:\scripts\MyScript.bat -pre -post
```

Verification Commands

You can use the dpmccm CLI to script and automate CA Configuration Automation commands (formerly CA Cohesion) and run actions based on the command results.

dpmccm addacmserver Command--Add a Server Instance

The dpmccm addacmserver command adds a specific CA Configuration Automation instance from CA Server Automation.

This command has the following format:

```
dpmccm addacmserver [-sc sc_url] -acm_server ccaservername -port ccaportnumber
-user_name ccausername -pswd ccapassword [-protocol http|https] [-default] [-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-user_name *ccausername*

Defines the login user name for the CA Configuration Automation server. Must be *specialist* or *superuser*.

-pswd *ccapassword*

Defines the password for the CA Configuration Automation Administrator or Architect.

-protocol={*http|https*}

(Optional) Specifies which protocol CA Server Automation uses to communicate with the CA Configuration Automation server over the network. Options include the following:

http

Specifies that CA Server Automation uses a Hypertext Transfer Protocol (HTTP) connection to communicate with the CA Configuration Automation server over the network.

https

Specifies that CA Server Automation uses a Hypertext Transfer Protocol Secure (HTTPS)--HTTP over encrypted Secure Sockets Layer (SSL) connection to communicate with the CA Configuration Automation server over the network.

-default

(Optional) Defines this CCA server as the CA Server Automation default CA Configuration Automation source.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, *fr_FR* for French. To use the locale of the command prompt, specify "native".

Example: Add a Server Instance

This example adds the CCA server instance named CCA1 to CA Server Automation.

```
dpmccm addacmsvr -acm_server CCA1 -port 8080 -user_name acmadmin -pswd ccapassword  
-protocol https
```


dpmccm addcompliancejob Command--Create a Compliance Job

The dpmccm addcompliancejob command creates a compliance job.

This command has the following format:

```
dpmccm addcompliancejob [-sc sc_url] -acm_server ccaserver -port ccaportnumber  
-job_name compliancejobname -rule_groups rulegroups [-job_description <compliance  
job description>] [-remediation_option <rule compliance remediation option>]  
[-compliance_systems <systems for compliance job>] [-compliance_services <services  
for compliance job>] [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm addremediationjob Command--Add a CA Configuration Automation Remediation Job

The dpmccm addremediationjob adds a remediation job.

This command has the following format:

```
dpmccm addremediationjob [-sc sc_url] -acm_server ccaserver -port ccaportnumber  
-job_name remediationjobname -profile_uuid remediationprofileuuid [-job_description  
remediationjobdesc] [-remediation_systems systems_for_job] [-remediation_services  
<services for remediation job>] [-stop_on_fail stopiffailontargetserver] [-ws_user  
username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmccm compare Command--Compare Systems

The dpmccm compare command compares the current snapshot of a target system with the standard snapshot of another system.

This command has the following format:

```
dpmccm compare [-sc sc_url] -host_name source host name -host_name2 hostname2  
-snapshotID snapshotID [-include {0|1}] [-ws_user username -ws_password  
password][-locale iso639value] [-as_job]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *source host name*

Defines the name of the source server in the operation.

-host_name2 *targethostname*

Defines the target server in the operation.

-snapshotID *snapshotID*

Defines the standard snapshot ID for the source server that is used in the comparison.

-include {*Q|1*}

(Optional) Indicates whether to include all differences in the comparison results (1) or only the component differences (0).

Default: 0

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Compare Systems

This example compares the current snapshot of sourceserver, to the standard snapshot on targetserver, and lists all differences found.

```
dpmccm compare -sc https://localhost/dpm/sc -host_name SourceServerA -host_name2 TargetServerB -snapshotID 1005651 -include 0
```

dpmccm createsnapshot Command--Create a Snapshot

The dpmccm createsnapshot command creates a snapshot.

This command has the following format:

```
dpmccm createsnapshot [-sc sc_url] -host_name hostname -snapshot snapshotname -snapshotType 1|2|3|4 [-NoOverride] [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-snapshot *snapshotname*

Defines a name for the snapshot.

-snapshotType=0|1|2|3|4

Defines the snapshot type.

0 = common snapshot without any type

1 = gold standard

2 = baseline

3 = silver

4 = bronze

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Create a Snapshot

This example creates a baseline snapshot.

```
dpmccm createsnapshot -sc https://localhost/dpm/sc -host_name ServerA -snapshot Snapshot999 -snapshotType 2
```

dpmccm deleteacmserver Command--Delete a CCA Server

The `dpmccm deleteacmserver` command deletes a specific CA Configuration Automation server from CA Server Automation.

This command has the following format:

```
dpmccm deleteacmserver [-sc sc_url] -acm_server ccaserver -port ccaportnumber [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a Server from CA Server Automation

This example deletes the server named, EngA from CA Server Automation.

```
dpmccm deleteteacmserver -acm_server EngA -port 8080
```

dpmccm deletecompliancejob Command--Delete a Compliance Job

The `dpmccm deletecompliancejob` command removes a compliance job.

This command has the following format:

```
dpmccm addcompliancejob [-sc sc_url] -acm_server ccaserver -port ccaportnumber  
-job_name compliancejobname [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm deletesnapshot Command--Delete a Snapshot

The `dpmccm deletesnapshot` command deletes a snapshot.

This command has the following format:

```
dpmccm deletesnapshot [-sc sc_url] -host_name hostname -snapshotID snapshotID
[-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-snapshotID *snapshotID*

Defines the server snapshot ID that you want to delete.

Limits: a numeric value

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Delete a Snapshot

This example deletes snapshot, ID 122905.

```
dpmccm deletesnapshot -sc https://localhost/dpm/sc -host_name ServerX -snapshotID 122905
```

dpmccm detect Command--Detect Server Configuration Changes

The dpmccm detect command detects configuration changes of a host by comparing the current host configuration to a baseline snapshot.

This command has the following format:

```
dpmccm detect [-sc sc_url] -host_name hostname [-include {0|1}] [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-include {*0*|*1*}

(Optional) Indicates whether to include all differences in the comparison results (1) or only the component differences (0).

Default: 0

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Detect Server Configuration Changes

This example detects all configuration changes on a server named, ServerA.

```
dpmccm detect -sc https://localhost/dpm/sc -host_name ServerA -include 0
```

dpmccm getaccessprofiles Command--Get Access Profiles

The dpmccm getaccessprofiles command retrieves the CA Configuration Automation agent access profiles defined for a specific source.

This command has the following format:

```
dpmccm getaccessprofiles [-sc sc_url] -acm_server ccaservername [-port ccaportnumber] [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get Profiles From CA Configuration Automation Source

This example retrieves the agent profiles for ServerA.

```
dpmccm getaccessprofiles -acm_server ServerA -port 8080
```

dpmccm getacmagentinfo Command--Get Agent Information

The `dpmccm getacmagentinfo` command retrieves CA Configuration Automation agent information for the system.

This command has the following format:

```
dpmccm getacmagentinfo [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name hostname

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Agent Information

This example gets the agent information for the EngX system.

```
dpmccm getacmagentinfo -host_name EngX
```

dpmccm getallcompliancejobs Command--Get All Compliance Jobs from CCA Server

The dpmccm getallcompliancejobs command obtains a list of compliance jobs from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getallcompliancejobs[-sc sc_url] -acm_server ccaserver -port ccaportnumber  
[-ws_user username -ws_password password][-locale iso639value]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-acm_server ccaservername

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmccm getallcomponents Command--Get Server Components

The `dpmccm getallcomponents` command retrieves all components on a server.

This command has the following format:

```
dpmccm getallcomponents [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Server Components

This example retrieves all components on ServerB.

```
dpmccm getallcomponents -sc https://localhost/dpm/sc -host_name ServerB
```

dpmccm getallremediationjobs Command--Get All CA Configuration Automation Remediation Jobs

The dpmccm getallremediationjobs command obtains the list of remediation jobs from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getallremediationjobs [-sc sc_url] -acm_server ccaserver -port port[-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmccm getallremediationprofiles Command--Get All CA Configuration Automation Remediation Profiles

The `dpmccm getallremediationprofiles` command obtains the list of remediation profiles from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getallremediationprofiles [-sc sc_url] -acm_server ccaserver -port ccaportnumber [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm getallrulegroups Command--Get All Compliance Rule Groups from CCA Server

The `dpmccm getallrulegroups` command obtains a list of compliance rule groups from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getallrulegroups [-sc sc_url] -acm_server ccaserver -port ccaportnumber [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm getallsnapshot Command--Get all Snapshots on a Server

The dpmccm getallsnapshot command retrieves CA Configuration Automation the specified snapshots on a server.

This command has the following format:

```
dpmccm getallsnapshot [-sc sc_url] -host_name hostname -snapshotType {0|1|2|3|4}  
[-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-snapshotType=0|1|2|3|4

Defines the snapshot type.

0 = common snapshot without any type

1 = gold standard

2 = baseline

3 = silver

4 = bronze

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Server Snapshots

This example retrieves all standard snapshots on the server named, MyServer.

```
dpmccm getallsnapshot -host_name MyServer -snapshotType 1
```

dpmccm getallsources Command--Get Servers

The dpmccm getallsources command retrieves a list of all CA Configuration Automation servers defined to CA Server Automation.

This command has the following format:

```
dpmccm getallsources [-sc sc_url] [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get All Servers

This example retrieves a list of all CCA servers defined in CA Server Automation.

```
dpmccm getallsources
```

dpmccm getallsystems Command--Get Server Names from CA Configuration Automation Server

The dpmccm getallsystems command gets a list of server names from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getallsystems [-sc sc_url] -acm_server ccaserver -port ccaportnumber [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Server Names from CA Configuration Automation

This example gets a list of server names from EngA.

```
dpmccm getallsystems -acm_server EngA -port 8080
```

dpmccm getcompliancejobbyname Command--Get a Compliance Job By Job Name

The `dpmccm getcompliancejobbyname` command obtains a compliance job.

This command has the following format:

```
dpmccm getcompliancejobbyname[-sc sc_url] -acm_server ccaserver -port port -job_name compliancejobname [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm getcompliancejobhistory Command--Get Compliance Job History for a Specific Job

The `dpmccm getcompliancejobhistory` command obtains the history of a compliance job by job name.

This command has the following format:

```
dpmccm getcompliancejobhistory [-sc sc_url] -acm_server ccaserver -port port -job_name compliancejobname [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmccm getcurrentactivity Command--Get Server Activity

The `dpmccm getcurrentactivity` command retrieves the current activity for a CA Configuration Automation server.

This command has the following format:

```
dpmccm getcurrentactivity [-sc sc_url] -host_name hostname [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name hostname

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Current Activity

This example gets the current activity from the system, SystemZ.

```
dpmccm getcurrentactivity -host_name SystemZ
```

dpmccm getmanagementprofiles Command--Get Management Profiles

The dpmccm getmanagementprofiles command retrieves all management profiles from a CA Configuration Automation server.

This command has the following format:

```
dpmccm getmanagementprofiles [-sc sc_url] -acm_server ccaservername [-port ccaportnumber] [-ws_user username -ws_password password] [-locale iso639value]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-acm_server ccaservername

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get Management Profiles

This example gets a list of management profiles from the server named, EngZ.

```
dpmccm getmanagementprofiles -acm_server EngZ -port 8080
```

dpmccm getremediationjobhistory Command--Get CA Configuration Automation Remediation Job History

The `dpmccm getremediationjobhistory` command obtains the history of a remediation job from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getremediationjobhistory [-sc sc_url] -acm_server ccaserver -port ccaportnumber -jobname jobname [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm getremediationjobinfo Command--Get CA Configuration Automation Remediation Job Details

The `dpmccm getremediationjobinfo` command obtains the information for a remediation job from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getremediationjobinfo [-sc sc_url] -acm_server ccaserver -port port -jobname jobname [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm getremediationprofile Command--Get CA Configuration Automation Remediation Profile

The dpmccm getremediationprofile command obtains remediation profiles from the CA Configuration Automation server.

This command has the following format:

```
dpmccm getremediationprofile [-sc sc_url] -acm_server ccaserver -port ccaportnumber
-profile_name profilename [-ws_user username -ws_password password] [-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-profile_name *profilename*

Defines the name of the profile for the specified CCA server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmccm getsystemaccessprofile Command--Get Access Profile

The `dpmccm getsystemaccessprofile` command gets the CA Configuration Automation agent access profile for a specified server.

This command has the following format:

```
dpmccm set getsystemaccessprofile [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Access Profile

This example gets the access profile for ServerZ.

```
dpmccm getsystemaccessprofile -host_name ServerZ
```

dpmccm getsystemdefaultsource Command--Get Assigned Source

The dpmccm getsystemdefaultsource command retrieves the assigned default CA Configuration Automation sources for the system.

This command has the following format:

```
dpmccm getsystemdefaultsource [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Assigned Sources

This example retrieves the assigned CA Configuration Automation source for ServerZ.

```
dpmccm getsystemdefaultsource -host_name ServerZ
```

dpmccm getsystemmanagementprofile Command--Get System Management Profile

The `dpmccm getsystemmanagementprofile` command retrieves the assigned management profile for the CA Configuration Automation server.

This command has the following format:

```
dpmccm getsystemmanagementprofile [-sc sc_url] -host_name hostname [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get System Management Profile

This example gets the system management profile for a server named, `EngZ`.

```
dpmccm getsystemmanagementprofile -host_name EngZ
```

dpmccm redoremediationjob Command--Redo a CA Configuration Automation Remediation Job

The dpmccm redoremediationjob redoes a remediation job.

This command has the following format:

```
dpmccm redoremediationjob [-sc sc_url] -acm_server ccaserver -port port -job_name remediationjobname -starttime start_time [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm removesystemdefaultsource Command--Remove Assigned Sources

The dpmccm removesystemdefaultsource command removes the assigned default CA Configuration Automation sources for the system.

This command has the following format:

```
dpmccm removesystemdefaultsource [-sc sc_url] -host_name hostname [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Remove CA Configuration Automation Sources

This example removes the assigned CA Configuration Automation source for ServerZ.

```
dpmccm removesystemdefaultsource -host_name ServerZ
```

dpmccm runacmdiscovery Command--Run Discovery

The `dpmccm runacmdiscovery` command runs a discovery on selected servers.

This command has the following format:

```
dpmccm runacmdiscovery [-sc sc_url] -host_name hostname [-wait] [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-wait

(Optional) Specifies not to return until the CA Configuration Automation discovery operation completes or fails. If you do not specify this option, the CLI returns without waiting for completion.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Run Discovery

This example runs discovery on a server named, `ServerZ`.

```
dpmccm runacmdiscovery -host_name ServerZ -wait
```

dpmccm runcompliancejob Command--Run a Compliance Job

The `dpmccm runcompliancejob` command starts a compliance job.

This command has the following format:

```
dpmccm runcompliancejob [-sc sc_url] -acm_server ccaserver -port port -job_name compliancejobname [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm runremediationjob Command--Run a CA Configuration Automation Remediation Job

The `dpmccm runremediationjob` command runs a remediation job from the CA Configuration Automation server.

This command has the following format:

```
dpmccm runremediationjob [-sc sc_url] -acm_server ccaserver -port ccaportnumber  
-job_name jobname [-ws_user username -ws_password password][-locale iso639value]  
[-as_job]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

dpmccm runsystemmanagementprofile Command--Run a Management Profile

The `dpmccm runsystemmanagementprofile` command runs a management profile on selected servers.

This command has the following format:

```
dpmccm runsystemmanagementprofile [-sc sc_url] -host_name hostname [-wait] [-ws_user username -ws_password password][-locale iso639value]
```


-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-wait

(Optional) Specifies not to return until the CA Configuration Automation discovery operation completes or fails. If you do not specify this option, the CLI returns without waiting for completion.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Run a Management Profile

This example runs a management profile on a server named, `ServerZ`.

```
dpmccm runsystemmanagementprofile -host_name ServerZ
```

dpmccm setacmdefaultmanagementprofile Command--Set the Default Management Profile

The dpmccm setacmdefaultmanagementprofile command sets the default management profile for a CA Configuration Automation server.

This command has the following format:

```
dpmccm setacmdefaultmanagementprofile [-sc sc_url] -profile_name profilename [-port ccaportnumber] -acm_server ccaservername [-port ccaportnumber] -profile_name profilename [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-profile_name *profilename*

Defines the name of the profile for the specified CCA server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Default Management Profile

This example sets the default management profile for the server named, EngZ.

```
dpmccm setacmdefaultmanagementprofile -acm_server EngZ -profile_name Discover OS and Hardware
```

dpmccm setdefaultacmserver Command—Set the Default CA Configuration Automation Server

The `dpmccm setdefaultacmserver` command sets the default CA Configuration Automation server for CA Server Automation.

This command has the following format:

```
dpmccm setdefaultacmserver [-sc sc_url] -acm_server ccaserver [-port ccaportnumber] [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Set the CA Server Automation Default CA Configuration Automation Server

This example sets the CA Server Automation default CA Configuration Automation server to ACMA.

```
dpmccm setdefaulttacmsserver -acm_server ACMA -port 8080
```

dpmccm setsnapshotttype Command--Set Snapshot Type

The dpmccm setsnapshotttype command sets the snapshot type for a snapshot.

This command has the following format:

```
dpmccm setsnapshotttype [-snapshottype_clean] [-sc sc_url] -host_name hostname  
-snapshotid snapshotID -snapshot snapshotname -snapshotType [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-snapshot *snapshotname*

Defines a name for the snapshot.

-snapshotType=0|1|2|3|4****

Defines the snapshot type.

0 = common snapshot without any type

1 = gold standard

2 = baseline

3 = silver

4 = bronze

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmccm setsystemaccessprofile Command--Set Access Profile

The `dpmccm setsystemaccessprofile` command sets the CA Configuration Automation agent access profile for a specified server.

This command has the following format:

```
dpmccm setsystemaccessprofile [-sc sc_url] -host_name hostname -profile_name profilename [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-profile_name *profilename*

Defines the name of the profile for the specified CCA server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Access Profile

This example sets the access profile for ServerZ.

```
dpmccm setsystemaccessprofile -host_name ServerZ -profile_name Agentless
```

dpmccm setsystemdefaultsource Command--Add a Server Source

The dpmccm setsystemdefaultsource command sets a default CA Configuration Automation server for a specific host.

This command has the following format:

```
dpmccm setsystemdefaultsource [-sc sc_url] -host_name hostname -acm_server ccaservername [-port ccaportnumber] [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://*hostname:port*/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Set Default Server

This example sets the CA Configuration Automation server named, `ACM2` as the default server for `ServerZ`.

```
dpmccm setsystemdefaultsource -host_name ServerZ -acm_server ACM2 -port 8080
```

dpmccm setsystemmanagementprofile Command--Set System Management Profile

The `dpmccm setsystemmanagementprofile` command sets the management profile for a CA Configuration Automation server.

This command has the following format:

```
dpmccm setsystemmanagementprofile [-sc sc_url] -host_name hostname -profile_name profilename [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-profile_name *profilename*

Defines the name of the profile for the specified CCA server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set System Management Profile

This example sets the management profile for the server named, EngZ.

```
dpmccm setsystemmanagementprofile -host_name ServerZ -acm_server EngZ
```

dpmccm syncacmserver Command—Synchronize a CA Configuration Automation Server

The dpmccm syncacmserver command synchronizes a CA Configuration Automation server from CA Server Automation.

This command has the following format:

```
dpmccm syncacmserver [-sc sc_url] -acm_server ccaserver [-port ccaportnumber]  
[-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Synchronize CA Configuration Automation Server

This example synchronizes the ACMA CA Configuration Automation server.

```
dpmccm syncacmsserver -acm_server ACMA -port 8080
```

dpmccm testsystemagent Command--Ping Agent

The `dpmccm testsystemagent` command pings the agent on the host for a response.

This command has the following format:

```
dpmccm testsystemagent [-sc sc_url] -host_name hostname [-ws_user username
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the hostname of the host computer. Valid entries: integer, 0-65536

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Ping an Agent for a Response

This example pings the agent on the EngZ host.

```
dpmccm testsystemagent -host_name EngZ
```

dpmccm undoremediationjob Command--Undo a CA Configuration Automation Remediation Job

The dpmccm undoremediationjob undoes a remediation job.

This command has the following format:

```
dpmccm undoremediationjob [-sc sc_url] -acm_server ccaserver -port port -job_name remediationjobname -starttime start_time [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmccm updatecompliancejob Command--Update a Compliance Job

The dpmccm updatecompliancejob command modifies an existing compliance job.

This command has the following format:

```
dpmccm updatecompliancejob [-sc sc_url] -acm_server ccaserver -port port  
-original_job_name compliancejobname [-job_name newcompliancejobname] [-rule_groups  
rulegroups] [-job_description <compliance job description>] [-remediation_option  
<rule compliance remediation option>] [-compliance_systems <systems for compliance  
job>] [-compliance_services <services for compliance job>] [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmccm updatemediationjob Command--Update a CA Configuration Automation Remediation Job

The dpmccm updatemediationjob updates a remediation job.

This command has the following format:

```
dpmccm updatemediationjob [-sc sc_url] -acm_server ccaserver -port ccaportnumber
-original_job_name origremediationjobname [-job_name remediationjobname]
[-profile_uuid remediationprofileuuid] [-job_description remediationjobdesc]
[-remediation_systems systems_for_job] [-remediation_services <services for
remediation job>] [-stop_on_fail stopiffailontargetserver] [-ws_user username
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmccm validateacminfo Command--Validate CCA Server Credentials

The dpmccm validateacminfo command validates credentials for a specific CA Configuration Automation server from CA Server Automation.

This command has the following format:

```
dpmccm validateacminfo [-sc sc_url] -acm_server ccaserver -port ccaportnumber  
user_name username -pswd password -protocol protocol [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-acm_server *ccaservername*

Defines the name of the CCA server.

-port *ccaportnumber*

Defines the communication port for the CCA source server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Validate CA Configuration Automation Server from CA Server Automation

This example validates the EngA server.

```
dpmccm validateacmsserver -acm_server EngA -port 8080 -user_name Admin2 password  
abcde12 -protocol prot_12
```

Discovery Commands

The Discovery commands are used for discovering computers using IP address or subnet IP address.

dpmcda cancelnetworkscan Command--Cancel a Network Discovery

The dpmcda cancelnetworkscan command cancels a network discovery that is in progress.

This command has the following format:

```
dpmcda cancelnetworkscan [-sc sc_url] -network_name networkname [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-network_name *networkname*

Subnet name. Duplicate network names are not allowed; you can only discover one subnet with a specific network name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Cancel a Network Discovery

This command cancels a network discovery that is in progress.

```
dpmcda cancelnetworkscan -network_name Network1
```

dpmcda deletenetwork Command--Delete a Network

The dpmcda deletenetwork command deletes a specific network.

This command has the following format:

```
dpmcda deletenetwork [-sc sc_url] -network_name networkname [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-network_name *networkname*

Subnet name. Duplicate network names are not allowed; you can only discover one subnet with a specific network name.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a Network

This command deletes the network `Network1`.

```
dpmcda deletenetwork -network_name Network1
```

dpmcda discoverhost Command--Discover a Host

The `dpmcda discoverhost` command discovers a host.

This command has the following format:

```
dpmcda discoverhost [sc sc_url] -host_name hostname [-system_user username  
-system_password password][-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostnames*

Defines one or more hosts. Valid entry: Comma-separated host names or IP addresses.

-system_user *username* -system_password *password*

Defines the target system credentials for enhanced discovery (retrieving hardware characteristics).

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Discovery a host

This example discovers a host.

```
dpmcda discoverhost -host_name 172.24.255.255 -ws_user wsuser -ws_password wsuserpassword
```


dpmcda discovernetwork Command--Discover a Network

The dpmcda discovernetwork command discovers a network.

This command has the following format:

```
dpmcda discovernetwork [-sc sc_url] -ipv4_addr ipv4address -network_name networkname
[-dnsserver dnsserver] [-domain domain][-system_user username -system_password
password][-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ipv4_addr *ipv4address*

Defines the subnet IPv4 address for discovery. Valid IPv4 address format for subnet: `xxx.xxx.x.*` or `xxx.xxx.x.{1-10}` or `xxx.xxx.x.0/24`.

-network_name *networkname*

Subnet name. Duplicate network names are not allowed; you can only discover one subnet with a specific network name.

-dns_server *dnsserver*

Defines the IP address of the DNS server.

-domain *domain*

Defines the domain name for DNS discovery.

-system_user *username* -system_password *password*

Defines the target system credentials for enhanced discovery (retrieving hardware characteristics).

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Discover a Network

This example discovers the network named NetworkA with an IPv4 address.

```
dpmcda discovernetwork -ipv4_addr 172.24.255.255 -network_name NetworkA -ws_user  
wsuser -ws_password wsuserpassword
```

Collection Engine Commands

You can use the CLI to script and automate Collection Engine commands and run actions based on the command results. Using these commands requires expert knowledge of SNMP.

Note: You must have expert knowledge of SNMP to use these commands. You must also have expert knowledge to configure column-based metrics for supported MIBs.

dpmce ceconfig -configuremetric Command--Configure Metric

The dpmce ceconfig -configuremetric command lets you add, change, and delete metrics on a host.

This command has the following format:

```
dpmce ceconfig -configuremetric [-sc sc_url] -host_name hostname -operation  
add|delete -sourceType 3 -metricname metricname -metric_type metrictype  
-metric_subtype metricsubtype -metric_instance metricinstance [-metric_method  
exact|complement] [-metric_datatype metricdatatype] [-lower_threshold  
lowerthreshold] [-upper_threshold upperthreshold] [-metric_enabled metricenabled]  
[metric_update yes|no][-ws_user username -ws_password password][-locale  
iso629value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name hostname

Defines the name of the host computer.

-operation add|delete

Adds or deletes an alias (`-setalias`) or metric (`-configuremetric`).

-sourceType

Defines the source for collecting metrics. Valid entries: 1 = System Metrics (Performance Lite metrics), 3 = collection from SNMP interface.

-metricname metricname

Defines a name for the metric.

-metric_type metrictype

The MIB name followed by the symbolic name of the metric.

-metric_subtype MIBOID

Defines the MIB Object Identifier (OID) that uniquely identifies managed objects.

-metric_instance metricinstance

The column number for collecting data.

-metric_method exact|complement|exact_delta|exact_complement

Defines the methods for calculation and collection. Valid entries:

- `exact` = uses the exact metric value. As the value of the metric increases, it contributes to a higher overall usage. Example: Memory.
- `complement` = uses the metric in a complementary way. The higher the value, it contributes to a lower overall usage. Example: CPU Idle Percent.
- `exact_delta` = uses the difference between consecutive values. Example: MIB2.Instance.ifInOctets (number of incoming bytes).

`complement_delta` = uses the difference between consecutive values in a complementary way. Example: CPU Total Idle.

-metric_datatype *metricdatatype*

Defines the data type for the metric. Valid entries: integer, floating-point, or double.

-lower_threshold *lowerthreshold*

Defines the lower overall usage threshold for the Data Center. The lower threshold must always be less than the upper threshold.

Limits: 0-100%

-upper_threshold *upperthreshold*

Defines the upper overall usage threshold for the Data Center. The upper threshold must always be greater than the lower threshold.

Limits: 1-100%

-metric_enabled *yes|no*

Defines if the metric is enabled for collection. Valid entries: yes or no.

-metric_update *yes|no*

Defines whether the metric is updated in the database. Valid entries: yes or no.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Configure Metric for Data Collection

This example adds metric configuration data for the host.

```
dpmce ceconfig -configuremetric -host_name S0L1054 -operation add -sourceType 3
-metricname SysEDGE.loadAverage15Min -metric_type SysEDGE.loadAverage15Min
-metric_subtype 1.3.6.1.4.1.546.1.1.7.8.28.0 -metric_method exact -metric_update
yes -ws_user admin -ws_password admin
```

dpmce ceconfig -configuremetricfilter Command--Configure Metric Filter

The dpmce ceconfig -configuremetricfilter command lets you add, change, and delete the metric filter.

This command has the following format:

```
dpmce ceconfig -configuremetricfilter [-sc sc_url] -host_name hostname -operation
add|delete -sourceType 3 -metric_os_type metricOSType -metric_type metrictype
-metric_subtype metricsubtype -metric_instance metricinstance [-metric_method
exact|complement] [-metric_datatype metricdatatype] [-metric_filter_default 0|1]
[-include_for_overall yes|no] [-lower_threshold lowerthreshold] [-upper_threshold
upperthreshold] [-metric_enabled yes|no] [-ws_user username -ws_password
password] [-locale iso629value]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-host_name hostname

Defines the name of the host computer.

-operation add|delete

Adds or deletes an alias (-setalias) or metric (-configuremetric).

-sourceType

Defines the source for collecting metrics. Valid entries: 1 = System Metrics (Performance Lite metrics), 3 = collection from SNMP interface.

-metric_os_type metricOSType

Defines the operating system type for metric. Valid entries: WINDOWS, AIX, HPUX, LINUX, or SOLARIS.

-metricname metricname

Defines a name for the metric.

-metric_subtype MIBOID

Defines the MIB Object Identifier (OID) that uniquely identifies managed objects.

-metric_instance *metricinstance*

The column number for collecting data.

-metric_method exact|complement|exact_delta|exact_complement

Defines the methods for calculation and collection. Valid entries:

- exact = uses the exact metric value. As the value of the metric increases, it contributes to a higher overall usage. Example: Memory.
- complement = uses the metric in a complementary way. The higher the value, it contributes to a lower overall usage. Example: CPU Idle Percent.
- exact_delta = uses the difference between consecutive values. Example: MIB2.Instance.ifInOctets (number of incoming bytes).
- complement_delta = uses the difference between consecutive values in a complementary way. Example: CPU Total Idle.

-metric_datatype *metricdatatype*

Defines the data type for the metric. Valid entries: integer, floating-point, or double.

-metric_filter_default yes|no

Defines whether the metric is collected by default. Valid entries: yes or no.

-include_for_overall Yes|No

Defines whether the metric is included in the overall calculation.

Yes

Includes the metric in the overall calculation.

No

Excludes the metric from the overall calculation.

-lower_threshold *lowerthreshold*

Defines the lower overall usage threshold for the Data Center. The lower threshold must always be less than the upper threshold.

Limits: 0-100%

-upper_threshold *upperthreshold*

Defines the upper overall usage threshold for the Data Center. The upper threshold must always be greater than the lower threshold.

Limits: 1-100%

-metric_enabled yes|no

Defines if the metric is enabled for collection. Valid entries: yes or no.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Configure Metric Filter

This example adds a metric filter.

```
dpmce ceconfig -configuremetricfilter -operation add -sourceType 3 -metric_os_type
WINDOWS -metricname "Fake.cpuTotalUserPercent" -metric_subtype "xxxxxxxxxxxxxxxx"
-metric_method complement_delta -ws_user dcauser -ws_password dcapassword
```

dpmce ceconfig -disablemetric Command--Disable a Metric

The `dpmce ceconfig -disablemetric` command lets you disable a metric that is being collected for a host.

This command has the following format:

```
dpmce ceconfig -disablemetric [-sc sc_url] -host_name hostname -metricname metricname
[-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the name of the host computer.

-metricname *metricname*

Defines a name for the metric.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Disable a Memory Metric for a Host

This example disables the metric Memory: AvailableMbytes for the host ServerZ.

```
dpmce ceconfig -disablemetric -host_name ServerZ -metricname Memory: AvailableMbytes
```

dpmce ceconfig -disablemetricforoverall Command--Disable and Exclude a Metric From Overall Calculation

The `dpmce ceconfig -disablemetricforoverall` command lets you disable a currently collected host metric and exclude it from the overall calculation.

This command has the following format:

```
dpmce ceconfig -disablemetricforoverall [-sc sc_url] -host_name hostname -metricname metricname [-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the name of the host computer.

-metricname *metricname*

Defines a name for the metric.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Disable a Memory Metric for a Host

This example disables the metric Memory: AvailableMbytes and excludes it from the overall calculation for the host ServerZ.

```
dpmce ceconfig -disablemetricforoverall -host_name ServerZ -metricname Memory:
AvailableMbytes
```

dpmce ceconfig -enablemetric Command--Enable a Metric

The dpmce ceconfig -enablemetric command lets you enable a host metric. The metric must be available for collection.

This command has the following format:

```
dpmce ceconfig -enablemetric [-sc sc_url] -host_name hostname -metricname metricname
[-include_for_overall Yes|No][ -ws_user username -ws_password password][ -locale
iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://*hostname*:*port*/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-host_name *hostname*

Defines the name of the host computer.

-metricname *metricname*

Defines a name for the metric.

-include_for_overall Yes|No

Defines whether the metric is included in the overall calculation.

Yes

Includes the metric in the overall calculation.

No

Excludes the metric from the overall calculation.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Enable a Memory Metric for a Host

This example enables the metric Memory: AvailableMbytes and includes it in the overall calculation for the host ServerZ.

```
dpmce ceconfig -enablemetric -host_name ServerZ -metricname Memory: AvailableMbytes  
-include_for_overall Yes
```

dpmce ceconfig -enablemetricforoverall Command--Enable and Include a Metric for Overall Calculation

The `dpmce ceconfig -enablemetricforoverall` command lets you enable a currently collected host metric and include it in the overall calculation.

This command has the following format:

```
dpmce ceconfig -enablemetricforoverall [-sc sc_url] -host_name hostname -metricname  
metricname [-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the name of the host computer.

-metricname *metricname*

Defines a name for the metric.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Enable a Memory Metric for a Host to be Included in Overall Calculation

This example enables the metric `Memory: AvailableMbytes` and includes it in the overall calculation for the host `ServerZ`.

```
dpmce ceconfig -enablemetricforoverall -host_name ServerZ -metricname Memory:
AvailableMbytes
```

dpmce ceconfig -getglobal Command--Retrieve Data Center Intervals

The `dpmce ceconfig -getglobal` command lets you retrieve data center level collection, recording, and retention intervals.

This command has the following format:

```
dpmce ceconfig -getglobal [-sc sc_url][-ws_user username -ws_password
password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Retrieve Collection, Recording, and Retention Intervals

This example retrieves the data center level collection, recording, and retention intervals.

```
dpmce ceconfig -getglobal
```

dpmce ceconfig -getnode Command--Retrieve Host Collection Intervals

The `dpmce ceconfig -getnode` command lets you retrieve collection, recording, and retention intervals for a host.

This command has the following format:

```
dpmce ceconfig -getnode [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the name of the host computer.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Retrieve Collection, Recording, and Retention Intervals for Host ServerZ

This example retrieves the intervals for the host `ServerZ`.

```
dpmce ceconfig -getnode -host_name ServerZ
```

dpmce ceconfig -getmetriclist Command--Retrieve Metric List

The `dpmce -getmetriclist` command lets you retrieve metrics for a specific host.

This command has the following format:

```
dpmce ceconfig -getmetriclist [-sc sc_url] -host_system hostname [-ws_user username
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the name of the host computer.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Retrieve Metric List

This example retrieves the metric list for the host, `SOL1054`.

```
dpmce ceconfig -getmetriclist -host_name SOL1054
```

dpmce ceconfig -refreshmetriclist Command--Retry Metric Collection

The `dpmce ceconfig -refreshmetriclist` command lets you retry metric collection. For example, if the agent was not installed during discovery, no metrics would be detected for collection. If you install the agent on the system later, you can use this command to retry the metric collection without having to rediscover the system.

This command has the following format:

```
dpmce ceconfig -refreshmetriclist [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the name of the host computer.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Retry Metric Collection

This example retries metric collection on several hosts.

```
dpmce ceconfig -refreshmetriclist -host_name SOL1054:engManager200:QALAB01
```

dpmce ceconfig -setalias Command--Set Alias

The `dpmce ceconfig -setalias` command creates an alias for an ESX host so its metrics can be included in the ESX Server overall calculation.

This command has the following format:

```
ceconfig -setalias [-sc sc_url] -operation add|delete -caller_id PMM -metricname metricname -alias_host_name aliashostname -parent_host_name parenthostname -include_for_overall yes|no [-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-operation *add|delete*

Adds or deletes an alias (`-setalias`) or metric (`-configuremetric`).

-caller_id *PMM*

Defines the interface (Product Management Module) for the alias. Valid entries: PMM.

-metricname *metricname*

Defines a name for the metric. Valid entries: The metric name or ALL (if you want to delete all metrics associated with the alias host name).

-alias_host_name *aliashostname*

Defines the ESX server or VM managed by the parent_host_name (AIM).

-parent_host_name *parenthostname*

Defines the system installed with the AIM.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Set Alias

This example sets an alias for an ESX host.

```
ceconfig -setalias -caller_id PMM -metricname SysEDGE.memInUseCapacity
-alias_host_name hostDA12 -parent_host_name engServer200 -include_for_overall yes
-ws_user admin -ws_password admin
```

dpmce ceconfig -setglobal Command--Set Data Center Intervals

The dpmce ceconfig -setglobal command lets you set the data center level collection, recording, and retention intervals and thresholds.

This command has the following format:

```
dpmce ceconfig -setglobal [-sc sc_url] -recording_interval value
-retention_sec value -retention_day value
-lower_threshold lowerthreshold -upper_threshold upperthreshold
[-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-recording_interval *value*

Defines the recording interval. This value is required if retention_day or retention_sec is not defined.

Unit: seconds

-retention_day *value*

Defines how long to store the average of the daily data in the Performance DB (daily rollup data retention). This value is required if recording_interval or retention_sec are not defined.

Unit: days

-retention_sec *value*

Defines how long to store the polled data in the Performance DB (polled data retention). Consider the number of managed systems, services, and metrics collected when defining this number. The stored polled data objects accumulate over time and can impact performance. If performance issues arise, decrease the number of retention days. This value is required if recording_interval or retention_day are not defined.

Unit: days

-lower_threshold *lowerthreshold*

Defines the lower overall usage threshold for the Data Center. The lower threshold must always be less than the upper threshold.

Limits: 0-100%

-upper_threshold *upperthreshold*

Defines the upper overall usage threshold for the Data Center. The upper threshold must always be greater than the lower threshold.

Limits: 1-100%

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Collection and Recording Intervals for the Data Center

This example sets the collection and recording intervals for the data center.

```
dpmce ceconfig -setglobal -recording_interval 120
```

Example: Set Thresholds for the Data Center

This example sets the lower and upper thresholds for the data center.

```
dpmce ceconfig -setglobal -lower_threshold 10 -upper_threshold 90
```

dpmce ceconfig -setnode Command--Set Intervals for a Host

The dpmce ceconfig -setnode command lets you set collection, recording, and retention intervals for a host.

This command has the following format:

```
dpmce ceconfig -setnode [-sc sc_url] -host_name hostname -recording_interval value
-retention_sec value -retention_day value [-ws_user username -ws_password
password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *hostname*

Defines the name of the host computer.

-recording_interval *value*

Defines the recording interval. This value is required if retention_day or retention_sec is not defined.

Unit: seconds

-retention_sec *value*

Defines how long to store the polled data in the Performance DB (polled data retention). Consider the number of managed systems, services, and metrics collected when defining this number. The stored polled data objects accumulate over time and can impact performance. If performance issues arise, decrease the number of retention days. This value is required if recording_interval or retention_day are not defined.

Unit: days

-retention_day *value*

Defines how long to store the average of the daily data in the Performance DB (daily rollup data retention). This value is required if recording_interval or retention_sec are not defined.

Unit: days

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Retention Interval for Host ServerZ

This example sets the retention interval for the host ServerZ.

```
dpmce ceconfig -setnode -host_name ServerZ -retention_day 300
```

dpmce ceconfig -updatemetric Command--Update a Metric

The dpmce ceconfig -updatemetric command lets you update currently collected host metrics, thresholds, and overall calculation attributes.

This command has the following format:

```
dpmce ceconfig -updatemetric [-sc sc_url] -host_name hostname -metricname metricname [-lower_threshold lowerthreshold][-upper_threshold upperthreshold] include_for_overall Yes|No [-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://*hostname:port*/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-host_name *hostname*

Defines the name of the host computer.

-metricname *metricname*

Defines a name for the metric.

-lower_threshold *lowerthreshold*

Defines the lower overall usage threshold for the Data Center. The lower threshold must always be less than the upper threshold.

Limits: 0-100%

-upper_threshold *upperthreshold*

Defines the upper overall usage threshold for the Data Center. The upper threshold must always be greater than the lower threshold.

Limits: 1-100%

-include_for_overall Yes|No

Defines whether the metric is included in the overall calculation.

Yes

Includes the metric in the overall calculation.

No

Excludes the metric from the overall calculation.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Update a Memory Metric for the Host ServerZ

This example updates the upper and lower thresholds for the metric Memory: AvailableMbytes and includes it in the overall calculation for the host ServerZ.

```
dpmce ceconfig -updatemetric -host_name ServerZ -metricname Memory: AvailableMbytes  
-lower_threshold 10 -upper_threshold 90 -include_for_overall Yes
```

Event Commands

You can use the CLI to script and automate Event commands and run actions based on the command results.

dpmeventmanager create_event Command--Create an Event

The dpmeventmanager create event command creates an event.

This command has the following format:

```
dpmeventmanager create_event [-sc sc_url] [-status eventstatus] [-component eventcomponent] [-message eventmessage] [-source eventsources] [-target eventtarget] [-action eventaction] [-audit {0|1}][--locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-status *eventstatus*

(Optional) Indicates the status of the event.

-component *eventcomponent*

(Optional) Defines the product component to which the event corresponds.

Example: `Policy, Imaging, and so on.`

-message *eventmessagesubstring*

(Optional) Defines the event message substring to match against.

-source *eventsources*

(Optional) Defines the computer that generates the event.

-target *eventtarget*

(Optional) Defines the target computer for which the event occurred.

-action *eventaction*

(Optional) Defines the category of the event.

-audit={0|1}

(Optional) Indicates whether an event is an auditing event. Use 1 (audit event) or 0 (not an audit event).

Default: 0

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create an Event for a Server Discovery

This example creates an event for a server discovery for ServerXYZ.

```
dpmeventmanager create_event -status 1 -component ccm -message "1 New Managed System Added - xyz" -source xyz -action "Inventory Event"
```

dpmeventmanager get_events Command--Retrieve Events

The dpmeventmanager get events command retrieves events.

This command has the following format:

```
dpmeventmanager get_events [-sc sc_url] [-status event_status] [-component eventcomponent] [-message eventmessagesubstring] [-source eventsources] [-target eventtarget] [-action eventaction] [-from_start_date_time] [-to_end_date_time] [-audit {0|1}] [-user_name username][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-status *eventstatus*

(Optional) Indicates the status of the event.

-component *eventcomponent*

(Optional) Defines the product component to which the event corresponds.

Example: Policy, Imaging, and so on.

-message *eventmessagesubstring*

(Optional) Defines the event message substring to match against.

-source *eventsource*

(Optional) Defines the computer that generates the event.

-target *eventtarget*

(Optional) Defines the target computer for which the event occurred.

-action *eventaction*

(Optional) Defines the category of the event.

-from *start_date_time*

(Optional) Specifies the start date and time for events that you want to retrieve using a specified month (*mm*), day (*dd*), or year (*yyyy*). The default is all events regardless of date. This parameter includes the following:

mm

Includes events created starting in this month.

Limits: 1-12

dd

Includes events created starting on this day.

Limits: 1-31

yyyy

Includes events created starting in this year.

Limits: 1970 or later

hh

Includes events created starting at this hour.

Limits: 0-23

mm

Includes events created starting at this minute.

Limits: 0-59

ss

Includes events created starting at this second.

Limits: 0-59

-to end_date_time

(Optional) Specifies the end date and time for events that you want to retrieve using a specified month (*mm*), day (*dd*), and year (*yyyy*). The default is all events regardless of date. This parameter includes the following:

mm

Includes events created ending in this month.

Limits: 1-12

dd

Includes events created ending on this day.

Limits: 1-31

yy

Includes events created ending in this year.

Limits: 01-09

hh

Includes events created ending at this hour.

Limits: 1-24

mm

Includes events created ending at this minute.

Limits: 1-60

ss

Includes events created ending at this second.

Limits: 1-60

-audit={0|1}

(Optional) Indicates whether an event is an auditing event. Use 1 (audit event) or 0 (not an audit event).

Default: 0

-user_name username

Defines the user name of the user who made the policy change that triggered the event. The CLI automatically determines the user name from the operating system.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Retrieve Policy Events for a Specific Time Period

This example retrieves all Policy events from 10:10:10 AM, March 8, 2009 to the present.

```
dpmeventmanager get_events -component policy -from "3/08/2009 10:10:10"
```

Help Desk Commands

You can use the CLI to script and automate help desk commands and run actions based on the command results.

dpmhd addcomment Command--Update a Ticket with a Comment

The dpmhd addcomment command adds a comment to an existing help desk ticket.

This command has the following format:

```
dpmhd addcomment [-sc sc_url] -id ticketid -value comment [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-id *ticketid*

Defines the unique ID assigned to the ticket by the help desk system when the ticket was opened.

Example: `cr:100400`

-value *comment*

Defines the comment string to update the ticket with.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Add Comment to Ticket

This example updates the help desk ticket, cr:400335, indicating that a power off operation started for all virtual machines in SERVICEA.

```
dpmhd addcomment -id cr:400335 -value "Powering off SERVICEA"
```

dpmhd close Command--Close a Help Desk Ticket

The dpmhd close command closes a help desk ticket.

This command has the following format:

```
dpmhd close [-sc sc_url] -id ticketid [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-id *ticketid*

Defines the unique ID assigned to the ticket by the help desk system when the ticket was opened.

Example: `cr:100400`

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Close a Help Desk Ticket

This example closes the help desk ticket identified by ticket ID cr:400335.

```
dpmhd close -id cr:400335
```

dpmhd getattribute Command--Get an Attribute Value of a Ticket

The dpmhd getattribute command gets the value of an existing help desk ticket attribute.

This command has the following format:

```
dpmhd getattribute [-sc sc_url] -id ticketid -name attr_name [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-id *ticketid*

Defines the unique ID assigned to the ticket by the help desk system when the ticket was opened.

Example: `cr:100400`

-name *attr_name*

Defines the name of a ticket attribute. The supported attribute names are as follows:

- `description`
- `status`
- `summary`

Note: These attribute names are case-sensitive.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get the Value of a Ticket Attribute

This example retrieves the value of the description attribute of the help desk ticket identified by the ticket ID `cr:400335`.

```
dpmhd getattribute -id cr:400335 -name description
```

dpmhd getstatus Command--Get the Status of a Ticket

The dpmhd getstatus command gets the status of an existing help desk ticket.

This command has the following format:

```
dpmhd getstatus [-sc sc_url] -id ticketid [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-id *ticketid*

Defines the unique ID assigned to the ticket by the help desk system when the ticket was opened.

Example: `cr:100400`

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Retrieve Ticket Status

This example retrieves the status of the help desk ticket identified by the ticket ID `cr:400335`.

```
dpmhd getstatus -id cr:400335
```

dpmhd open Command--Open a Help Desk Ticket

The dpmhd open command opens a new help desk ticket.

This command has the following format:

```
dpmhd open [-sc sc_url] -summary summarytext [-description descriptiontext]  
[-affected_user username] [-entity_name entityname] [-type tickettype] [-template  
templatename] [-pre] [-post] [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-summary *summarytext*

Defines an abbreviated description for the reason the ticket is being opened.

-description *descriptiontext*

(Optional) Specifies the reason the ticket is being opened.

-entity_name *entityname*

(Optional) Defines the name of the server or service used to match the ticket with a known configuration item in the help desk system. If the configuration item host name is the same as the entity name, the ticket is associated with that configuration item.

-affected_user *username*

(Optional) Defines the end user associated with this ticket.

-type *tickettype*

(Optional) Defines the type of ticket to open. This parameter is required with the `-template` parameter. If you do not specify a type, the default ticket for which the help desk system is configured is used. Ticket types that are supported depend on the configuration of the help desk system. For example, if CA SDM is configured for ITIL support, Change Order, Incident, Problem, and Request ticket types are supported.

-template *templatename*

(Optional) Defines the name of the template to use when opening a ticket. The template contains the default ticket settings. When using the template option, the ticket type is required to locate the template.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmhd setattribute Command--Set an Attribute of a Ticket

The dpmhd setattribute command updates an attribute of an existing help desk ticket.

This command has the following format:

```
dpmhd setattribute [-sc sc_url] -id ticketid -name attr_name -value attr_value
[-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://*hostname:port*/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-id *ticketid*

Defines the unique ID assigned to the ticket by the help desk system when the ticket was opened.

Example: cr:100400

-name *attr_name*

Defines the name of a ticket attribute. The supported attribute names are as follows:

- description
- status
- summary

Note: These attribute names are case-sensitive.

-value *attr_value*

Defines the value to update an attribute with.

Note: If using `setattribute` to update the ticket status, enter the internal code associated with the status.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Update Ticket Attribute

This example sets the value of the description attribute of the help desk ticket identified by ticket ID 400335.

```
dpmhd setattribute -id cr:400335 -name description -value "Poweroff server1"
```

dpmhd setstatus Command--Set the Status of a Ticket

The `dpmhd setstatus` command updates the status of an existing help desk ticket.

This command has the following format:

```
dpmhd setstatus [-sc sc_url] -id ticketid -value statuscode [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-id *ticketid*

Defines the unique ID assigned to the ticket by the help desk system when the ticket was opened.

Example: `cr:100400`

-value *statuscode*

Indicates the status code that you are updating the ticket with. For example, for CA SDM enter the internal code associated with the status. Valid codes used to set the status of CA SDM tickets include:

APP

Sets the status to approved.

CL

Sets the status to closed.

OP

Sets the status to open.

REJ

Sets the status to rejected.

RSCH

Sets the status to researching.

WIP

Sets the status to work in progress.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Update Ticket Status

This example updates the status of the help desk ticket identified by ticket ID cr:400335 to Approved.

```
dpmhd setstatus -id cr:400335 -value APP
```

Solaris Imaging Commands

You can use the dpming CLI to script and automate Solaris Imaging commands and run actions based on the command results.

dpmimg image Command--Deploy a Solaris Image

Use the `dpmimg image` command to deploy a Solaris image from a JumpStart server to a Solaris client, to image a Solaris client using a JumpStart boot server that is different from the JumpStart installation server, or to image Solaris 10 x86 clients.

This command has the following format:

```
dpmimg image [-sc sc_url] -target_host targethostname -target_mac macaddress
[-itcm_server itcm domain manager] -auto_deploy {Yes|No} [-deploy_template
templatename] -target_username username [-target_password password] [-auth_file
authorizationfile] [-auth_comp componentID] -img_host imagehostname -img_name
imagename -img_location location [-boot_img_name bootimagename] [-boot_img_location
bootimagelocation] [-boot_img_host bootimagehost] [-servproc_hostname
serviceprochostname] [-servproc_user serviceprocusername] [-servproc_pw
serviceprocpassword] -dhcp_hostname dhcphostname [-itcm_server SD_adapter_server]
[-scalability_server scalabilityservername] [-wait [timeout]] [-pre] [-post]
[[-ws_user <value> -ws_password <value>]][-prompt <yes|no>]][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-target_host *targethostname*

Defines the name of the target host server to which you are deploying the image.

-target_mac *macaddress*

Defines the hardware address of the computer to which you are deploying the image.

-itcm_server *itcmdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

Note: Valid for CA Server Automation only.

-auto_deploy {yes|no}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-target_username *targetusername*

Defines the user name used for deploying agents to the target host server to which you are deploying the image.

-target_password *targetpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying the image. If you do not specify the password, it is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-img_name *imagename*

Defines the name of the image to deploy.

-img_location *imagelocation*

Defines the location of the image to deploy.

-boot_img_name *bootimagename*

(Optional) Defines the name of the boot image that you want to deploy.

-boot_img_location *bootimagelocation*

(Optional) Defines the location of the image that you want to deploy.

-boot_img_host *bootimagehost*

(Optional) Defines the name of the server where the boot image is located.

-servproc_hostname *serviceprochostname*

(Optional) Defines the host name (Solaris 10 x86 type) of the service processor.

-servproc_user *serviceprocusername*

(Optional) Defines the user name (Solaris 10 x86 type) of the user connecting to the service processor.

-servproc_pw *serviceprocpassword*

(Optional) Defines the password of the user (Solaris 10 x86 type) connecting to the service processor.

-dhcp_hostname *dhcphostname*

Defines the host name (Solaris 10 x86 type) that is used to connect to the DHCP server.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user wsuser

(Optional) Specifies the web service user name to connect to the Imaging Service module.

-ws_password wspassword

(Optional) Specifies the password for the web service user to connect to the Imaging Service module.

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deploy an Image to a Solaris Client Computer

This example deploys a Solaris image from a JumpStart server to a Solaris client computer and does not wait for image deployment to complete.

```
dpmimg image -imghost sunserver001 -img_name jump_5.10 -auto_deploy Yes -target_host sunserver002 -target_username rootuser -target_password rootuserpassword -target_mac 00:00:00:00:00:00 -img_location /export/jump_5.10
```

Example: Deploy an Image to a Solaris 10 x86 Client Computer

This example deploys a Solaris image from a JumpStart server to a Solaris 10 x86 client computer.

```
dpmimg image -target_host sunserver001 -target_mac 00:00:00:00:00:00 -auto_deploy no -target_username root -target_password rootuserpassword -img_host sunimageserver001 -img_name sol_10_508_x86 -img_location /jsimages/SunImageServer/sol_10_508_x86 -servproc_hostname sunserver002 -servproc_user root -servproc_pw rootuserpassword -dhcp_hostname sunimageserver002 -boot_img_host sunimageserver001 -boot_img_name sol_10_508_x86_bt -boot_img_location /jsboot/SunImageServer/sol_10_508_x86_bt -wait 120 -pre -post
```

Example: Use Multiple JumpStart Servers to Deploy a Solaris Image to a Solaris Server

This example uses multiple JumpStart servers to deploy a Solaris image to a Solaris server sunserver002. The install OS image is located on sunserver003 and the boot image is located on the boot server sunserver004. The command waits 2 hours for the image to deploy and then runs.

```
dpmimg image -target_host sunserver002 -target_mac 00:00:00:00:00:00 -auto_deploy
yes -target_username root -target_password rootuserpassword -img_host sunserver003
-img_name sol_10_sparc -img_location /jsimages/sol_10_sparc -boot_img_host
sunserver004 -boot_img_name sol_10_sparc_boot -boot_img_location
/jsboot/sol_10_sparc_boot -wait 120 -pre -post
```

Example: Use Multiple JumpStart Servers to Deploy a Solaris 10 x86 Image to a Solaris 10 x86 Server

This example uses multiple JumpStart servers to deploy a Solaris 10 x86 image to a Solaris 10 x86 server sunserver002. The installation OS image is located on sunserver002 and the boot image is located on the boot server sunserver003. It connects to the service processor server sunserver004 and the DHCP server dhcpserver001.

```
dpmimg image -target_host sunserver002m -target_mac 00:00:00:00:00:00 -auto_deploy
no -target_username root -target_password rootuserpassword -img_host sunserver002
-img_name sol_10_x86 -img_location /jsimages/sol_10_x86 -servproc_hostname
sunserver004 -servproc_user root -servproc_pw rootuserpassword -dhcp_hostname
dhcpserver001 -boot_img_host sunserver003 -boot_img_name sol_10_x86_boot
-boot_img_location /jsboot/sol_10_x86_boot -wait 120 -pre -post
```

dpmimg imgjobcheck Command--Get Solaris Imaging Job Status

The dpmimg imgjobcheck command retrieves the Solaris image job status for a specific CA Server Automation job ID.

This command has the following format:

```
dpmimg imgjobcheck [-sc sc_url] -status jobID [-pre] [-post] [[-ws_user <value>
-ws_password <value>]][-prompt <yes|no>]][-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-status={job ID}

Defines the job ID used to obtain the job status.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user wsuser

(Optional) Specifies the web service user name to connect to the Imaging Service module.

-ws_password wspassword

(Optional) Specifies the password for the web service user to connect to the Imaging Service module.

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get Job Status for Solaris Provisioning Job

This example retrieves the job status for a specific Solaris provisioning job.

```
dpming imgjobcheck -status 45
```

dpmimg refresh_images Command--Refresh Boot and Install Images from Solaris JumpStart Servers

The `dpmimg refresh_images` command retrieves and updates CA Server Automation with the Solaris boot and install images from the specified JumpStart servers. This command has the following format:

```
dpmimg refresh_images [-sc sc_url] --server [all | [server, server]] [-pre] [-post]
[[-ws_user <value> -ws_password <value>][[-prompt <yes|no>]][-locale iso639value]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-server [all | server, server]

Defines the JumpStart server or servers to refresh the boot and install images. This updates CA Server Automation with any changes to boot or install images.

all

Specifies to query all servers for boot and install images.

server

Specifies to query one or more comma-separated server names for boot and install images.

-ws_user wsuser

(Optional) Specifies the web service user name to connect to the Imaging Service module.

-ws_password wspassword

(Optional) Specifies the password for the web service user to connect to the Imaging Service module.

-prompt {yes|no}

(Optional) Specifies whether you are prompted for the web service user ID and password.

Default: Yes

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Refresh CA Server Automation with All Boot and Install Images from All JumpStart Servers

This example retrieves all the boot and install images from every JumpStart Server configured in CA Server Automation and update the images with any changes.

```
dpmimg refresh_images --server all
```

Example: Refresh CA Server Automation with All Boot and Install Images from a Specific JumpStart Server

This example retrieves all the boot and install images from a specific JumpStart Server configured in CA Server Automation and update the image with any changes.

```
dpmimg refresh_images --server sunserver001
```

Network Installation Management Commands

You can use the dpmnim CLI to script and automate Network Installation Management (NIM) commands and run actions based on the command results.

dpmnim image Command—Deploy an IBM AIX Image Using Individual Resources

The dpmnim image command deploys an IBM AIX image from a NIM master server to a target IBM AIX server using individual resources.

This command has the following format:

```
dpmnim image [-sc sc_url] -type individual_res [-install_type {rte|mksysb}] -spot
spotResource [mksysb mksysbResource] [-lpp lppResource] [-boinst_data
bosdataResource] [-resolv_conf resolveconf] [-fb_script fbscript]
[-post_inst_scripts script1, script2, script3] [-image_data imageDataResource]
-machine_res_name machineresourcenname -auto_deploy {Yes|No} -target_username
targetusername [-target_password targetpassword] [-auth_file authorizationfile]
[-auth_comp componentID] -nim_master_host_name nimmasterhostname
[-scalability_server servername] [-deploy_template templatename] [-wait [timeout]]
[-pre] [-post][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-type {*res_group* | *individual_res*}

Specifies to use the imaging operation type resource group or individual resources.

res_group

Specifies to use the resource group for imaging.

individual_res

Specifies to use individual resources for imaging.

-install_type {*rte* | *mksysb*}

(Optional) NIM installation type. Accepted values are *rte* or *mksysb*. *rte* is the default if `-install_type` is not specified.

-spot *spotResource* [*mksysbResource*]

Defines the shared product object tree to use for an imaging request.

-mksysb *mksysbResource*

mksysb resource. Only valid if `-install_type` is *mksysb*.

-lpp *lpp_resource*

Specifies the licensed program product files to use for an imaging request. Optional if `-install_type` is *mksysb*. Required if `-install_type` is *rte*.

-bosinst_data *bos_install_data_resource*

Specifies a file that contains information for the base operating system (BOS) installation program. Optional if `-install_type` is *mksysb*. Required if `-install_type` is *rte*.

-resolve_conf *resolveconf*

(Optional) Defines a file that contains valid `/etc/resolv.conf` entries that define Domain Name Protocol name-server information for local resolver routines.

-fb_script *fbscript*

(Optional) Defines the name of the file to use to configure devices when a NIM client is initially booting after the BOS installation process is complete.

-post_inst_scripts *script1,script2,script3*

(Optional) Specifies a comma-separated list of scripts to run after installation.

-image_data *imageDataResource*

(Optional) Specifies the image data resource file that describes how physical disks, volume groups, logical volumes, file systems, and paging space are configured on the root volume.

-machine_res_name *machineresourcenam*

Defines the name of the machine resource. This name must be predefined at the NIM Master.

-auto_deploy {*yes* | *no*}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-target_username *targetusername*

Defines the user name used for deploying agents to the target host server to which you are deploying the image.

-target_password *targetpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying the image. If you do not specify the password, it is retrieved from the authorization file.

Note: Use the `dpmutil` CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-nim_master_host_name *nimmasterhostname*

Defines the NIM master host name to perform the image deployment.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deploy an IBM AIX image to a server using individual resources

This example deploys an IBM AIX image from a NIM Master server to a server using individual resources.

```
dpmnim image -type individual_res -machine_res_name development_1 -lpp 5301l_res
-spot 530spot_res -bosinst_data 530_bid_ow -resolv_conf master_net_conf -fb_script
fb1 -post_inst_scripts script1,script2 -auto_deploy yes -target_username root
-target_password testpw -nim_master_host_name main_master
```

Example: Deploy an IBM AIX image by specifying individual resources with a MKSYSB image.

This example deploys an IBM AIX image from a NIM Master server to a server using MKSYSB installation.

```
dpmnim image -type individual_res -install_type mksysb -machine_res_name
development_1 -spot 530spot_res -mksysb 5300_mksysb_image -target_username root
-target_password testpw -nim_master_host_name main_master
```

Example: Deploy an IBM AIX image to a server using RTE installation by specifying individual resources.

This example deploys an IBM AIX image from a NIM Master server using RTE installation using individual resource to an IBM AIX server.

```
dpmnim image -type individual_res -install_type rte -machine_res_name development_1
-lpp 5301l_res -spot 530spot_res -bosinst_data 530_bid_ow -resolv_conf
master_net_conf -fb_script fb1 -post_inst_scripts script1,script2 -target_username
root -target_password testpw -nim_master_host_name main_master
```

dpmnim image Command—Deploy an IBM AIX Image Using a Resource Group

The dpmnim image command deploys an IBM AIX image from a NIM master server to a target IBM AIX server using a resource group.

This command has the following format:

```
dpmnim image -type res_group [-sc sc_url] [-install_type
{rte|mksysb}] -res_group_name resource group name -machine_res_name machine resource
name -auto_deploy {Yes|No} [-deploy_template templatename] -target_username
targetusername [-target_password targetpassword] [-auth_file authorizationfile]
[-auth_comp componentID] -nim_master_host_name nim master host name
[-scalability_server servername] [-wait [timeout]] [-pre] [-post][-locale
iso629value]
```

-type {res_group|individual_res}

Specifies to use the imaging operation type resource group or individual resources.

res_group

Specifies to use the resource group for imaging.

individual_res

Specifies to use individual resources for imaging.

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-install_type {rte|mksysb}

(Optional) NIM installation type. Accepted values are rte or mksysb. rte is the default if `-install_type` is not specified.

-res_group_name resource_group_name

Defines the name of the resource group.

-machine_res_name machineresourcename

Defines the name of the machine resource. This name must be predefined at the NIM Master.

-auto_deploy {yes|no}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-target_username *targetusername*

Defines the user name used for deploying agents to the target host server to which you are deploying the image.

-target_password *targetpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying the image. If you do not specify the password, it is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-nim_master_host_name *nimmasterhostname*

Defines the NIM master host name to perform the image deployment.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deploy an IBM AIX Image Using a Resource Group

This example deploys an IBM AIX image from a NIM master server to an IBM AIX server using a resource group.

```
dpmim image -type res_group -res_group_name basic_res_group -machine_res_name
development_1 -auto_deploy yes -target_username root -target_password testpw
-nim_master_host_name main_master
```

Example: Deploy an IBM AIX Image by specifying a Resource Group that contains a MKSYSB resource

This example deploys an IBM AIX image from a NIM Master server using MKSYSB installation using a resource group to an IBM AIX Server.

```
dpmim image -type res_group -install_type mkysyb -res_group_name mkysyb_res_group
-machine_res_name development_1 -auto_deploy yes -target_username root
-target_password testpw -nim_master_host_name main_master
```

Example: Deploy an IBM AIX Image to an Server using RTE installation by specifying a Resource Group

This example deploys an IBM AIX image from a NIM Master server using RTE installation using a resource group to an IBM AIX Server.

```
dpmim image -type res_group -install_type rte -res_group_name basic_res_group
-machine_res_name development_1 -auto_deploy yes -target_username root
-target_password testpw -nim_master_host_name main_master
```


dpmnim image Command—Deploy an IBM AIX Image to an Existing LPAR Using Individual Resources

The `dpmnim image` command uses individual resources to deploy an IBM AIX image from a NIM master server to an existing logical partition (LPAR).

This command has the following format:

```
dpmnim image [-sc sc_url] -type individual_res [-install_type {rte|mksysb}] -spot
spotResource [mksysb mksysbResource] [-lpp lppresource] [-boinst_data
bosDataResource] [-resolv_conf resolveconf] [-fb_script fbscript]
[-post_inst_scripts script1,script2,script3] [-image_data imageDataResource]
-machine_res_name machineresourcenname -nim_master_host_name nimmasterhostname
-auto_deploy {Yes|No} -target_username targetusername [-target_password
targetpassword] [-auth_file authorizationfile] [-auth_comp componentID]
-provision_lpar {Yes|No} -hmc name -managed_system managedsystemname -partition_name
partitionname [-profile_name lparprofile|lparname] [-scalability_server servername]
[-wait [timeout]] [-pre] [-post][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-type {*res_group* | *individual_res*}

Specifies to use the imaging operation type resource group or individual resources.

res_group

Specifies to use the resource group for imaging.

individual_res

Specifies to use individual resources for imaging.

-install_type {*rte* | *mksysb*}

(Optional) NIM installation type. Accepted values are `rte` or `mksysb`. `rte` is the default if `-install_type` is not specified.

-spot *spotResource* [*mksysbResource*]

Defines the shared product object tree to use for an imaging request.

-mksysb *mksysbResource*

mksysb resource. Only valid if `-install_type` is mksysb.

-lpp *lpp_resource*

Specifies the licensed program product files to use for an imaging request. Optional if `-install_type` is mksysb. Required if `-install_type` is rte.

-spot *spotResource [mksysbResource]*

Defines the shared product object tree to use for an imaging request.

-bosinst_data *bos_install_data_resource*

Specifies a file that contains information for the base operating system (BOS) installation program. Optional if `-install_type` is mksysb. Required if `-install_type` is rte.

-resolve_conf *resolveconf*

(Optional) Defines a file that contains valid `/etc/resolv.conf` entries that define Domain Name Protocol name-server information for local resolver routines.

-fb_script *fbscript*

(Optional) Defines the name of the file to use to configure devices when a NIM client is initially booting after the BOS installation process is complete.

-post_inst_scripts *script1,script2,script3*

(Optional) Specifies a comma-separated list of scripts to run after installation.

-image_data *imageDataResource*

(Optional) Specifies the image data resource file that describes how physical disks, volume groups, logical volumes, file systems, and paging space are configured on the root volume.

-machine_res_name *machineresourcename*

Defines the name of the machine resource. This name must be predefined at the NIM Master.

-nim_master_host_name *nimmasterhostname*

Defines the NIM master host name to perform the image deployment.

-auto_deploy {yes|no}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-target_username *targetusername*

Defines the user name used for deploying agents to the target host server to which you are deploying the image.

-target_password *targetpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying the image. If you do not specify the password, it is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-provision_lpar {Yes|No}

Indicates whether the logical partition is imaged. Options are:

Yes

Images the logical partition.

No

Treats partition as machine to provision to, if basic NIM requirements are met.

Default: No

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-profile_name *lparprofile* (HMC only)

[-profile_name *lparname*] (IVM only)

Specifies the partition profile to use when you activate the logical partition. Required for HMC. Optional for IVM, and if specified the name must match the partition name.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deploy an IBM AIX image to an LPAR using individual resources

This example deploys an IBM AIX image from a NIM Master server to an IBM AIX LPAR using individual resources.

```
dpmnim image -type individual_res -lpp lpp_res -spot spot -bosinst_data bid_ow  
-resolv_conf master_net_conf -post_inst_scripts postins_scripts -machine_res_name  
ResPartition -nim_master_host_name machine.domain.com -auto_deploy no  
-target_username root -target_password testpw -provision_lpar yes -hmc hmc01  
-managed_system testMS -partition_name lpartest01 -profile_name Default
```

Example: Deploy an Image to an IBM AIX LPAR by specifying individual resources with a mksysb image.

This example deploys an IBM AIX image from a NIM Master server using MKSYSB installation to an IBM AIX LPAR.

```
dpmnim image -type individual_res -install_type mksysb -machine_res_name  
development_1 -spot 530spot_res -mksysb 5300_mksysb_image -target_username root  
-target_password testpw -nim_master_host_name main_master -provision_lpar yes -hmc  
hmc01 -managed_system testMS -partition_name lpartest01 -profile_name Default
```

Example: Deploy an IBM AIX image to an LPAR using RTE installation by specifying individual resources.

This example deploys an IBM AIX image from a NIM Master server using RTE installation using individual resource to an IBM AIX LPAR.

```
dpmnim image -type individual_res -install_type rte -machine_res_name development_1  
-lpp 530l1_res -spot 530spot_res -bosinst_data 530_bid_ow -resolv_conf  
master_net_conf -fb_script fb1 -post_inst_scripts script1,script2 -target_username  
root -target_password testpw -nim_master_host_name main_master -provision_lpar yes  
-hmc hmc01 -managed_system testMS -partition_name lpartest01 -profile_name Default
```

dpmnim image Command—Deploy an IBM AIX Image to an Existing LPAR Using a Resource Group

The `dpmnim image` command uses a resource group to deploy an IBM AIX image from a NIM master server to an existing IBM AIX logical partition (LPAR).

This command has the following format:

```
dpmnim image -type res_group [-sc sc_url] [-install_type {rte|mksysb}]
-res_group_name resourcegroupname -machine_res_name machineresourcenam
-provision_lpar {Yes|No} -hmc name -managed_system managedsystemname -partition_name
partitionname [-profile_name lparprofile|lparname] -auto_deploy {Yes|No}
-target_username targetusername [-target_password targetpassword] [-auth_file
authorizationfile] [-auth_comp componentID] -nim_master_host_name nimmasterhostname
[-scalability_server servername] [-wait [timeout]] [-pre] [-post][--locale
iso629value]
```

-type {res_group|individual_res}

Specifies to use the imaging operation type resource group or individual resources.

res_group

Specifies to use the resource group for imaging.

individual_res

Specifies to use individual resources for imaging.

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-install_type {rte|mksysb}

(Optional) NIM installation type. Accepted values are `rte` or `mksysb`. `rte` is the default if `-install_type` is not specified.

-machine_res_name *machineresourcenam*

Defines the name of the machine resource. This name must be predefined at the NIM Master.

-res_group_name *resourcegroupname*

Specifies name of the resource group.

-provision_lpar {*Yes|No*}

Indicates whether the logical partition is imaged. Options are:

Yes

Images the logical partition.

No

Treats partition as machine to provision to, if basic NIM requirements are met.

Default: No

-hmc *name*

Specifies the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) that controls the managed systems where the logical partition resides.

-managed_system *managedsystemname*

Specifies the managed system on which the logical partition resides. The managed system must exist on the HMC/IVM.

-partition_name *partitionname*

Specifies the logical partition to which the resources are going to be applied. The partition must exist on the managed system.

-profile_name *lparprofile* (HMC only)**[-profile_name *lparname*] (IVM only)**

Specifies the partition profile to use when you activate the logical partition. Required for HMC. Optional for IVM, and if specified the name must match the partition name.

-auto_deploy {*yes|no*}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-target_username *targetusername*

Defines the user name used for deploying agents to the target host server to which you are deploying the image.

-target_password *targetpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying the image. If you do not specify the password, it is retrieved from the authorization file.

Note: Use the `dpmutil` CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-nim_master_host_name *nimmasterhostname*

Defines the NIM master host name to perform the image deployment.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deploy an IBM AIX Image Using MKSYB Installation

This example deploys an IBM AIX image by using a resource group that contains a mksysb resource.

```
dpmnim image -type res_group -install_type mksysb -res_group_name mksysb_res_group
-machine_res_name development_1 -auto_deploy yes -target_username root
-target_password testpw -nim_master_host_name main_master -provision_lpar yes -hmc
hmc01 -managed_system testMS -partition_name lpartest01 -profile_name Default
```

Example: Deploy an IBM AIX Image Using RTE Installation

This example deploys an IBM AIX image by RTE installation using a resource group.

```
dpmnim image -type res_group -install_type rte -res_group_name basic_res_group
-machine_res_name development_1 -auto_deploy yes -target_username root
-target_password testpw -nim_master_host_name main_master -provision_lpar yes -hmc
hmc01 -managed_system testMS -partition_name lpartest01 -profile_name Default
```

Example: Deploy an Image to an IBM AIX LPAR Using a Resource Group

This example deploys an IBM AIX image from a NIM Master server to an IBM AIX LPAR using a resource group.

```
dpmnim image -type res_group -res_group_name basic_res_group -machine_res_name
ResPartition -provision_lpar yes -hmc hmc01 -managed_system testMS -partition_name
lpartest01 -profile_name Default -auto_deploy no -target_username root
-target_password testpw -nim_master_host_name machine.domain.com
```

dpmnim imgjobcheck Command—Retrieve NIM IBM AIX Imaging Job Status

The dpmnim imgjobcheck command retrieves the IBM AIX image job status for a specific CA Server Automation job ID.

This command has the following format:

```
dpmnim imgjobcheck [-sc sc_url] -status jobID [[-ws_user username -ws_password
password]][-prompt yes|no] [-pre] [-post][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-status *jobID*

Specifies the CA Server Automation job ID used to obtain the job status.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Retrieve the IBM AIX Job Status Using the Job ID

This example obtains the job status of the IBM AIX image using the CA Server Automation job ID 42.

```
dpmnim imgjobcheck -status 42
```

Object Model Utility Commands

Caaipaomwsclient.exe is a general utility that you can use to diagnose problems in the product infrastructure. Some of the tasks it can do include:

- Enumerating, updating, or deleting objects and associations
- Showing statistics on used classes, associations and indications
- Displaying the chain of propagated state for an object
- Displaying detailed status of a web service

Be careful when performing any operation directly on the database. Generally, this utility is used only when working with CA Technologies Technical Support.

caaipaomwsclient Command--Enumerating Objects

This command runs a filtered query against the database.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/enumerate=class_name [/whereFilter=where_filter]  
[/output=output_type] /locale=iso639value
```

user

Defines the CA EEM user name.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class being queried. Output also includes instances of any subclasses. Use '*' to dump the entire database.

whereFilter

Optional section of the query. For example,

```
/whereFilter="where Type='A0M174' ".
```

output type

(Optional) Valid types are 'console', 'XML' and 'null'. Null is useful if only a count of the objects is needed.

Default: console

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

caaipaomwsclient Command--Updating Objects

This command updates one or more properties of an object. Supply the keys and the properties to be updated as key=value pairs on the command line.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/setProperty=class_name key1=value key2=value property1=value  
property2=value /locale=iso639value
```

user

Defines the CA EEM user name.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class of the object being queried.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

Example

In this example, Name and CreationClassName are the required keys of the object, and HealthState is the property being updated.

```
/setProperty=ComputerSystem Name=0019CC  
CreationClassName=ComputerSystem HealthState=15
```

caaipaomwsclient Command--Deleting Objects

This command deletes an object. Enter key=value pairs on the command line. Because objects in the topology usually have several associations and dependencies, use the dedicated command-line utilities whenever possible for deletions.

This command has the following format:

```
caaipaomwsclient /user=user /password=password /delete=class_name  
key1=value key2=value /locale=iso639value
```

user

Defines the CA EEM user name.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class of the object being queried.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

Example

```
/delete=Action Name=0019CC
```

caaipaomwsclient Command--List Associated Objects

This command lists the associations or the associated objects of an object. Enter the object as key=value pairs on the command line.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/dumpAssociations=class_name key1=value key2=value [/verbose]  
/locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class of the object being queried.

verbose

(Optional) Lists the associated objects rather than the associations.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

Examples

- In this example, ContainerId is the required key of the object. All associations that reference this object are listed on the console:

```
/dumpAssociations=CA_Container ContainerId=3
```
- In this example, all objects associated with this container are listed on the console:

```
/dumpAssociations=CA_Container /verbose ContainerId=3
```

caaipaomwsclient Command--Summarize Class Use

This command displays a count of the classes currently in the database, grouped by name.

This command has the following format:

```
caaipaomwsclient /user=user /password=password /summarizeClasses  
/locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

caaipaomwsclient Command--Summarize Object Associations

This command lists all the associations currently applied to a class.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/summarizeAssociations=class_name /locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class of the object being queried.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

caaipaomwsclient Command--List Available Associations

This command lists all the association classes that could be applied to the specified class.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/getAvailableAssociations=class_name /locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class of the object being queried.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

caaipaomwsclient Command--Summarize Indications

This command lists the indications stored in the database, grouped by class name.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/summarizeIndications /locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

caaipaomwsclient Command--Display Class Information

This command lists the properties, property types, keys, and superclasses of a class.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/dumpClass=class_name /locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class of the object being queried.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

caaipaomwsclient Command--Display Propagated State

This command follows the chain of objects that have propagated their state up to the specified container. The keys of the container are supplied as key=value pairs on the command line.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/dumpPropagatedState=class_name key1=value key2=value  
/locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

class name

Specifies the class of the object being queried.

verbose

(Optional) Lists the associated objects rather than the associations.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

Example

In this example, ContainerId is the required key of the object. Output to the console shows each object in the topology that has propagated the state up to this container.

```
/dumpPropagatedState=CA_Container ContainerId=3
```

caaipaomwsclient Command--Display Detailed Status of a Web Service

This command displays all available information related to the status of the specified web services. The status includes the Service Controller state, the service entry in the database, and the results of actually calling the service.

This command has the following format:

```
caaipaomwsclient /user=user /password=password  
/getServiceStatus=service_names /locale=iso639value
```

user

Defines the CA EEM username.

password

Defines the password for the specified CA EEM user.

service_names

Specifies the name of the service being queried. To query multiple services, enclose them in quotation marks and separate them with commas.

locale

(Optional) Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

Example

In this example, we request the status of the Performance Monitor and Data Service web services.

```
/getServiceStatus='ce aom'
```

Policy Commands

You can use the `dppolicy` CLI to script and automate Policy commands and run actions based on the command results.

`dppolicy createdefaultactions` Command--Create Default Actions

The `dppolicy createdefaultactions` command creates the four default Set Health State actions. These actions are created during installation. Use this command if you want to create these actions again, for example, after having removed them accidentally.

This command has the following format:

```
dppolicy createdefaultactions [-sc sc_url] [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmpolicy exportaction Command--Export Actions to a File

The dpmpolicy exportaction command exports actions to a file.

This command has the following format:

```
dpmpolicy exportaction [-sc sc_url] [-action_name actionname] -file filename
[-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-action_name *actionname*

Defines the name of the action. If you do not specify this parameter, all actions are exported.

-file *filename*

Defines the file name.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Export Actions to a File

This example exports a single action to a file.

```
dpmpolicy exportaction -action_name ACTION_A -file C:\ACTION_A.txt
```

dpmpolicy exportrule Command--Export Rules to a File

The dpmpolicy exportrule command exports rules to a file.

This command has the following format:

```
dpmpolicy exportrule [-sc sc_url] [-rule_name rulename] -file filename [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-rule_name *rulename*

Defines the name that you assigned to the rule that you created.

-file *filename*

Defines the file name.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Export a Rule to a File

This example exports a single rule to a file.

```
dpmpolicy exportrule -rule_name SERVER_RULE_A -file C:\SERVER_RULE_A.txt
```

Example: Export All Rules to a File

This example exports all rules to a file.

```
dpmpolicy exportrule -file C:\ALL_RULES.txt
```

dpmpolicy findbestmachine Command--Find the Best Server

The dpmpolicy findbestmachine command finds the server with the lowest overall usage from a specified list of servers.

This command has the following format:

```
dpmpolicy findbestmachine [-sc sc_url] -machine_list machinelist [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Find the Best Server

This example finds the server with the lowest overall usage from the specified server list.

```
dpmpolicy findbestmachine -machine_list "machine1.ca.com, machine2.ca.com,  
172.24.36.107"
```

dpmpolicy findcandidatemachine Command--Find the Candidate Servers

The dpmpolicy findcandidatemachine command finds the servers that match user-defined specifications.

This command has the following format:

```
dpmpolicy findcandidatemachine [-sc sc_url] [-machine_list machinelist] -constraint  
Source:OP:Value [-constraint Source:OP:Value] [-ws_user username -ws_password  
password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-constraint={Source:OP:Value},...

Defines the specifications that the candidate servers must meet. Options include the following:

Source={MEMORY|CPUSPEED|STORAGE|ARCH}

Specifies the type of constraint and an associated value.

MEMORY

Specifies a memory constraint.

Limits: MB

CPUSPEED

Specifies a CPU processing speed constraint.

Limits: MHz

STORAGE

Specifies a storage constraint.

Limits: MB

ARCH

Specifies a hardware constraint.

Limits: X86, SPARC, PA-RISC, POWERPC or X64.

OP={EQ|NEQ|GT|GTE|LT|LTE}

Specifies an operand. Operands include the following:

EQ=equal to

NEQ=not equal to

GT=greater than

GTE=greater than or equal to

LT=less than

LTE=less than or equal to

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Find Servers

This example finds servers with more than 2000 MB of system memory.

```
dpmpolicy findcandidatemachine -constraint MEMORY:GT:2000
```

Example: Find Servers in the List

This example finds servers with more than 2000 MB of system memory and 2400 MHz of CPU or more from the provided list of servers.

```
dpmpolicy findcandidatemachine -constraint MEMORY:GT:2000 -constraint  
CPUSPEED:GTE:2400 -machine_list alert.ca.com, 141.202.224.59
```

dpmpolicy getruleexeccount Command--Retrieve rule execution count

The `dpmpolicy getruleexeccount` command returns the number of times the rule has attempted to run.

This command has the following format:

```
dpmpolicy getruleexeccount [-sc sc_url] -rule_name rulename [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-rule_name *rulename*

Defines the name that you assigned to the rule that you created.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get Rule Execution Count

This example retrieves the number of times a rule has been executed.

```
dpmpolicy getruleexccount -rule_name GENERIC_RULE
```

dpmpolicy importaction Command--Import Actions from a File

The dpmpolicy importaction command imports actions from a file.

This command has the following format:

```
dpmpolicy importaction [-sc sc_url] -file filename [-action_name actionname]  
[-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-file *filename*

Defines the file name.

-action_name *actionname*

Defines the name of the action. When the file contains only one action, this option can be used to rename the action upon import. If the file contains more than one action, the value is ignored.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Import Actions From a File

This example imports an action and renames it.

```
dppolicy importaction -file C:\ACTION_A.txt -action_name ACTION_B
```

dppolicy importrule Command--Import Rules from a File

The dppolicy importrule command imports rules from a file.

This command has the following format:

```
dppolicy importrule [-sc sc_url] [-entity_type entitytype -entity_name entityname]
-file filename [-rule_name rulename] [-ws_user username -ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-entity_type={*server*|*service*}

Defines whether this rule is associated with a server or service.

-entity_name *entityname*

Defines the name of the server or service.

-file *filename*

Defines the file name.

-rule_name *rulename*

Defines the name that you assigned to the rule that you created. When the file contains only one rule, this option can be used to rename the rule upon import. If the file contains more than one rule, the value is ignored.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Import a Rule From a File

This example imports a rule and renames it.

```
dpmpolicy importrule -file C:\RULE_A.txt -rule_name RULE_B
```

dpmpolicy resetruleexccount Command--Reset Retries to Run a Rule

The dpmpolicy resetruleexccount command resets the number of times the rule has attempted to run back to zero.

This command has the following format:

```
dpmpolicy resetruleexccount [-sc sc_url] -entity_type entitytype -entity_name entityname -rule_name rulename [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-entity_type={*server*|*service*}

Defines whether this rule is associated with a server or service.

-entity_name *entityname*

Defines the name of the server or service.

-rule_name *rulename*

Defines the name that you assigned to the rule that you created.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Reset Server Threshold

This example resets the threshold for a rule that runs on a server.

```
dpmpolicy resetruleexccount -entity_type 0 -entity_name machine1.ca.com -rule_name  
SERVER1_RULE
```

Example: Reset Retries for a Service Rule

This example resets the number of retries to zero for a rule that runs on a service.

```
dpmpolicy resetruleexccount -entity_type 1 -entity_name SERVICE1 -rule_name  
SERVICE1_RULE
```

dpmpolicy runaction Command--Run an Action

The dpmpolicy runaction command runs actions and action sequences.

This command has the following format:

```
dpmpolicy runaction [-sc sc_url] -action_name action name [-event_source event source] [-event_message event_message] [-rule_name rule name] [-server_name server name] [-service_name service name] [-propagate] [-vm_name vm name] [-datacenter datacenter name] [-host_system host system] [-wait_rc] [-pre] [-post] [-ws_user username -ws_password password] [-locale iso639value] [-D-overwrite Parameter=value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-action_name *actionname*

Defines the name of the action.

-event_source *eventsource*

(Optional) Defines the value with which to populate the %EVENTSOURCE% substitution variable.

-event_message *eventmessage*

(Optional) Defines the value with which to populate the %EVENTMESSAGE% substitution variable.

-rule_name *rulename*

(Optional) Defines the value with which to populate the %RULENAME% substitution variable.

-server_name *servername*

(Optional) Defines the value with which to populate the %SERVER% substitution variable.

-service_name *servicename*

(Optional) Defines the value with which to populate the %SERVICE% substitution variable.

-propagate

(Optional) Specifies that you want to run an action against all servers in the service that you specified in the `-service_name` option.

-vm_name *vmname*

(Optional) Defines the value with which to populate the `%VMNAME%` substitution variable.

-datacenter_name *datacentername*

(Optional) Defines the value with which to populate the `%DATACENTER%` substitution variable.

-host_system *hostsystem*

(Optional) Defines the value (the name of the VMware ESX host computer) with which to populate the `%HOSTSYSTEM%` substitution variable.

-wait_rc

(Optional) Specifies whether to wait for a Run Command Script action to finish.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

-D-*overwrite Parameter=value*

(Optional) Defines a new value for the specified parameter.

Example: Run Action to Bring a Server Online

This example runs an action that brings a server online.

```
dppolicy runaction -action_name BringServerOnline -server_name Server001  
-service_name Production -pre -post
```

Example: Run Action

This example runs an action.

```
dpmpolicy runaction -action_name Action1
```

Example: Run an Action Against all Servers in a Service

This example runs an action against all servers in the SERVICE1 service.

```
dpmpolicy runaction -action_name ACTION1 -service_name SERVICE1 -propagate
```

Example: Run an Action Requiring a VM

This example runs an action using a VM.

```
dpmpolicy runaction -action_name vc_action -vm_name Finance\Group\FinVMServer1  
-datacenter FINANCE\NYC
```

Example: Run an Action with a Different Parameter Value

This example runs an action with a vm_name parameter value that is different to the value set in the CA Server Automation user interface.

```
dpmpolicy runaction -action_name Create_snapshot -D-vm_name=VM02
```

dpmpolicy setruleexeclimit Command--Set Limits for a Rule

The dpmpolicy setruleexeclimit command sets a limit that determines how many times your rule retries.

This command has the following format:

```
dpmpolicy setruleexeclimit [-sc sc_url] -entity_type entitytype -entity_name  
entityname -rule_name rulename -limit_option option [-limit_value value] [-ws_user  
username -ws_password password] [-locale iso639value]
```


-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-entity_type={server|service}

Defines whether this rule is associated with a server or service.

-entity_name entityname

Defines the name of the server or service.

-rule_name rulename

Defines the name that you assigned to the rule that you created.

-limit_option=[limited|unlimited|disabled]

Specifies the limit option for the number of times the rule runs. Options are as follows:

limited

Specifies the number of times the rule runs.

unlimited

Specifies that the rule can run an unlimited number of times.

disabled

Specifies that the rule limit option is disabled.

-limit_value value

(Option) Defines the limit value. This option is valid only when you use it with the limit option set to *limited*.

Limits: 0 and greater

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Rule Retries

This example sets the limit to 40 retries for a rule that runs on a service.

```
dmpolicy setruleexeclimit -entity_type service -entity_name SERVICE1 -rule_name SERVICE1_RULE -limit_option limited -limit_value 40
```

Example: Set Rule Retries to Unlimited

This example sets no limit on the number of times the rule can run on a server.

```
dmpolicy setruleexeclimit -entity_type server -entity_name machine1.ca.com -rule_name MACHINE1_RULE -limit_option unlimited
```

Example: Disable the Limit Option

This example disables the limit option for this rule.

```
dmpolicy setruleexeclimit -entity_type server -entity_name machine2.ca.com -rule_name MACHINE2_RULE -limit_option disabled
```

Policy Configuration

You can use the caismutility CLI to add new functionality to the Policy Configuration user interface.

caismutility Command -- Add Functionalities to Policy Configuration UI

Use the caismutility command to provide functionality policies to configure the user interface. Functionality policies include:

- Import the existing eHealth templates to CA Server Automation as Policy Configuration templates. The Policy Configuration templates are updated on the CA Server Automation UI and applied to SystemEDGE agents.
- Instruct SystemEDGE to report to a new Distribution Server during the following condition:
 - The old Distribution Server is decommissioned.
 - Additional Distribution Servers are added.
 - Moved to a different location where another Distribution Server manages the agents.
- Instruct SystemEDGE to report to a new Domain Server during the following condition:
 - The old Domain Server is decommissioned.
 - A parallel upgrade is performed on CA Server Automation.
- Import and export the SystemEDGE policies and templates, and SRM policies to the Domain Server during the following conditions:
 - The new policies and templates are created on test computers are migrated to a production computer.
 - A parallel upgrade is performed on CA Server Automation.

This command has the following format:

```
caismutility
[-f server]
[-o sysEDGE|sysEDGE|SRM]
[-l]
[-d -p policy]
[-importEH dir|file]
[-useDirName]
[-overwrite]
[-c manager agent1]
[-user user]
[-password password]
[-locale locale]
[-import dir|file]
[-type SEPolicy|SETemplate|SRMPolicy|SRMThreshold|SRMTest|auto]
[-export dir]
```

Note: All options are case-sensitive.

-f *server*

Specifies to connect to a domain server.

-o {*sysEDGE|sysEDGESRM*}

Specifies the type of configuration to use.

-l

Lists the objects that are available.

-d {-p *policy*}

Delivers the policy to an agent.

-importEH {*dir|file*}

Imports the eHealth templates.

-useDirName

Specifies the parent directory name.

-overwrite

Specifies to overwrite an existing template.

-c manager {*agent1 ...*}

Instructs the agents to report to the new manager.

-user *user*

Specifies the user name.

-password *password*

Specifies the password.

-locale *iso639value*

Specifies an ISO 639_3166 combination (for example: fr_FR for French) to override the default English output. To use the locale of the command prompt, specify "native".

-import {*dir|file*}

Specifies the policies or templates to add to filestore.

-type {*SEPolicy|SETemplate|SRMPolicy|SRMThreshold|SRMTest|auto*}

Specifies the type of file to import.

-export *dir*

Copies all policies and templates to the specified directory.

Example: To import a set of eHealth templates

This example imports a set of eHealth templates.

```
caismutility -user xxx -password yyy -importEH c:\templates\
```

Example: To import SystemEDGE policies into a domain server

This example imports some SystemEDGE policies into this domain server.

```
caismutility -user xxx -password yyy -type SEPolicy -import c:\policies
```

Example: Instruct the agent to report to a new manager

This example moves an agent on host agent.com to a new manager manager.com.

```
caismutility -user xxx -password yyy -o sysEDGE -c manager.com agent.com
```

CA Process Automation Commands

You can use the dpmitpamadapter CLI to script and automate CA Process Automation commands and run actions based on the command results.

dpmitpamadapter checkstatus Command--Check Status

The dpmitpamadapter checkstatus command verifies the status of the CA Process Automation process.

This command has the following format:

```
dpmitpamadapter checkstatus [-sc sc_url] -interactionid ID [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-interactionid *ID*

Specifies the interaction ID of a running process.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Verify the Status of Process

This example verifies the status of the CA Process Automation process.

```
dpmitpamadapter checkstatus 78
```

dpmitpamadapter getform Command--Get Form

The dpmitpamadapter getform command obtains all the required input and values necessary for the form to run the process.

This command has the following format:

```
dpmitpamadapter getform [-sc sc_url] -formname form name -refpath ref path [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-formname *form name*

Defines the name of the form that you used for the process.

-refpath *ref path*

Specifies the reference path for the form.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get a Form

This example obtains a form and all of its associated information.

```
dpmitpamadapter getform -formname ConfigurationAuditForm -refpath /Processes/MyProcess
```

dpmitpamadapter listforms Command--List Forms

The dpmitpamadapter listforms command obtains the list of all requested CA Process Automation forms. Forms are the entry point to the process.

This command has the following format:

```
dpmitpamadapter listforms [-sc sc_url][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get the List of Requested Forms

This example obtains the list of requested forms.

```
dpmitpamadapter listforms
```


dpmitpamadapter startprocess Command--Start Process

The dpmitpamadapter startprocess command starts a CA Process Automation process.

This command has the following format:

```
dpmitpamadapter startprocess [-sc sc_url] -formname form name -refpath ref path [-pre] [-post] [-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-formname *form name*

Defines the name of the CA Process Automation form that you used for the process.

-refpath *ref path*

Specifies the reference path for the form.

-inputparam {*parameter name*|*parameter value*}

Specifies the parameter name and value associated with the process you want to start.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Start a Process

This example starts a process.

```
dpmiTpmaadapter startprocess -formname GetJobForm -refpath /Processes/MyProcess  
-inputparam Job_ID|89
```

Remote Monitoring Commands

Remote Monitoring provides a command-line interface (rmonwatch) to the RM AIM that you can use to automate tasks. For example, you can use it to assign a Remote Monitoring system to a Remote Monitoring configuration set.

This command has the following format:

```
rmonwatch [options] getVersion | {add|del} configfile | getTable tablename
```

rmonwatch add Command--Add Rows to a MIB Table

Use the add command to add rows to the rmonWbemSysTable and rmonWbemCredTable MIB tables.

The add command reads a configuration file containing details about the systems and credentials entries. This file uses an .ini style format where each section specifies a system or credentials entry. Because section names must be unique, the term `_<n>` is appended with `<n>` starting from 1. Each section contains variable and value assignments for the configurable attributes of the respective MIB table.

Example: A sample configuration file (rmonwatch.cf)

```
[System_1]
SystemName = sys01-01
RowStatus = active
ConfigSet = default
Credentials = test
MaxInstances = 100
[Credentials_1]
Credentials = test
RowStatus = active
UserName = Administrator
Password = IE7vse8zkBzurAwz5880vyTpc9v
```

The Password attribute of a credentials entry must be specified in encrypted form. Use the CA SystemEDGE `se_enc` utility to encrypt a password as follows:

```
se_enc -s -i in.txt -o out.txt
```

Put the password to be encrypted into `in.txt` (do not add any new lines). After calling the command, copy the encrypted password from `out.txt` and use it as the value for the Password attribute.

The add command uses the following syntax:

```
rmonwatch [options] add configfile
```

rmonwatch uses the following options:

--version

Displays the program version number and exits.

--help

Displays the rmonwatch help

-h hostname | -h ipAddr

(Optional) Specifies the CA SystemEDGE host running the RM AIM.

Default: localhost

-p port

(Optional) Specifies the CA SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP read/write community string for SNMP version 1 and 2c.

Default: public

Example: Using the add command to add table rows defined in rmonwatch.cf

```
rmonwatch -c admin add rmonwatch.cf
RMONWBEM AIM Watch Program, version 1.0.0
### Processing Credentials entry 'test'
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.12.1.3.4.116.101.115.116 -s Administrator
1.3.6.1.4.1.546.16.22.12.1.3.4.116.101.115.116 Administrator
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.12.1.4.4.116.101.115.116 -s IE7vse8zkBzurAwz5880vyTpc9v=
1.3.6.1.4.1.546.16.22.12.1.4.4.116.101.115.116 IE7vse8zkBzurAwz5880vyTpc9v=
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 -i 1
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 1
### Processing System entry ' sys01-01'
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.3.8.115.121.115.48.49.45.48.49 -s "default"
1.3.6.1.4.1.546.16.22.11.1.3.8.115.121.115.48.49.45.48.49 default
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.4.8.115.121.115.48.49.45.48.49 -s "test"
1.3.6.1.4.1.546.16.22.11.1.4.8.115.121.115.48.49.45.48.49 test
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.5.8.115.121.115.48.49.45.48.49 -i 1
1.3.6.1.4.1.546.16.22.11.1.5.8.115.121.115.48.49.45.48.49 1
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.6.8.115.121.115.48.49.45.48.49 -i 100
1.3.6.1.4.1.546.16.22.11.1.6.8.115.121.115.48.49.45.48.49 100
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 -i 1
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 1
```

rmonwatch del Command--Delete Rows from a MIB Table

Use the del command to delete rows from the rmonWbemSysTable and rmonWbemCredTable MIB tables.

The del command reads a configuration file containing details about the systems and credentials entries. This file uses an .ini style format where each section specifies a system or credentials entry. Because section names must be unique, the term `_<n>` is appended with `<n>` starting from 1. Each section contains variable and value assignments for the configurable attributes of the respective MIB table.

Example: A sample configuration file (rmonwatch.cf)

```
[System_1]
SystemName = sys01-01
RowStatus = active
ConfigSet = default
Credentials = test
MaxInstances = 100
[Credentials_1]
Credentials = test
RowStatus = active
UserName = Administrator
Password = password01
```

The delete command uses the following syntax:

```
rmonwatch [options] del configfile
```

rmonwatch uses the following options:

--version

Displays the program version number and exits.

--help

Displays the rmonwatch help

-h *hostname* | -h *ipAddr*

(Optional) Specifies the CA SystemEDGE host running the RM AIM.

Default: localhost

-p *port*

(Optional) Specifies the CA SystemEDGE SNMP port.

Default: 161

-c *community*

(Optional) Specifies the SNMP read/write community string for SNMP version 1 and 2c.

Default: public

Example: Using the del command to delete table rows defined in rmonwatch.cf

```
rmonwatch -c admin del rmonwatch.cf
```

Note: Deletion of rows only requires the specification of index attributes in the configuration file.

```
RMONWBEM AIM Watch Program, version 1.0.0
### Processing Credentials entry 'test'
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 -i 6
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 6
### Processing System entry ' sys01-01'
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 -i 6
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 6
```

rmonwatch getTable Command--View rmonWbem MIB Tables

Use the getTable command to view the contents of the following MIB tables. The name of the table must include the term printed in bold; case does not matter.

- rmonWbem**CSet**Table
- rmonWbem**Sys**Table
- rmonWbem**Cred**Table
- rmonWbem**Conf**Table
- rmonWbem**Query**Table
- rmonWbem**Inst**Table

The getTable command uses the following syntax:

```
rmonwatch [options] getTable {cset | sys | cred | conf | query | inst}
```

rmonwatch uses the following options:

--version

Displays the program version number and exits.

--help

Displays the rmonwatch help

-h hostname | -h ipAddr

(Optional) Specifies the CA SystemEDGE host running the RM AIM.

Default: localhost

-p port

(Optional) Specifies the CA SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP read/write community string for SNMP version 1 and 2c.

Default: public

Example: getTable command run against the system table

```
rmonwatch getTable sys
RMONWBEM AIM Watch Program, version 1.0.0
E:\sysedge\bin\walktree -h localhost -p 161 -v 1 -c public -m 0 -t 30 -r 3 -o
1.3.6.1.4.1.546.16.22.11
rmonWbemSysSystemName <sys01-01> .sys01-01
rmonWbemSysSystemName <sys02-01-8> .sys02-01-8
rmonWbemSysSystemName <sys03-01-7> .sys03-01-7
rmonWbemSysRowStatus active .sys01-01
rmonWbemSysRowStatus active .sys02-01-8
rmonWbemSysRowStatus active .sys03-01-7
rmonWbemSysDescr <nul> .sys01-01
rmonWbemSysDescr <nul> .sys02-01-8
rmonWbemSysDescr <nul> .sys03-01-7
rmonWbemSysContact <nul> .sys01-01
rmonWbemSysContact <nul> .sys02-01-8
rmonWbemSysContact <nul> .sys03-01-7
rmonWbemSysQualifiers .sys01-01
rmonWbemSysQualifiers .sys02-01-8
rmonWbemSysQualifiers .sys03-01-7
rmonWbemSysConfigSet <default> .sys01-01
rmonWbemSysConfigSet <metricNet,metricFS,metricDisk> .sys02-01-8
rmonWbemSysConfigSet <default> .sys03-01-7
rmonWbemSysCredentials <test> .sys01-01
rmonWbemSysCredentials <win2008> .sys02-01-8
rmonWbemSysCredentials <win7> .sys03-01-7
rmonWbemSysProtocol dcom .sys01-01
rmonWbemSysProtocol dcom .sys02-01-8
rmonWbemSysProtocol dcom .sys03-01-7
rmonWbemSysMaxInstances <100> .sys01-01
rmonWbemSysMaxInstances <100> .sys02-01-8
rmonWbemSysMaxInstances <100> .sys03-01-7
```

```
rmonWbemSysIPAddress <192.168.0.10> .sys01-01
rmonWbemSysIPAddress <192.168.0.11> .sys02-01-8
rmonWbemSysIPAddress <192.168.0.12> .sys03-01-7
rmonWbemSysMACAddress <00:12:3F:64:1D:AE> .sys01-01
rmonWbemSysMACAddress <00:50:56:97:7D:BA> .sys02-01-8
rmonWbemSysMACAddress <00:25:64:A9:98:86> .sys03-01-7
rmonWbemSysAdded <2010-5-17,17:26:35.0,+2:0> .sys01-01
rmonWbemSysAdded <2010-5-17,17:26:37.0,+2:0> .sys02-01-8
rmonWbemSysAdded <2010-5-17,17:26:40.0,+2:0> .sys03-01-7
rmonWbemSysLastBoot <2010-5-17,7:52:25.0,+2:0> .sys01-01
rmonWbemSysLastBoot <2010-5-1,20:53:22.0,+2:0> .sys02-01-8
rmonWbemSysLastBoot <2010-5-1,20:53:22.0,+2:0> .sys03-01-7
rmonWbemSysLastPoll <0d 00:10:41> .sys01-01
rmonWbemSysLastPoll <0d 00:11:11> .sys02-01-8
rmonWbemSysLastPoll <0d 00:11:11> .sys03-01-7
rmonWbemSysLastSuccess <0d 00:10:41> .sys01-01
rmonWbemSysLastSuccess <0d 00:11:11> .sys02-01-8
rmonWbemSysLastSuccess <0d 00:11:11> .sys03-01-7
rmonWbemSysLastError success .sys01-01
rmonWbemSysLastError success .sys02-01-8
rmonWbemSysLastError success .sys03-01-7
rmonWbemSysQueryNumber <14> .sys01-01
rmonWbemSysQueryNumber <17> .sys02-01-8
rmonWbemSysQueryNumber <14> .sys03-01-7
rmonWbemSysQueryFailed <0> .sys01-01
rmonWbemSysQueryFailed <0> .sys02-01-8
rmonWbemSysQueryFailed <0> .sys03-01-7
```

rmonwatch getVersion Command--View Version Information

Use the getVersion command to view the value of the rmonWbemVersion MIB variable.

The getVersion command uses the following syntax:

```
rmonwatch [options] getVersion
```

rmonwatch uses the following options:

--version

Displays the program version number and exits.

--help

Displays the rmonwatch help

-h *hostname* | -h *ipAddr*

(Optional) Specifies the CA SystemEDGE host running the RM AIM.

Default: localhost

-p *port*

(Optional) Specifies the CA SystemEDGE SNMP port.

Default: 161

-c *community*

(Optional) Specifies the SNMP read/write community string for SNMP version 1 and 2c.

Default: public

Example: Results returned for the `getVersion` command

```
RMONWBEM AIM Watch Program, version 1.0.0
E:\sysedge\bin\snmpget -h localhost -p 161 -v 1 -c public -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.1.0
1.3.6.1.4.1.546.16.22.1.0 SysEDGE RMONWBEM AIM, VERSION:1.0.0 BUILD:10137
REQUIRES:5.0.0
```

Advanced SNMP Parameters

The following parameters are identical for `rmonwatch add`, `del`, `getTable`, and `getVersion` commands. These parameters are only required if you are using SNMP version 2c or 3 or if you are using a non-default time-out and retries.

-v *snmpVersion*

(Optional) Specifies the SNMP version. Possible values are:

1

2c

3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-s secLevel

(Optional) Specifies the level of security for SNMPv3. Possible values are (no default):

1 = noAuthNoPriv

2 = AuthNoPriv

3 = AuthPriv

-n contextName

(Optional) Specifies the instance name for a MIBMuxed agent (no default).

-a authPassword

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv (no default).

-A authProtocol

(Optional) Specifies the authentication protocol. Possible values are:

MD5 = authentication protocol HMAC-MD5

SHA = authentication protocol HMAC-SHA

Default: MD5

-x privPassword

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X encryptProtocol

(Optional) Specifies the use of encryption protocol for privacy (no default). Possible values are:

DES – Data Encryption Standard

AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)

-m FIPSmode

(Optional) Specifies the FIPS mode. Possible values are:

0 = non-FIPS

1 = FIPS coexistence

2 = FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP set and get command timeout in seconds.

Default: 5

--timeout=walktreetimeout

(Optional) Specifies the SNMP timeout for the walktree command.

Default: 30

-r snmpretries

(Optional) Specifies the number of times to retry the SNMP command.

Default: 3

Reservation Manager Commands

dpmrm CLI commands let you manage the Reservation Manager portal.

dpmrm cancel Command--Cancel a Reservation

The cancel command cancels a reservation.

This command has the following format:

```
dpmrm cancel [-sc sc_url] -resid reservationid [-ws_user username -ws_password
password] [-prompt yes|no] [-locale iso639value] [-pre] [-post]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-resid *reservationid*

Defines the reservation identification number. Valid entry: positive integer.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-prompt yes | no

Defines whether to prompt the user for username and password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Cancel a Reservation

In this example, the reservation 4321 is canceled after prompting the user for validation.
dpmrm cancel -resid 4321 -prompt yes

dpmrm checkreservation and updatereservation Commands

The checkreservation command examines reservation records to see if there is an ESX hostname or VC pool mismatch between the Reservation Manager DB records and the actual VMware environment. The result is output to the console.

The updatereservation command provides check and update functionality to reservation records. The check functionality is the same as the checkreservation command. When a DB record mismatch is found, the DB record is updated to have the correct ESX hostname and VC pool model path.

```
dpmrm checkreservation -resid reservationid | * [-ws_user username -ws_password password] [-prompt yes | no]
```

```
dpmrm updatereservation -resid reservationid | * [-ws_user username -ws_password password] [-prompt yes | no]
```

-resid *reservationid*

Indicates a specific reservation ID or '*' for all reservation IDs.

-ws_user *username*

(Optional) Identifies the security user name.

-ws_password *password*

(Optional) Identifies the security user password.

-prompt *yes | no*

(Optional) Indicates prompt for username and password. The default is yes.

Examples

To check reservation 18 for ESX hostname and VC pool DB record consistency:

```
dpmrm checkreservation -resid 18 -ws_user admin -ws_password ca_admin -prompt no
```

To check all reservations for ESX hostnames and VC pools DB record consistency:

```
dpmrm checkreservation -resid * -ws_user admin -ws_password ca_admin -prompt no
```

To check and update reservation 18 for ESX hostname and VC pool DB record mismatches:

```
dpmrm updatereservation -resid 18 -ws_user admin -ws_password ca_admin -prompt no
```

To check and update all reservations for ESX hostname and VC pool DB record mismatches:

```
dpmrm updatereservation -resid * -ws_user admin -ws_password ca_admin -prompt no
```

dpmrm createfromtemplate Command--Create Reservation

The createfromtemplate command creates a reservation using an existing template.

This command has the following format:

```
dpmrm createfromtemplate -template template -startdate startdate -starttime starttime -enddate enddate -endtime endtime -username username [-sc sc_url] -orgunit orgunit [-notes notes] [-email_address emailaddress] [-projectid projectid] [-ws_user username -ws_password password] [-prompt yes|no] [-locale iso639value] [-pre] [-post]
```

-template *template*

Specifies the name of a reservation defined in Reservation Manager. Valid entry: 1-256 alphanumeric characters.

-startdate *startdate*

Defines the start date for the reservation. Valid entry: mm/dd/yyyy.

-starttime *starttime*

Defines the start time for the reservation. Valid entries: 0-23.

Note: If you define a start time within the current hour, always select the *next hour*. For example, if the current time is 12:23pm Pacific time, and you enter Startday=7/13/2010 and Starttime=12, the time is viewed as already passed, and is not allowed. In this case, select Starttime=13 or above.

-enddate *enddate*

Defines the end date for the reservation. Valid entry: mm/dd/yyyy.

-endtime *endtime*

Defines the end time for the reservation. Valid entry: 0-23.

-username *username*

Defines the user name for web security check.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-orgunit *orgunit*

Defines the organizational unit for the reservation. Not required if the user is a member of only one organizational unit. Valid entry: 1-256 alphanumeric characters.

-notes *notes*

(Optional) Defines notes for creating a reservation. Valid entry: 1-256 alphanumeric characters.

-emailaddress *emailaddress*

(Optional) Defines the user email address for the reservation.

-projectid *projectid*

(Optional) Defines the project identification for the reservation.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

-prompt *yes|no*

(Optional) Defines whether to prompt the user for username and password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Create a Reservation

In this example, the existing template, `WindowsMachine` is used to create a reservation.

```
dpmrm createfromtemplate -template WindowsMachine -startdate 02/02/2010
-starttime 1 -enddate 03/02/2010 -endtime 4 -username cartog123 -orgunit 5467
```

dpmrm createsoftwaregroup Command

Use this command to create a software group.

```
dpmrm createsoftwaregroup [-sc sc_url] -name groupname -package
packagename[:procedurename] [-description description] [-category category]
[-orgunit orgunit] [-systemimage systemimagename] [-ws_user username] [-ws_password
password] [-prompt yes|no] [-pre] [-post] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-name *groupname*

Specifies the unique name of the software group to create.

-package *packagename*

Identifies one or more software delivery package names. This parameter is a two-part data value in the format of *packagename:procedurename*. If *procedure name* is missing, the default is "Install".

-description *description*

(Optional) Contains the description for the software group.

-category *category*

(Optional) Specifies the Category name of the software group.

-orgunit *orgunit*

(Optional) Contains zero or more organizational units that can access the software group.

-systemimage *systemimagename*

(Optional) Contains zero or more system images to be associated with this software package group.

-ws_user *username*

(Optional) Identifies the security user name.

-ws_password *password*

(Optional) Identifies the security user password.

-prompt *yes|no*

(Optional) Specifies prompt for username and password. The default is yes.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Examples

Multiple Package and Procedure names with spaces, one organizational unit and one system image.

```
dpmrm createsoftwaregroup -name testgroup -package "Package 1":"Install Proc"
-package "Package 2":"uninstall proc" -orgunit public -systemimage w2k8
-ws_user testadmin -ws_password testadmin
```

Multiple Package and Procedure names, multiple organizational units and system images.

```
dpmrm createsoftwaregroup -name testgroup -package "Package 1":"Install Proc"
-package "Package 2":"uninstall proc" -orgunit public -orgunit admin -orgunit "admin
1"
systemimage w2k8 -systemimage "windows 2003" -ws_user testadmin
-ws_password testadmin
```

Single Package with description and no Procedure name.

```
dpmrm createsoftwaregroup -name testgroup -description "Windows utilities"
-package Package1 -ws_user testadmin -ws_password testadmin
```

dpmrm extend Command--Extend a Reservation

The extend command extends a reservation after the original expiration date.

This command has the following format:

```
dpmrm extend -resid reservationid -newdate newdate -newtime newtime [-sc sc_url]
[-ws_user username -ws_password password] [-prompt yes|no] [-pre] [-post]
```

-resid *reservationid*

Defines the reservation identification number. Valid entry: positive integer.

-newdate *newdate*

Defines the new date for extending the reservation. Valid entry: mm/dd/yyyy

-newtime *newtime*

Defines the new time for extending the reservation. Valid entry: 0-23.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

-prompt *yes|no*

Defines whether to prompt the user for username and password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Extend a Reservation

In this example, the reservation 4321 is extended until March 10, 2010 at 10:00 a.m.

```
dpmmr extend -resid 4321 -newdate 02/10/2010 -newtime 10
```

dpmrm importvirtualmachines Command

The `importvirtualmachines` command imports the existing VMware virtual machines to the Reservation Manager so that you can extend, cancel, and return the virtual machine using the Reservation ID.

This command has the following format:

```
dpmrm importvirtualmachines [-sc sc_url] -vcenter vcenter name -datacenter datacenter name -vm vm name[|vm name...] -enddate MM/DD/YYYY -endtime 0-23 -tenant tenant [-orgunit org unit [-reservation_user reservationusername] [-email_address emailaddress] [-ws_user security username] [-ws_password security password] [-prompt yes|no] [-pre] [-post] [-locale is0639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: **80**

-vcenter *vcenter name*

Specifies the vCenter server host name.

-datacenter *datacenter name*

Specifies the vCenter datacenter name.

-vm *vm name*[*|vm name...*]

Specifies the virtual machine names to add to the Reservation manager.

-enddate *enddate*

Defines the end date for the reservation. Valid entry format is `mm/dd/yyyy`.

-endtime *endtime*

Defines the end time for the reservation. Valid entry is `0-23`.

-tenant *tenant*

Defines the name of the tenant.

-orgunit *orgunit*

Defines the organizational unit for the reservation. Not required if the user is a member of only one organizational unit. Valid entry: 1-256 alphanumeric characters.

-reservation_user *reservationusername*

(Optional) Defines the user name for self-service reservation manager.

-emailaddress *emailaddress*

(Optional) Defines the user email address for the reservation.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

-prompt *yes|no*

(Optional) Defines whether to prompt the user for username and password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Import a virtual machine to the Reservation manager.

For example, the following command imports the SA128–SA1 virtual machine to SWAT Reservation manager:

```
dpmrm importvirtualmachines -vcenter VC.org.com -datacenter SWAT -vm "SA128-SA1"
-enddate 08/05/2013 -endtime 5 -orgunit SWAT -ws_user admin -ws_password demo123
```

dpmrm listsoftwaregroups Command

The `listsoftwaregroups` command provides a list of software groups that are available to the specific organization unit.

```
dpmrm listsoftwaregroups -orgunit orgunit [-sc sc_url] [-ws_user username]
[-ws_password password] [-prompt yes|no] [-pre] [-post] [-locale iso639value]
```

-orgunit *orgunit*

Identifies the organizational unit that can access the software group.

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username*

(Optional) Identifies the security user name.

-ws_password *password*

(Optional) Identifies the security user password.

-prompt *yes | no*

(Optional) Indicates prompt for username and password. The default is yes.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Examples

```
dpmrm listsoftwaregroups -orgunit public -ws_user testadmin -ws_password testadmin
```

The list shows each package group with an identifier (0-n) and each package:procedure combination with a unique identifier (0-n) similar to the following:

```
Software Group 0: name = Windows Utilities
Software Group 0:Package 0 = SQL Server 2005
Software Group 0:Procedure 0 = Install
Software Group 0:Package 1 = Putty
Software Group 0:Procedure 1 = Install
Software Group 0:Package 2 = WinSCP
Software Group 0:Procedure 2 = Install
Software Group 1: name = Backup Software
Software Group 1:Package 0 = Fast Backup
Software Group 1:Procedure 0 = Install
```

dpmm listtemplates Command--Display List Reservation Templates

The listtemplates command displays the list reservation templates.

This command has the following format:

```
dpmm listtemplates [-sc sc_url] -orgunit orgunit [-ws_user username] [-ws_password password] [-locale iso639value] [-prompt yes|no] [-pre] [-post]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-orgunit *orgunit*

Specifies the organization units.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-prompt yes|no

(Optional) Specifies whether to prompt the user for username and password.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

Example: Display the list reservation templates

This example displays the list reservation templates for an organizational unit.

```
dpmrm listtemplates -orgunit Public -ws_user testadmin -ws_password adminpw
```

Resource Manager Commands

You can use the `dpmresourcemgr` CLI to script and automate Resource Manager commands and run actions based on the command results.

`dpmresourcemgr addbaremetal` Command--Add a Physical Computer

The `addbaremetal` command adds a physical computer to the data center.

This command has the following format:

```
dpmresourcemgr addbaremetal [-sc sc_url] -server_name hostname -MAC_address macaddress [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *source host name*

Defines the name of the source server in the operation.

-MAC_address *macaddress*

Defines the MAC address of the target computer. Valid entry: 12 digit hexadecimal values (for example, A-F, 0-9, a-f) in the format: 0013724C2140 or 00:13:72:4C:21:40.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Add a Physical Computer

This example adds computer baremetal2 with a MAC address of 00:13:72:4C:21:40.

```
dpmresourcemgr addbaremetal -sc https://servicecontroller/dpm/sc -server_name  
baremetal2 -MAC_address 00:13:72:4C:21:40 -ws_user wsuser -ws_password  
wsuserpassword
```

dpmresourcemgr addcomputertonetwork Command--Add Server to Network

The dpmresourcemgr addcomputertonetwork command adds servers to an existing network.

This command has the following format:

```
dpmresourcemgr addcomputertonetwork [-sc sc_url] -vlan_id vlanid -mac_list  
macaddresses [-vlan_tagged][-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-vlan_id *vlanid*

Defines the integer for the virtual network (VLAN). Valid entry: Integer 1-4094, where 1 is the default network interface card (NIC).

-mac_list *macaddresses*

Defines a list of NIC MAC addresses for the server. Valid entry: Comma-separated NIC MAC addresses in the format, xx:xx:xx:xx:xx:xx.

-vlan_tagged

Defines the VLAN as tagged. If not specified, the VLAN is untagged.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Add Server to Network

This example adds a server to the VLAN 4556.

```
dpmresourcemgr addcomputertonetwork -vlan 4556 -mac_list 11:22:33:44:55:66 -ws_user
wsuser -ws_password wsuserpassword
```

dpmresourcemgr addaddresspool Command--Add IP Address Pools

The `dpmresourcemgr addaddresspool` command adds an IP address pool using static IP addresses or DHCP.

This command has the following format:

```
dpmresourcemgr addaddresspool [-sc sc_url] -subnet_address address -starting_ip
ipaddress -ending_ip ipaddress -ip_assignment_type static|dhcp [-name name][-owner
ownername][-ws_user username -ws_password password][-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subnet_address address

Defines the IP address of the network. Valid entry: `xxx.xxx.xxx.xxx` or `xxx.xxx.xxx.xxx/xx`.

-starting_ip ipaddress

Defines the starting valid IP address in the range for use in the static IP address pool. Valid entry: `xxx.xxx.xxx.xxx`.

-ending_ip ipaddress

Defines the ending IP address in an IP range. Valid entries: `xxx.xxx.xxx.xxx`.

-ip_assignment_type static | dhcp

Defines the method in which servers get their IP addresses (static IP address or DHCP).

-name name

Defines the item name. Valid entry: 1-128 alphanumeric characters.

-owner ownername

Defines the owner of the IP address pool. Valid entry: 1-128 alphanumeric characters.

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Add IP Address Pool

This example adds the IP address pool, `engpool`, with the IP address range of 10.10.21.130 to 10.10.21.140 and sets the IP assignment method to DHCP.

```
dpmresourcemgr addipaddresspool -subnet_address 10.10.21.0 -starting_ip
10.10.21.130 -ending_ip 10.10.21.140 -ip_assignment_type DHCP -ws_user wsuser
-ws_password wsuserpassword
```

dpmresourcemgr addmachinestosome Command--Add Servers to a Service

The `addmachinestosome` command adds servers to a service.

This command has the following format:

```
dpmresourcemgr addmachinestosome [-sc sc_url] -service_name servicename
-machine_list machinelist [-ws_user username -ws_password password] [-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-service_name *servicename*

Defines the name of the service. Valid entry: Full path to the service name. Use backslash `\` to delimit parent service and sub service.

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Add Servers to a Service

This example adds a list of servers to a service named SERVICEA.

```
dpmresourcemgr addmachinestoservice -service_name SERVICEA -machine_list  
"machine4.MyCompany.com, machine5.MyCompany.com, 192.168.255.255" -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr addtocontainer Command--Add Content to a Container

The dpmresourcemgr addtocontainer command adds servers or a subcontainers to a container.

This command has the following format:

```
dpmresourcemgr addtocontainer [-sc sc_url] -container_name containername  
[-machine_list machinelist|-subcontainer_name subcontainername] [-ws_user username  
-ws_password password][-locale iso639value]
```

-container_name *containername*

Defines a collection of objects. Valid entry: the full path to the container name using backslash (\) to delimit parent container and child container.

-machine_list *machinelist* | -subcontainer *subcontainername*

Defines the machine list or subcontainer and its objects.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Add Content to a Container

This example adds a subcontainer named, ManagedVMs, to the container named, Managed.

```
dpmresourcemgr addtocontainer -container_name Managed -subcontainer_name  
ManagedVMs.
```

dpmresourcemgr createcontainer Command--Create a Container

The createcontainer command creates an object to hold a collection of other objects.

This command has the following format:

```
dpmresourcemgr createcontainer [-sc sc_url] -parent_container parentcontainer  
-container_name containername -machine_list machinelist [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-parent_container *parentcontainer*

Defines the full path service name for the parent service.

-container_name *containername*

Defines a collection of objects. Valid entry: the full path to the container name using backslash (\) to delimit parent container and child container.

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a Container

This example creates a container named, ManagedVMs.

```
dpmresourcemgr createcontainer -parent_container Managed -container_name ManagedVMs
```

dpmresourcemgr createdefaultservice Command--Create Default Service

The `createdefaultservice` command creates the default service with the local system and any external systems configured during the installation.

This command has the following format:

```
dpmresourcemgr createdefaultservice [-sc sc_url] [-retry retrynumber]  
[-discover_local_subnet] [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-retry *retrynumber*

Defines the number of retries for creating the service.

-discover_local_subnet

Automatically discovers the local subnet after creating a default service if specified.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Create a Default Service

This example creates a default service that tries to connect ten times, and automatically discovers the local subnet.

```
dpmresourcemgr createdefaultservice -retry 10 -discover_local_subnet -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr createipnetwork Command--Create Network

The dpmresourcemgr createipnetwork command specifies an existing network in your datacenter for use with Reservation Manager. The Reservation Manager end user can then reserve a virtual machine on a specific network.

This command has the following format:

```
dpmresourcemgr createipnetwork [-sc sc_url] -subnet_address subnetaddress -vlan_id vlanid -ip_netmask ipnetmask -name networkname -ip_gateway_list ipaddresses -dns_domainname domainname [-description description] [-dns_list ipaddresses][-dns_suffix_list suffixes][-netbios_server_list ipaddresses][-ntp_server_list ipaddresses][-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-subnet_address *address*

Defines the IP address of the network. Valid entry: xxx.xxx.xxx.xxx or xxx.xxx.xxx.xxx/xx.

-vlan_id *vlanid*

Defines the integer for the virtual network (VLAN). Valid entry: Integer 1-4094, where 1 is the default network interface card (NIC).

-ip_mask *ipnetmask*

Defines the subnet mask. Valid entry: Standard IP netmask in the format: xxx.xxx.xxx.x, or CIDR format: xxx.xxx.xxx.x/8-28. Prefix length is 8-28.

-name *name*

Defines the item name. Valid entry: 1-128 alphanumeric characters.

-ip_gateway_list *ipaddresses*

Defines the IP addresses for default and alternate gateways using a comma-separated list. Default and alternate gateways must be on the same network. Valid entry: xxx.xxx.xxx.xxx.

-dns_domainname *domainname*

Defines the DNS domain name of the network. Valid entry: 1-63 alphanumeric characters per dotted address (for example, xxx.yyy.com). Address cannot start with a number.

-description *description*

Defines the network for user understanding. Valid entry: 1-128 alphanumeric characters.

-dns_list *ipaddresses*

Defines the DNS list of preferred and alternate DNS servers, and primary and secondary Windows Internet Name Service (WINS) servers. Valid entry: Comma-separated IP addresses in the format: xxx.xxx.xxx.xxx.

-dns_suffix_list *suffixes*

Defines a list of DNS suffixes for the network. Valid entry: Comma-separated list in the format, *name.name.com*.

-netbios_server_list *ipaddresses*

Defines a list of NETBIOS Name Server IP addresses. Valid entry: Comma-separated list of IP addresses in the format: xxx.xxx.xxx.xxx.

-ntp_server_list *ipaddresses*

Defines the Network Time Protocol (NTP) servers. Valid entry: Comma-separated list with the format: xxx.xxx.xxx.xxx.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create a Network

This example specifies the existing datacenter network, san diego 209 subnet, for use by the Reservation Manager end-user portal when reserving a virtual machine.

```
dpmresourcemgr createipnetwork -subnet_address 10.10.21.0 -vlan_id 4001 -ip_mask 255.255.255.0 -name "san diego 209 subnet" -ip_gateway_list 10.10.21.124,10.10.21.125 -dns_domainname turbo.com -ws_user wsuser -ws_password wsuserpassword
```


dpmresourcemgr createresourcegroup Command--Create Resource Group

The createresourcegroup command creates a package group.

This command has the following format:

```
dpmresourcemgr createresourcegroup [-sc sc_url] -group_name groupname  
-group_description description -group_type type -group_attribute attribute  
-package_list packagelist [-ws_user username -ws_password password] [-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-group_name *groupname*

Defines the name of the group.

-group_description *description*

Defines the description of the group.

-group_type *type*

Defines the type of resource group. Set to 1 for packaging group. Set to zero for all groups.

-group_attribute *attribute*

Defines the group attribute.

-package_list *packagelist*

Defines individual packages.

Example: `-package_list "<package name A>|<package procedure A>|<procedure type A>|<package name B>|..."`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create Resource Group

This example creates a resource group.

```
dpmresourcemgr createresourcegroup -group_name "AGENT PACKAGE GROUP"
-group_description "Agent package group" -group_type 1 -group_attribute 0
-package_list "CA Product - Win r11.2|Install|INSTALL|Generic Agent Win32
r5.0|Install|INSTALL"
```

dpmresourcemgr createresourcetemplate Command--Create Server Group Template

The createresourcetemplate command creates a package template.

This command has the following format:

```
dpmresourcemgr createresourcetemplate [-sc sc_url] -template_name templatename
-template_description description -template_os_type OStype [-group_list grouplist]
[-package_list packagelist][-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-template_name *templatename*

Defines the name of the template.

-template_description *description*

Defines a description of the template.

-template_os_type *OStype*

Defines the OS type of the resource template for deployment.

-group_list *grouplist*

Defines list of groups for adding to the resource template. Valid entry: -group_list "<group name A>|<group type 1=Software Package Group>|<group name B>|... "

-package_list *packagelist*

Defines individual packages. Define this option if the group type is set to 1(PACKAGE). Valid entries: -package_list "<package name A>|<package procedure A>|<procedure type A>|<package name B>|... "

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Create Resource Template

This example creates a package template.

```
dpmresourcemgr createresourcetemplate -template_name "Generic template"
-template_description "Generic template description" -template_os_type 0 -group_list
"Package group A|1|Package group B|1" -package_list "CAProduct - Win
r11.2|Install|INSTALL|Generic Agent Win32 r5.0|Install|INSTALL"
```

dpmresourcemgr createservice Command--Create a Service

The createservice command creates a service. A service can have multiple sub services.

This command has the following format:

```
dpmresourcemgr createservice [-sc sc_url] -service_name servicename [-machine_list
machinelist] [-lower_threshold lowerthreshold] [-upper_threshold upperthreshold]
[-lag lag] [-priority priority] [-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-service_name *servicename*

Defines the name of the service. Valid entry: Full path to the service name. Use backslash \ to delimit parent service and sub service.

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-lower_threshold *lowerthreshold*

(Optional) Specifies the lower overall usage threshold for the service. Lower threshold must be less than upper threshold. Valid entry: 1-100 (%)

-upper_threshold *upperthreshold*

(Optional) Specifies the upper overall usage threshold for the service. Upper threshold must always be greater than lower threshold. Valid entry: 1-100 (%)

-lag *lagvalue*

(Optional) Specifies the lag value. Valid entry: 0 or higher (seconds).

-priority *priorityvalue*

(Optional) Specifies the priority for the service. Valid entry: 1 or higher.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Create a Service

This example creates a service, named SERVICE1.

```
dpmresourcemgr createservice -service_name SERVICE1 -machine_list
"ComputerList.MyCompany.com, 172.31.255.255, Computer1.MyCompany.com"
-lower_threshold 20 -upper_threshold 80 -lag 1 -priority 1 -ws_user wsuser
-ws_password wsuserpassword
```

dpmresourcemgr deletebaremetal Command--Delete a Physical Computer

The deletebaremetal command deletes a physical computer from the data center.

This command has the following format:

```
dpmresourcemgr deletebaremetal [-sc sc_url] -server_name hostname -MAC_address
macaddress [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *source host name*

Defines the name of the source server in the operation.

-MAC_address *macaddress*

Defines the MAC address of the target computer. Valid entry: 12 digit hexadecimal values (for example, A-F, 0-9, a-f) in the format: 0013724C2140 or 00:13:72:4C:21:40.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a Physical Computer

This example deletes the EngServer1 server from the data center.

```
dpmresourcemgr deletebaremetal -server EngServer1 -MAC_address 11:22:33:44:55:66  
-ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr deletecontainer Command--Delete a Container

The deletecontainer command deletes a container and its contents.

This command has the following format:

```
dpmresourcemgr deletecontainer [-sc sc_url] -container_name containername  
[-recursive] [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-container_name *containername*

Defines a collection of objects. Valid entry: the full path to the container name using backslash (\) to delimit parent container and child container.

-recursive

If set, the command is run recursively to sub services or containers.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a Container

This example creates a container named, ManagedVMs.

```
dpmresourcemgr deletecontainer -container_name ManagedVMs.
```

dpmresourcemgr deleteipnetworks Command--Delete a Network

The `dpmresourcemgr deleteipnetworks` command deletes a network from CA Server Automation, but only if there are no current or existing reservations that use the network.

This command has the following format:

```
dpmresourcemgr deleteipnetworks [-sc sc_url] -subnet_address_list subnetaddress [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subnet_address_list *addresses*

Defines a list of networks by IP addresses. Valid entry: Comma-separated IP addresses in the format `xxx.xxx.xxx.xxx` or `xxx.xxx.xxx.xxx/xx`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Delete a Network

This example deletes the san diego 209 subnet for use by CA Server Automation.

```
dpmresourcemgr deleteipnetworks -subnet_address_list 10.10.21.0,10.10.21.130  
-ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr deletesystem Command--Delete a System

The deletesystem command deletes servers from the data center.

This command has the following format:

```
dpmresourcemgr deletesystem [-sc sc_url] -machine_list machinelist [-post] [-ws_user  
username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Delete a System

This example deletes the server named, ServerB.

```
dpmresourcemgr deletesystem -machine_list "machine4.MyCompany.com,  
machine5.MyCompany.com, 192.168.255.255" -ws_user wsuser -ws_password  
wsuserpassword
```

dpmresourcemgr getdiscoverednetworks Command--Get Discovered Networks

The getdiscoverednetworks command retrieves a list of all discovered networks.

This command has the following format:

```
dpmresourcemgr getdiscoverednetworks [-sc sc_url][-ws_user username -ws_password  
password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Retrieve All Networks

This example retrieves a list of all discovered networks.

```
dpmresourcemgr getdiscoverednetworks -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getipaddresspools Command--List IP Address Pools

The dpmresourcemgr getipaddresspools command lists existing the IP address pools by owner or assignment type.

This command has the following format:

```
dpmresourcemgr getipaddresspools [-sc sc_url] -subnet_address subnetaddress [-owner ownername][-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subnet_address *address*

Defines the IP address of the network. Valid entry: `xxx.xxx.xxx.xxx` or `xxx.xxx.xxx.xxx/xx`.

-owner *ownername*

Defines the owner of the IP address pool. Valid entry: 1-128 alphanumeric characters.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Update IP Address Pool

This example lists the existing IP address pools for the network at IP address 10.10.21.30.

```
dpmresourcemgr getipaddresspools -subnet_address 10.10.21.130 -owner Reservation
Manager -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getipnetworks Command--List Networks

The `dpmresourcemgr getipnetworks` command lists the networks specified in CA Server Automation using the `createipnetwork` command.

This command has the following format:

```
dpmresourcemgr getipnetworks [-sc sc_url][-ws_user username -ws_password
password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: List Networks

This example lists the networks for use by CA Server Automation.

```
dpmresourcemgr getipnetworks -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getserverlist Command--Get a List of Servers

The `getserverlist` command retrieves a list of servers from the data center.

This command has the following format:

```
dpmresourcemgr getserverlist [-sc sc_url] -service_name servicename [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-service_name *servicename*

Defines the name of the service. Valid entry: Full path to the service name. Use backslash \ to delimit parent service and sub service.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Get a List of Servers

This example retrieves a list of servers that belong to the service, `SERVICE2`.

```
dpmresourcemgr getserverlist -service_name SERVICE2 -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getservicelist Command--Get List of Services

The getserverlist command retrieves a list of services from a server.

This command has the following format:

```
dpmresourcemgr getservicelist [-sc sc_url][-parent_service_name  
parentservicename][-server_name servername][-recursive][-ws_user username -ws_password  
password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-parent_service_name *parentservicename*

Defines a parent service. Valid entries: the full path service to the parent service.

-server_name *servername*

Defines the name of the server. If not specified, the command returns a list of all defined services.

-recursive

If set, the function is executed recursively to subservices.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get List of Services

This example retrieves a list of services that belong to the parent service, SERVICE2 and the server, EngServer1.

```
dpmresourcemgr getservicelist -parent service_name SERVICE2 -server_name EngServer1  
-ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getsubnetdefaults Command--List Subnet Defaults

The dpmresourcemgr getsubnetdefaults command lists the current network defaults.

This command has the following format:

```
dpmresourcemgr getsubnetdefaults [-sc sc_url] -subnet_address address [-ws_user  
username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subnet_address *address*

Defines the IP address of the network. Valid entry: `xxx.xxx.xxx.xxx` or `xxx.xxx.xxx.xxx/xx`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: List Network Defaults

This example updates the defaults for the network with the IP address, 10.10.21.130.

```
dpmresourcemgr getsubnetdefaults -subnet_address 10.10.21.130 -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr removecomputerfromipnetwork Command--Remove Server From Network

The `dpmresourcemgr removecomputerfromipnetwork` removes servers from a network.

This command has the following format:

```
dpmresourcemgr removecomputerfromipnetwork [-sc sc_url] -vlan_id vlanid -mac_list  
macaddresses [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-vlan_id *vlanid*

Defines the integer for the virtual network (VLAN). Valid entry: Integer 1-4094, where 1 is the default network interface card (NIC).

-mac_list *macaddresses*

Defines a list of NIC MAC addresses for the server. Valid entry: Comma-separated NIC MAC addresses in the format, `xx:xx:xx:xx:xx:xx`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Remove Server From Network

This example removes the server, 11:22:33:44:55:66 from the VLAN 4556.

```
dpmresourcemgr removecomputerfromipnetwork -vlan 4556 -mac_list 11:22:33:44:55:66  
-ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr removefromcontainer Command--Remove Content from a Container

The removefromcontainer command removes a list of machines or a subcontainer from a container.

This command has the following format:

```
dpmresourcemgr removefromcontainer [-sc sc_url] -container_name containername  
[-machine_list machinelist|-subcontainer_name subcontainername] [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-container_name *containername*

Defines a collection of objects. Valid entry: the full path to the container name using backslash (\) to delimit parent container and child container.

-machine_list *machinelist* |-subcontainer *subcontainername*

Defines the list of machines or subcontainer and its objects. Valid entries: The full path container for the subcontainer.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Remove Content From a Container

This example removes a subcontainer named, ManagedVMs, from the container, Managed.

```
dpmresourcemgr removefromcontainer -container_name Managed -subcontainer_name ManagedVMs.
```

dpmresourcemgr removeipaddresspool Command--Remove IP Address Pool

The dpmresourcemgr removeipaddresspool command removes the IP address pool.

This command has the following format:

```
dpmresourcemgr removeipaddresspool [-sc sc_url] -name name|-starting_ip ipaddress [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-name *name* | -starting_ip *ipaddress*

Defines the name or starting IP address range. Valid entry: 1-128 alphanumeric characters or xxx.xxx.xxx.xxx.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Remove IP Address Pool

This example removes the IP address pool starting with the address, 10.10.21.130

```
dpmresourcemgr removeipaddresspool -starting_ip 10.10.21.130 -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr removemachinesfromservice Command--Remove Servers From a Service

The removemachinesfromservice command removes servers from a service.

This command has the following format:

```
dpmresourcemgr removemachinesfromservice [-sc sc_url] -service_name servicename  
-machine_list machinelist [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-service_name *servicename*

Defines the name of the service. Valid entry: Full path to the service name. Use backslash \ to delimit parent service and sub service.

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Remove Servers From a Service

This example removes two servers from the service, SERVICE2.

```
dpmresourcemgr removemachinesfromservice -service_name SERVICE2 -machine_list
"machine1.MyCompany.com, machine2.MyCompany.com" -ws_user wsuser -ws_password
wsuserpassword
```

dpmresourcemgr removeresourcegroup Command--Remove Resource Group

The removeresourcegroup command removes a resource group.

This command has the following format:

```
dpmresourcemgr removeresourcegroup [-sc sc_url] -group_name groupname -group_type
grouptype [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-group_name *groupname*

Defines the name of the group.

-group_type *type*

Defines the type of resource group. Set to 1 for packaging group. Set to zero for all groups.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Remove Resource Group

This example removes the resource group named, Agent Package Group.

```
dpmresourcemgr removeresourcegroup -group_name "AGENT PACKAGE GROUP" -group_type 1
```

dpmresourcemgr removeservice Command--Remove a Service

The removeservice command removes a service from the data center.

This command has the following format:

```
dpmresourcemgr removeservice [-sc sc_url] -service_name servicename  
[-recursive][-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-service_name *servicename*

Defines the name of the service. Valid entry: Full path to the service name. Use backslash \ to delimit parent service and sub service.

-recursive

If set, the command is executed recursively to subservices.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Remove a Service

This example removes the service named, TESTGROUP.

```
dpmresourcemgr removeservice -service_name TESTGROUP -ws_user wsuser -ws_password
wsuserpassword
```

dpmresourcemgr removesourcetemplate Command--Remove Resource Template

The removeresourcetemplate command removes a package template.

This command has the following format:

```
dpmresourcemgr removeresourcetemplate [-sc sc_url] -template_name templatename
-template_os_type OStype [-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-template_name *templatename*

Defines the name of the template.

-template_os_type *OStype*

Defines the OS type of the resource template for deployment.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Remove Resource Template

This example removes the package template named, Generic template.

```
dpmresourcemgr removeresourcetemplate -template_name "Generic template"
-template_os_type 0
```

dpmresourcemgr retrieveresourcegroup Command--Retrieve Resource Group

The retrieveresourcegroup command retrieves a package group.

This command has the following format:

```
dpmresourcemgr retrieveresourcegroup [-sc sc_url] [-group_name groupname]
-group_type type [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-group_name *groupname*

Defines the name of the group.

-group_type *type*

Defines the type of resource group. Set to 1 for packaging group. Set to zero for all groups.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Retrieve Resource Group

This example retrieves all package groups of any resource type.

```
dpmresourcemgr retrieveresourcegroup -group_type 0
```

Example: Retrieve All Package Groups

This example retrieves all package group information.

```
dpmresourcemgr retrieveresourcegroup -group_type 1
```

Example: Retrieve Specific Package Group

This example retrieves specific package group information

```
dpmresourcemgr retrieveresourcegroup -group_name "AGENT PACKAGE GROUP" -group_type 1
```

dpmresourcemgr retrieveresourcetemplate Command--Retrieve Resource Template

The retrieveresourcetemplate command retrieves a package template.

This command has the following format:

```
dpmresourcemgr retrieveresourcetemplate [-sc sc_url] [-template_name templatename]
-template_os_type Ostype [-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-template_name *templatename*

Defines the name of the template.

-template_os_type *Ostype*

Defines the OS type of the resource template for deployment. Set to -1 for all OS types.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Retrieve Resource Template

This example retrieves package template information of any OS type.

```
dpmresourcemgr retrieveresourcetemplate -template_name "Generic template"
-template_os_type -1
```

dpmresourcemgr setcontainername Command--Set Container Name

The `setcontainername` command changes the name of an existing container.

This command has the following format:

```
dpmresourcemgr setcontainername [-sc sc_url] -container_name containername
-container_new_name containernewname [-ws_user username -ws_password
password][-locale iso639value]
```


-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-container_name *containername*

Defines a collection of objects. Valid entry: the full path to the container name using backslash (\) to delimit parent container and child container.

-container_new_name *containernewname*

Defines the new name of the container.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Set Container Name

This example changes the name of the container from `ManagedVM1` to `ManagedVM3`.
`dpmresourcemgr setcontainername -container_name ManagedVM -container_new_name ManagedVM3.`

dpmresourcemgr setcontainerservice Command--Set Container to a Service

The `setcontainerservice` command converts a container to a service.

This command has the following format:

```
dpmresourcemgr setcontainerservice [-sc sc_url] -container_name containername
-machine_list machinelist -lower_threshold lowerthreshold -upper_threshold
upperthreshold -lag lag -priority priority [-ws_user username -ws_password
password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-lower_threshold *lowerthreshold*

(Optional) Specifies the lower overall usage threshold for the service. Lower threshold must be less than upper threshold. Valid entry: 1-100 (%)

-upper_threshold *upperthreshold*

(Optional) Specifies the upper overall usage threshold for the service. Upper threshold must always be greater than lower threshold. Valid entry: 1-100 (%)

-lag *lagvalue*

(Optional) Specifies the lag value. Valid entry: 0 or higher (seconds).

-priority *priorityvalue*

(Optional) Specifies the priority for the service. Valid entry: 1 or higher.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Convert a Container to a Service

This example converts a container named, Managed1, into a service.

```
dpmresourcemgr setcontainterservice -container_name Managed1 -machine_list
"ComputerList.MyCompany.com, 172.31.255.255, Computer1.MyCompany.com"
-lower_threshold 20 -upper_threshold 80 -lag 1 -priority 1
```

dpmresourcemgr setmanagementip Command--Set the Management IP for a Host

The dpmresourcemgr setmanagementip command sets the management IP address for a host.

Note: If any duplicate *hostname* exists for the server-name parameter, a console message requests use of the *system_uuid* parameter instead. When this occurs, you can acquire the UUID from the corresponding record of the [_Name] column in the database table [aom2].[dbo].[_CA_ComputerSystem]; or you can get it from the output of "...\\productname\\bin\\caaipaomwsclient.exe /enum=CA_ComputerSystem".

This command has the following format:

```
dpmresourcemgr setmanagementip [-sc sc_url] -server_name hostname | -system_uuid uuid
-ip_address ipaddress [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmresourcemgr setmanagementstatus Command--Set the Management Status for a Host

The setmanagementstatus command sets a host to a managed or unmanaged state.

This command has the following format:

```
dpmresourcemgr setmanagementstatus [-sc sc_url] -server_name hostname  
-management_status 0|1 [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-host_name *source host name*

Defines the name of the source server in the operation.

-management_status 0 | 1

Defines the management status of the server. Valid entry: 0 = unmanaged state; 1 = managed state.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Set Host to Managed State

This example sets the management status for a host named ServerA to a managed state.

```
dpmcda setmanagementstatus -sc https://localhost/dpm/sc -host_name ServerA  
-management_status 1
```

Example: Set Host to Unmanaged State

This command lets you set the management status of host named ServerB to an unmanaged state.

```
dpmcda setmanagementstatus -host_name ServerB -management_status 0 -ws_user wsuser
-ws_password wsuserpassword
```

dpmresourcemgr setserviceproperties Command--Set Service Properties

The setserviceproperties command lets you create service properties.

This command has the following format:

```
dpmresourcemgr setserviceproperties [-sc sc_url] -servicename servicename
-lower_threshold lowerthreshold -upper_threshold upperthreshold -lag lag -priority
priority [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-service_name *servicename*

Defines the name of the service. Valid entry: Full path to the service name. Use backslash \ to delimit parent service and sub service.

-lower_threshold *lowerthreshold*

(Optional) Specifies the lower overall usage threshold for the service. Lower threshold must be less than upper threshold. Valid entry: 1-100 (%)

-upper_threshold *upperthreshold*

(Optional) Specifies the upper overall usage threshold for the service. Upper threshold must always be greater than lower threshold. Valid entry: 1-100 (%)

-lag *lagvalue*

(Optional) Specifies the lag value. Valid entry: 0 or higher (seconds).

-priority *priorityvalue*

(Optional) Specifies the priority for the service. Valid entry: 1 or higher.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Service Properties

In this example, service properties are set for the service, Service12.

```
dpmresourcemgr setserviceproperties -servicename Service12 -lower_threshold 2  
-upper_threshold 4 -log 2 -priority 2 -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr updateresourcegroup Command--Update Resource Group

The updateresourcegroup command updates a package group.

This command has the following format:

```
dpmresourcemgr updateresourcegroup [-sc sc_url] -group_name groupname  
-group_description description -group_type groupstype -group_attribute attribute  
[-package_list packagelist][-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-group_name *groupname*

Defines the name of the group.

-group_description *description*

Defines the description of the group.

-group_type *type*

Defines the type of resource group. Set to 1 for packaging group. Set to zero for all groups.

-group_attribute *attribute*

Defines the group attribute.

-package_list *packagelist*

Defines individual packages. Define this option if the group type is set to 1(PACKAGE). Valid entries: -package_list "<package name A>|<package procedure A>|<procedure type A>|<package name B>|...".

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Update Resource Group

This example updates a resource group.

```
dpmresourcemgr updateresourcegroup -group_name "AGENT PACKAGE GROUP"
-group_description "Agent package group" -group_type 1 -group_attribute 0
-package_list "Generic Agent Win32 r5.0|install|INSTALL"
```

dpmresourcemgr updateresourcetemplate Command--Update Resource Template

The updateresourcetemplate command updates a package template.

This command has the following format:

```
dpmresourcemgr updateresourcetemplate [-sc sc_url] -template_name templatename
-template_description templatedescription -template_os_type OStype [-group_list
grouplist][-package_list package_list][-ws_user username -ws_password
password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-template_name *templatename*

Defines the name of the template.

-template_description *description*

Defines a description of the template.

-template_os_type *OSType*

Defines the OS type of the resource template for deployment.

-group_list *grouplist*

Defines list of groups for adding to the resource template. Valid entry: `-group_list "<group name A>|<group type 1=Software Package Group>|<group name B>|..."`

-package_list *packagelist*

Defines individual packages. Define this option if the group type is set to 1(PACKAGE). Valid entries: `-package_list "<package name A>|<package procedure A>|<procedure type A>|<package name B>|..."`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Update Package Template

This example updates a package template.

```
dpmresourcemgr updateresourcetemplate -template_name "Generic template"
-template_description "Generic template description" -template_os_type 0 -group_list
"Package group A|1|Package group B|1" -package_list "Generic Agent Win32
r5.0|Install|INSTALL
```

dpmresourcemgr updateservice Command--Update a Service

The updateservice command changes service characteristics.

This command has the following format:

```
dpmresourcemgr updateservice [-sc sc_url] -service_name servicename -machine_list
machinelist -lower_threshold lowerthreshold -upper_threshold upperthreshold -lag lag
-priority priority [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-service_name *servicename*

Defines the name of the service.

-machine_list *machinelist*

(Optional) Specifies the list of available servers. Valid entry: Comma-separated machine names.

-lower_threshold *lowerthreshold*

(Optional) Specifies the lower overall usage threshold for the service. Lower threshold must be less than upper threshold. Valid entry: 1-100 (%)

-upper_threshold *upperthreshold*

(Optional) Specifies the upper overall usage threshold for the service. Upper threshold must always be greater than lower threshold. Valid entry: 1-100 (%)

-lag *lagvalue*

(Optional) Specifies the lag value. Valid entry: 0 or higher (seconds).

-priority *priorityvalue*

(Optional) Specifies the priority for the service. Valid entry: 1 or higher.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Update a Service

This example updates the thresholds, lag, and priority for service, TESTGROUP1.

```
dpmresourcemgr updateservice -service_name TESTGROUP1 -machine_list
"machine4.MyCompany.com, machine5.MyCompany.com, 192.168.255.255" -lower_threshold
10 -upper_threshold 85 -lag 0 -priority 2 -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr updateipnetwork Command--Update a Network

The `dpmresourcemgr updateipnetwork` command updates parameters of an existing network.

This command has the following format:

```
dpmresourcemgr updateipnetwork [-sc sc_url] -subnet_address subnetaddress [-vlan_id
vlanid] [-name name] [-description description] [-ws_user username -ws_password
password] [-locale iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subnet_address *address*

Defines the IP address of the network. Valid entry: xxx.xxx.xxx.xxx or xxx.xxx.xxx.xxx/xx.

-vlan_id *vlanid*

Defines the integer for the virtual network (VLAN). Valid entry: Integer 1-4094, where 1 is the default network interface card (NIC).

-name *name*

Defines the item name. Valid entry: 1-128 alphanumeric characters.

-description *description*

Defines the network for user understanding. Valid entry: 1-128 alphanumeric characters.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Update a Network

This example adds a description to the network, san diego 209 subnet.

```
dpmresourcemgr updateipnetwork -subnet_name 10.10.21.130 -vlan_id 4001 -name "san
diego 209 subnet" -description "5th floor lab" -ws_user wsuser -ws_password
wsuserpassword
```

dpmresourcemgr updatesubnetdefaults Command--Update Network Defaults

The dpmresourcemgr updatesubnetdefaults command updates existing network defaults.

This command has the following format:

```
dpmresourcemgr updatesubnetdefaults [-sc sc_url] -subnet_address subnetaddress
-ip_gateway_list ipaddresses -dns_domainname domainname -dns_list ipaddresses
-dns_suffix_list suffixes -netbios_server_list ipaddresses -ntp_server_list
ipaddresses [-ws_user username -ws_password password][-locale iso639value]
```

-sc_sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-subnet_address address

Defines the IP address of the network. Valid entry: `xxx.xxx.xxx.xxx` or `xxx.xxx.xxx.xxx/xx`.

-ip_gateway_list ipaddresses

Defines the IP addresses for default and alternate gateways using a comma-separated list. Default and alternate gateways must be on the same network. Valid entry: `xxx.xxx.xxx.xxx`.

-dns_domainname domainname

Defines the DNS domain name of the network. Valid entry: 1-63 alphanumeric characters per dotted address (for example, `xxx.yyy.com`). Address cannot start with a number.

-dns_list ipaddresses

Defines the DNS list of preferred and alternate DNS servers, and primary and secondary Windows Internet Name Service (WINS) servers. Valid entry: Comma-separated IP addresses in the format: `xxx.xxx.xxx.xxx`.

-dns_suffix_list suffixes

Defines a list of DNS suffixes for the network. Valid entry: Comma-separated list in the format, `name.name.com`.

-netbios_server_list ipaddresses

Defines a list of NETBIOS Name Server IP addresses. Valid entry: Comma-separated list of IP addresses in the format: `xxx.xxx.xxx.xxx`.

-ntp_server_list ipaddresses

Defines the Network Time Protocol (NTP) servers. Valid entry: Comma-separated list with the format: `xxx.xxx.xxx.xxx`.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Update Network Defaults

This example updates the defaults for the network, san diego 209 subnet.

```
dpmresourcemgr updatesubnetdefaults -subnet_address 10.10.21.130 -ip_gateway_list
10.10.33.0,10.10.34.0 -dns_domainname turbo.com -dns_list 10.10.55.21
-netbios_server_list 10.10.21.130 -ntp_server_list 10.10.33.135 -ws_user wsuser
-ws_password wsuserpassword
```

dpmresourcemgr updateaddresspool Command--Update IP Address Pool

The dpmresourcemgr updateaddresspool command lets you edit the method for IP assignment for an existing pool.

This command has the following format:

```
dpmresourcemgr updateaddresspool [-sc sc_url] -name name|-starting_ip ipaddress
-ip_assignment_type static|dhcp [-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-name *name* | -starting_ip *ipaddress*

Defines the name or starting IP address range. Valid entry: 1-128 alphanumeric characters or xxx.xxx.xxx.xxx.

-ip_assignment_type static | dhcp

Defines the method in which servers get their IP addresses (static IP address or DHCP).

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Update IP Address Pool

This example changes the IP assignment method to DHCP for the IP address pool starting with 10.10.21.130.

```
dpmresourcemgr updateipaddresspool -starting_ip 10.10.21.130 -ip_assignment_type DHCP -ws_user wsuser -ws_password wsuserpassword
```

Service Response Monitor CLI Commands

You can use svcwatch CLI commands to script and automate CA Service Response Monitor and run actions based on the command results.

svcwatch add adir Command--Add an Active Directory Test

The svcwatch add adir command adds an Active Directory test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr adir destination
username password args interval samples timeout winsiz tos
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMixed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

adir

Specifies the Active Directory service type.

destination

Specifies the domain controller for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

domain=*domain* - The domain in which the Active Directory server is located.

query=*query* - The query to send to the Active Directory server.

filter=*filter* - The server-side result filter.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an Active Directory test:

```
svcwatch -p 161 -c admin -o add 1360740 "AD-TEST" adir "DC.com" "ADUser"
"bXVyaWM=" "domain=mylab.com&query=cn=Registered,cn=Users,dc=mylab,
dc=com&filter=cn=" 30 3 30 60 0 0 0x100 "AD-TEST" "ClassName"
"ContextName" 7
```

svcwatch add custom Command--Add a Custom Test

The svcwatch add custom command adds a custom test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr custom destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

custom

Specifies the Custom service type.

destination

Specifies the path of the script.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

No arguments available for the Custom service type. An empty string in quotes "" specifies no arguments.

Note: For details of the particular arguments for each service type, see Keywords for Tests

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a custom test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360741 "CUSTOM-TEST" custom
"c:\scripts\custom-test.bat" "" "" "" 30 3 30 60 0 0 0x100 "Custom-TEST"
"ClassName" "ContextName" 7
```


svcwatch add dhcp Command--Add a DHCP Test

The svcwatch add dhcp command adds a DHCP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr dhcp destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

dhcp

Specifies the DHCP service type.

destination

Specifies the host name or IP address of the DHCP server to test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

No arguments available for the DHCP service type. An empty string in quotes "" specifies no arguments.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a DHCP test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360742 "DHCP-TEST" dhcp
"dhcpservername" "" "" "" "" 30 3 30 60 0 0 0x100 "DHCP-TEST" "ClassName"
"ContextName" 7
```

svcwatch add dns Command--Add a DNS Test

The svcwatch add dns command adds a DNS test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr dns destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

dns

Specifies the DNS service type.

destination

Specifies the DNS server for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service argument. The argument is a pair of a parameter and a value (`hostname=host`) which is enclosed in quotes.

`hostname=host` - The host name to look up.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a DNS test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360743 "DNS-TEST"  
dns "mydnsserver" "" "" "hostname=testhost" 30 3 30 60 0 0 0x100  
"DNS-Test" "ClassName" "ContextName" 7
```

svcwatch add fileio Command--Add a File IO Test

The svcwatch add fileio command adds a file IO test to SRM on the specified host.

This command has the following format:

```
svcwatch [-h] -p | -v | -u | -n | -a | -A | -x | -X | -m | -t | -d | -f] -o add index descr fileio destination username password args interval samples timeout winsiz tos limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

fileio

Specifies the File IO service type.

destination

Specifies the remote file to test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

op=r - Reads the file.

op=w - Writes the contents of a local reference file to a test file located on a remote file system, and then deletes the test file.

op=rw - Writes the contents of a local reference file to a test file on a remote file system, reads the test file, and then deletes the test file.

op=cmp - Reads in one file and then another, and compares their contents.

local=*path* - The local path and file name to use for write, read/write, and compare operations.

domain=*domain* - The domain of the user logging in to the server (Windows only).

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a file I/O test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360744 "FILEIO-TEST"  
fileio "F:\Test\CompTest.bin" "" "" "op=cmp&local=C:\sysedge\bin\saFileIOTest.bin"  
30 1 10 120 0 0 0x100 "FILEIO-TEST" "" ""
```

svcwatch add ftp Command--Add an FTP Test

The svcwatch add ftp command adds an FTP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr ftp destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

ftp

Specifies the FTP service type.

destination

Specifies the FTP server for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

op=g (get) - Log in and read the specified file (but does not perform a write operation), then log out.

op=p (put) - Log in and write the specified file out to the FTP Server, then log out. If the remote directory does not have writer permission, the test fails.

op=login - Log in using the specified username and password and then log out.

remote=*path* - Specifies the path of the file to read.

local=*path* - Specifies the name of the file to write to on the FTP Server.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an ftp test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360745 "FTP-TEST"  
ftp "ftpstage.mydomain.com:21" "ftpuser" "ftp123" "op=login" 3600 1  
10 604800 0 0 0x100 "FTP-TEST" "" "" 1
```

svcwatch add http | https Command--Add an HTTP or HTTPS Test

The `svcwatch add http | https` command adds an HTTP or HTTPS test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr {http|https} destination
username password args interval samples timeout winsiz tos
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

http | https

Specifies the HTTP or HTTPS service type.

destination

Specifies the web server for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

max_depth=*number* - The number of levels the test traverses when downloading nested frames (HTTP and HTTPS tests download all frames, images, external scripts, and applets during page download so that the measurement reflects the user experience when downloading a page).

Default: 3

search=*pattern* - A regular expression you want SRM to match on the pages you test.

minmatch=*number* - Minimum number of times to find the search pattern.

content_dl=true|false - Download content including scripts, images, applets, and so on.

content_err=true|false - Any errors while downloading cause the test to fail.

proxy=*proxy* - The hostname of the proxy server to use if the system from which you are testing does not have direct internet access.

proxyuser=*puser* - The user name for this proxy.

proxypass=*ppass* - The password of the proxy user.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an https test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360746 "HTTPS-TEST"  
https "https://chargeMycredit.com" "creditAcct" "secret"  
"max_depth=3&minmatch=1&content_dl=true&content_err=false" 60 1 20  
300 0 0 0x100 "HTTPS-TEST" "" ""
```

svcwatch add imap Command--Add an IMAP Test

The svcwatch add imap command adds an IMAP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr imap destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

imap

Specifies the IMAP service type.

destination

Specifies the IMAP server and port for the test (*server:port*). Default port: 143

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

download=Download First Message - Downloads only the first message for this user account.

download=Download All Messages - Downloads all messages for this user account.

delete=true - Deletes downloaded messages.

delete=false - Does not delete downloaded messages.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an IMAP test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360747 "IMAP-TEST"  
imap "imapserver.yourdomain.com:143" "IMAPuser@server.domain" "IMAP123"  
"download=Download All Messages&delete=true" 300 1 10 7200 0 0 0x100  
"IMAP-TEST" "" ""
```

svcwatch add ldap Command--Add an LDAP Test

The svcwatch add ldap command adds an LDAP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr ldap destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

ldap

Specifies the LDAP service type.

destination

Specifies the LDAP server and port for the test (*server:port*). Default port: 389

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

query=*query* - Specifies the query to send to the LDAP server.

domain=*domain* - Specifies the LDAP domain.

filter=*filter* - Specifies the server-side result filter.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an LDAP test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360748 "LDAP-TEST"  
ldap "arch" "ldapUser" "ldapPassword" "query=cn=annuity,ou=Boston,  
dc=fleet,dc=com&domain=Test&filter=cn=*" 30 1 10 120 0 0 0x100  
"LDAP-TEST" "" ""
```

svcwatch add mapi Command--Add a MAPI Test

The svcwatch add mapi command adds a MAPI test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr mapi destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

mapi

Specifies the MAPI service type.

destination

Specifies the MAPI server for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

domain=*domain* - The Windows domain for the user account.

op=send - Sends the test message.

op=recv - Receives a test message.

download=Download First Message - Downloads only the first message for this user account.

download=Download All Messages - Downloads all messages for this user account.

to=*recipient* - Specifies the email recipient of the test message.

size=*number of bytes* - Specifies the size of the test message in bytes.
Default 256

delete=true - Deletes downloaded messages.

delete=false - Does not delete downloaded messages.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a MAPI test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360749"MAPI-TEST"  
mapi "mapiserver.yourdomain.com" "MAPIuser" "MAPI123" "domain=myDomain  
&op=send&to=congo@yourdomain.com&size=256" 120 1 10 3600 0 0 0x100  
"MAPI-TEST" "" ""
```

svcwatch add nis Command--Add a NIS Test

The svcwatch add nis command adds an NIS test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr nis destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

nis

Specifies the NIS service type.

destination

Specifies the NIS server for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

domain=domain - Specifies the domain on which the map file exists.

map=map - Specifies the map file to test. Default: hosts

download=true|false - Specifies whether to download the NIS map.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an NIS test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360750 "NIS-TEST"  
nis "nisserver.yourdomain" "" "" "domain=Test.com&map=hosts&download=true"  
300 1 20 7200 0 0 0x100 "NIS-TEST" "" ""
```

svcwatch add nntp Command--Add an NNTP Test

The svcwatch add nntp command adds an NNTP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr nntp destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

nntp

Specifies the NNTP service type.

destination

Specifies the NNTP server and port for the test (*server:port*). Default port: 119

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

No arguments available for the NNTP service type. An empty string in quotes "" specifies no arguments.

Note: For details of the particular arguments for each service type, see Keywords for Test Arguments.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an NNTP test:

```
svcwatch -h localhost -p 161 -c topsecret -o add 1360751 "NNTP-TEST" nntp
"nntpservername:119" "" "" "" 30 3 30 60 0 0 0x100 "TestNNTP" "ClassName"
"ContextName" 7
```

svcwatch add ping Command--Add a PING Test

The svcwatch add ping command adds a PING test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr ping destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

ping

Specifies the PING service type.

destination

Specifies the target computer for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service argument. The argument is a pair of a parameter and a value (payload=*number of bytes*) which is enclosed in quotes.

payload=*number of bytes* - The size of the packet sent in the ping. The default packet size is 64 bytes.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a ping test:

```
svcwatch -h localhost -p 161 -c admin -o add 1360752 "TEST" ping "127.0.0.1"  
"" "" "payload=1000" 30 3 30 60 0 0 0x100 "TestPING" "ClassName" "ContextName" 7
```

svcwatch add pop3 Command--Add a POP3 Test

The svcwatch add pop3 command adds a POP3 test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr pop3 destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

pop3

Specifies the POP3 service type.

destination

Specifies the POP3 server and port (*server:port*) for the test. Default port: 110

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

download=Download First Message - Downloads only the first message for this user account.

download=Download All Messages - Downloads all messages for this user account.

delete=true - Deletes downloaded messages.

delete=false - Does not delete downloaded messages.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a POP3 test:

```
svcwatch -h localhost -p 161 -c doublesecret -o add 1360753 "POP3-TEST"  
pop3 "mypop3server:110" "pop3user" "pop3pass" "download=Download First Message  
&delete=true" 30 3 30 60 0 0 0x100 "TestPOP3" "ClassName" "ContextName" 7
```

svcwatch add rtemail Command--Add a Round Trip Email Test

The `svcwatch add rtemail` command adds a round-trip email test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr rtemail destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The `add` command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

rtemail

Specifies the round-trip email service type.

destination

Specifies the SMTP or MAPI server for the test (*smtpserver:port*). Default SMTP port: 25

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

send_proto=smtp|mapi - Specifies the sending protocol.

to=*recipient* - Specifies the name of the email recipient.

size=*number of bytes* - Specifies the size of the test message in bytes.
Default 256

shost=*senderhost* - Specifies the host name which sends the message.

smtp_ssl=true|false - Specifies whether SMTP SSL encryption is enabled.

send_user=*user* - Specifies the name of the sender account.

send_pass=*spass* - Specifies the password of the sender.

sdomain=*domain* - (MAPI only) Specifies the domain the sender belongs to.

get_proto=pop3|imap|mapi - Specifies the receiving protocol.

source=*targethost* - Specifies the host to which the email is sent.

source_port=*port* - Port number for IMAP or POP3. Defaults: 110 (POP3), 143 (IMAP)

gdomain=*domain* - (MAPI only) Specifies the domain the receiver belongs to.

check=*poll interval* - Specifies the poll interval in seconds. Default: 600

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a round-trip email test:

```
svcwatch -p 161 -c admin -o add 1360754 "RT-EMAIL TEST" rtemail  
"mysmtp.com:25" "smtpuser" "zrvr8tx" "send_proto=smtp&to=recipient  
&size=256&shost=senderhost&smtp_ssl=true&send_user=sender&send_pass=secret  
&get_proto=pop3&source=popserver&source_port=110&check=600"  
30 3 30 60 0 0 0x100 "TestRTEMAIL" "ClassName" "ContextName" 7
```

svcwatch add smtp Command--Add an SMTP Test

The svcwatch add smtp command adds an SMTP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr smtp destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

smtp

Specifies the SMTP service type.

destination

Specifies the SMTP server and port (*server:port*) for the test. Default port: 25

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

to=*recipient* - Specifies the name of the email recipient.

size=*number of bytes* - Specifies the size of the test message in bytes.
Default 256

shost=*senderhost* - Specifies the host name which sends the message.

ssl=true|false - Specifies whether SMTP SSL encryption is enabled.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding an SMTP test:

```
svcwatch -p 161 -c admin -o add 1360755 "SMTP TEST" smtp "mysmtp.com:25"  
"smtpuser" "zrvr9tx" "to=recipient&size=256&shost=senderhost&smtp_ssl=true"  
30 3 30 60 0 0 0x100 "TestSMTP" "ClassName" "ContextName" 7
```

svcwatch add snmp Command--Add an SNMP Test

The svcwatch add snmp command adds an SNMP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr snmp destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

snmp

Specifies the SNMP service type.

destination

Specifies the SNMP agent host and port (*server:port*) for the test. Default SNMP port: 161

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

snmpversion=snmp|snmp3 - Specifies SNMPv1/2 or SNMPv3.

user=*user name* - Specifies the SNMPv3 user name.

securitylevel=AuthPriv|NoAuthNoPriv|AuthNoPriv - Specifies the SNMPv3 security level

auth_protocol=MD5|SHA - Specifies the SNMPv3 authentication protocol.

auth_password=*password* - Specifies the authentication password.

priv_protocol=DES|AES|3DES - Specifies the SNMPv3 privacy protocol.

priv_password=*password* - Specifies the privacy password.

oid=*oid* - The object identifier of the variable to query.

community=*string* - The community string for reading the variable.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Examples

Adding an SNMPv1 test:

```
svcwatch -p 161 -c admin -o add 1360757 "SNMP TEST" snmp "myhost:161" "" ""  
"snmpversion=snmp&oid=1.3.6.1.4.1.1.546.1.1.7.8.27.0&community=admin" 30 3 30  
60 0 0 0x100 "TestSNMP" "" "" 7
```

Adding an SNMPv3 test:

```
svcwatch -p 161 -c admin -o add 1360756 "SNMP3 TEST" snmp "myhost:161" "" ""  
"snmpversion=snmp3&user=admin&securitylevel=AuthPriv&auth_protocol=MD5  
&auth_password=XP1abcTZ&oid=1.3.6.1.4.1.1.546.1.1.7.8.27.0&community=topsecret"  
30 3 30 60 0 0 0x100 "TestSNMP3" "" "" 7
```

svcwatch add sql Command--Add an SQL Test

The svcwatch add sql command adds a SQL test to SRM on the specified host. For details and prerequisites, see SQL Query Tests in this guide.

This command has the following format:

```
svcwatch [options] -o add index descr sql destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMixed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

sql

Specifies the SQL service type.

destination

Specifies the database server and port (*server:port*) for the test. Default ports: 1433 (SQL Server), 1521 (Oracle)

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (*key=value*). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "*key1=value1[&key2=value2& ...]*"

query=query - Specifies the SQL query.

qtype=query|stored|update - (Optional) Specifies the query type (Default: *query*)

dbtype=oracle|mssql|other - Specifies the database type.

dbname=name - (*oracle* and *mssql* only) Specifies the database name. When you select "*other*", specify the database name in the connect string.

driver=driver - (*other* only) The JDBC driver to use for the query.

url=string - (*other* only) The database connect string; varies by database type. For information about connect strings, refer to your database documentation.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Examples

Adding a SQL server test:

```
svcwatch -p 161 -c admin -o add 1360758 "TestSqlServer" sql
"192.168.100.100:1433" "sa" "AdminLvl1" "query=select * from
vas_system&dbtype=mssql&dbname=vasdb" 30 3 30 60 0 0 0x100
"TestSQL" "" ""
```

Adding a SQL server JDBC driver test:

```
svcwatch -p 161 -c admin -o add 1360761 "TestSqlServerJDBCdriver"
sql "192.168.100.100:1433" "sa" "AdminLvl1" "query=select * from
my_system&dbtype=other&driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
&url=jdbc:sqlserver://hostnameValue:portValue;databaseName=vasdb;
user=unameValue;password=pwdValue" 30 3 30 60 0 0 0x100
"TestSqlServerJDBCdriver" "" ""
```

Adding an Oracle test:

```
svcwatch -p 161 -c admin -o add 1360759 "TestOracle" sql
"192.168.101.101:1521" "joe" "AdminLvl1" "query=select * from big_system
&dbtype=oracle&dbname=bigdb" 30 3 30 60 0 0 0x100 "TestOracle" "" ""
```

Adding an Oracle JDBC Thin Driver test:

```
svcwatch -p 161 -c admin -o add 1360762 "TestOracleJDBCThinDriver" sql
"192.168.101.101:1521" "joe" "AdminLvl1" "query=select table_name from
user_tables&dbtype=other&driver=oracle.jdbc.OracleDriver
&url=jdbc:oracle:thin:unameValue/pwdValue@hostnameValue:portValue:orcl"
30 3 30 60 0 0 0x100 "TestOracleThin" "" ""
```

svcwatch add tcpconnect Command--Add a TCP Connect Test

The svcwatch add tcpconnect command adds a TCP Connect test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr tcpconnect destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add *testparams*

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

tcpconnect

Specifies the TCP Connect service type.

destination

Specifies the remote host and port (*server:port*) for the test.

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

No arguments available for the DHCP service type. An empty string in quotes "" specifies no arguments.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a TCP connect test:

```
svcwatch -p 161 -c admin -o add 1360763 "Test" tcpconnect  
"192.168.90.10:1433" "" "" "" 60 5 30 60 0 0 0x100  
"Testtcpconnect" "" "" 1
```

Here, as host is not specified, default is localhost. The port number is 161. 'add' adds new test to SRM. The unique svcRspTable index number is 2013. "Test" is the description of this test. tcpconnect specifies TCP Connect service type. The details of remote host and port tested is 192.168.90.10:1433.

The two empty strings in quotes ("") specify no user name and password used for authentication. "" specifies no arguments available for tcpconnect service type. The test interval is specified as 60 seconds. The number of samples tested per interval is 5. The timeout is 30 seconds. The statistics window size is 60 seconds. 0 (zero) specifies a normal service (and not IP Type of Service or Differentiated Services)

0 (zero) specifies the acceptable performance limit (or threshold) for the total response time of this test. This test is executed on request only (run once) with flag set to 0x100. Specifies the unique name per service type as Testtcpconnect. Specifies no class name, and context information in "". This test is executed with log level as critical with '1'.

svcwatch add tftp Command--Add a TFTP Test

The `svcwatch add tftp` command adds a TFTP test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr tftp destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

tftp

Specifies the TFTP service type.

destination

Specifies the TFTP server and port (*server:port*) for the test. Default port: 69

username

Specifies the user name for authentication in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

op=r - Reads a file from the server.

op=w - Writes a file to a remote file system.

path=*path* - The path and file name of the file to read or write.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a TFTP test:

```
svcwatch -p 161 -c admin -o add 1360764 "Test" tftp "192.168.120.10:69"  
"" "" "op=r&path=I:\SA\TFTP\get.txt" 60 3 30 60 0 0 0X100 "Testtftp"  
"" "" 0
```

svcwatch add vuser Command--Add a Virtual User Test

The svcwatch add vuser command adds a Virtual User test to SRM on the specified host.

This command has the following format:

```
svcwatch [options] -o add index descr vuser destination  
username password args interval samples timeout winsiz tos  
limit flags name class contextInfo logLevel
```

The add command uses the following parameters:

options

Specifies the possible options for this command.

-h hostname | -h ipAddr

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p port

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c community

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v snmpVersion

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t timeout

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d logLevel

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f logFile

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o add testparams

Adds a new test to SRM.

testparams

Specifies the parameters for the new test.

index

Specifies the svcRspTable index.

descr

Specifies the description of the test in quotes. An empty string in quotes "" specifies no description.

vuser

Specifies the Virtual User service type.

destination

Specifies the path of the script for the test.

username

Specifies the user name to run the script in quotes. An empty string in quotes "" specifies no user name.

password

Specifies the password for authentication in quotes. An empty string in quotes "" specifies no password.

args

Specifies the service arguments. Each argument is a pair of a keyword and a value (key=value). Multiple arguments are concatenated and delimited by ampersands (&). The complete arguments string is enclosed in quotes.

General syntax for arguments: "key1=value1[&key2=value2& ...]"

host=*hostname:port* - The hostname of the destination system; this value is used for DNS lookup and connection timings. The port on which the script is running.

domain=*domain* - The domain of the user who is running the script.

Note: For details of the particular arguments for each service type, see Keywords for Tests.

interval

Specifies the test interval in seconds.

Limits: multiple of 30 seconds

samples

Specifies the samples per interval.

timeout

Specifies the timeout in seconds.

winsiz

Specifies the statistics window size in seconds.

tos

Specifies the IP Type of Service or Differentiated Services Code. Use 0 (zero) for a normal service. See also RFC 1349.

limit

Specifies the acceptable performance limit (or threshold) for the total response time of this test. This value is used in reports.

flags

Specifies the following flags:

0x001 = collect performance cubes

0x100 = execute on request only (run once)

name

Specifies the unique name per service type. An empty string in quotes "" specifies no test name.

class

Specifies the class name. An empty string in quotes "" specifies no class name.

contextInfo

Specifies context information. An empty string in quotes "" specifies no context information.

logLevel

Specifies the log level for the test execution code. Possible values are:

-2 = use SRM-global log level (default)

-1 = do not log

0 = fatal (only the most important messages)

1 = critical

...

7 = debug3 (log all messages)

Example

Adding a virtual user test:

```
svcwatch -h localhost -p 161 -c snmp_admin -o add 1360765 "vUser Test"
vuser "C:\WinTask\bin\taskexec.exe C:\WinTask\Scripts\notepad.rob" ""
"" "" 30 1 10 120 0 0 0x001 "vUserTEST" "ClassName" "ContextName" 7
```

svcwatch delete Command--Delete a Test

The svcwatch delete command deletes a test on the specified host.

This command has the following format:

```
svcwatch [options] -o delete index
```

The delete command uses the following parameters:

options

Specifies the possible options for this command.

-h *hostname* | -h *ipAddr*

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p *port*

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c *community*

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v *snmpVersion*

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMixed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o delete *index*

Deletes the specified test.

index

Specifies the svcRspTable index used to identify the test to delete.

Example

Delete a test on myremote host:

```
svcwatch -h myremote -p 161 -c admin -o delete 1360739
```

svcwatch list Command--View Test Information

The svcwatch list command lists the available tests on the specified host.

This command has the following format:

```
svcwatch [options] -o list
```

The list command uses the following parameters:

options

Specifies the possible options for this command.

-h *hostname* | -h *ipAddr*

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p *port*

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c *community*

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v *snmpVersion*

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMixed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o *list*

Lists available tests.

Example

Listing tests on localhost:

```
svcwatch -o list
```

svcwatch setstatus Command--Change the Status of a Test

The svcwatch setstatus command changes the status of a test on the specified host.

This command has the following format:

```
svcwatch [options] -o setstatus index status
```

The setstatus command uses the following parameters:

options

Specifies the possible options for this command.

-h *hostname* | -h *ipAddr*

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p *port*

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c *community*

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v *snmpVersion*

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMixed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o *setstatus index status*

Specifies the new status of a test.

index

Specifies the svcRspTable index used as an identifier of the test.

Note: You overwrite an existing test with a new test, when you use the add command with the index of that test.

status

Specifies the status. Possible values are:

active (1),
notInService (2),
notReady (3),
createAndGo (4),
createAndWait (5),
destroy (6)

Example

Change the status of a test:

```
svcwatch -p 161 -c admin -o setstatus 1360739 2
```

svcwatch version Command--View SRM Version Information

The svcwatch version command displays the version of the SRM AIM on the specified host.

This command has the following format:

```
svcwatch [options] -o version
```

The version command uses the following parameters:

options

Specifies the possible options for this command.

-h *hostname* | -h *ipAddr*

(Optional) Specifies the SystemEDGE host.

Default: localhost

-p *port*

(Optional) Specifies the SystemEDGE SNMP port.

Default: 161

-c *community*

(Optional) Specifies the SNMP community string for SNMP version 1 and 2c.

Default: public

-v *snmpVersion*

(Optional) Specifies the SNMP version. Possible values are:

- 1
- 2c
- 3

Default: 1

-u *secName*

(Optional) Specifies the name of the SNMPv3 secure user.

-u *secLevel*

(Optional) Specifies the level of security for SNMPv3. Possible values are:

- 1 – noAuthNoPriv
- 2 – AuthNoPriv
- 3 – AuthPriv

-n *contextName*

(Optional) Specifies the instance name for a MIBMuxed agent.

-a *authPassword*

(Optional) Identifies the authentication password required when SNMPv3 is selected with security AuthNoPriv or AuthPriv.

-A *authProtocol*

(Optional) Specifies the authentication protocol. Possible values are:

- MD5 – authentication protocol HMAC-MD5
- SHA – authentication protocol HMAC-SHA

Default: MD5

-x *privPassword*

(Optional) Specifies the privacy (encryption) password for SNMPv3 with security level 3 (AuthPriv).

-X *encryptProtocol*

(Optional) Specifies the use of encryption protocol for privacy. Possible values are:

- DES – Data Encryption Standard
- AES – Advanced Encryption Standard using cryptographic keys of 128 bits (AES128)
- 3DES – Triple Data Encryption Standard

-m *FIPSmode*

(Optional) Specifies the FIPS mode. Possible values are:

- 0 – non-FIPS
- 1 – FIPS coexistence
- 2 – FIPS only

Default: 0

-t *timeout*

(Optional) Specifies the SNMP command timeout.

Default: 10 seconds

-d *logLevel*

(Optional) Specifies the log level for SNMP messages. Possible values are:

- 0 – log fatal messages
- 1 – log critical messages
- 2 – log warning messages
- 3 – log information messages
- 4 – log all messages
- 5 – log all messages including debugging messages

Default: 0

-f *logFile*

(Optional) Specifies the name of the logfile.

Default: sysedge_utility.log

-L

(Optional) Detects the current locale of the console and language catalog if available. If a language catalog is not found, the utility falls back to English as a default language.

-o *version*

Displays the version of the CA SystemEDGE AIM for Service Response Monitoring.

Example

Display the version on myremote host:

```
svcwatch -h myremote -o version
```

Software Delivery Commands

You can use dpmsd CLIs to script and automate Software Delivery commands and run actions based on the command results.

dpmsd cancelimage Command--Cancel a Pending Image

The `dpmsd cancelimage` command cancels an image that has been deployed to a server. You can cancel the OS image jobs that are pending or already in progress. When you cancel an operating system image that is already in progress, the state of the target host becomes unpredictable. You may need to reset the target host for the next task.

Note: You can modify the `casdaconf.cfg` file to disable canceling an operating system image deployment in progress. The flag is set to enable canceling an image in progress: `SD_Img_Cancel_Job_In_Progress=Yes`. To disable this setting, change “Yes” to “No” and recycle Apache service.

This command has the following format:

```
dpmsd cancelimage [-sc sc_url] -target_host targethostname [-itcm_server  
itcmdomainmanager] [-pre][-post[-ws_user username -ws_password password][-locale  
iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-target_host *targethostname*

Defines the name of the target host server to which you are deploying the image.

-itcm_server *itcmdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Cancel an Image on a Host

This example cancels an operating system image on the target host server001.

```
dpmsd cancelimage -target_host server001 -pre -post
```

Example: Cancel an Image on a Host in a Multiple ITCM Domain Manager Environment

This example cancels an operating system image on the target host server001. This operation is routed to the sd adapter on ITCM domain manager domainmanager001.

```
dpmsd cancelimage -target_host server001 - itcm_server domainmanager001 -pre -post
```

dpmsd deliver Command--Deliver a Package

The dpmsd deliver command delivers a package to a host computer.

This command has the following format:

```
dpmsd deliver [-sc sc_url] -package_name packagename -procedure_name procedurename  
-computer_host computerhostname [-scalability_server scalabilityservername]  
[-itcm_server itcmdomainmanager] -computer_username username [-computer_password  
password] [-auth_file authorizationfilename] [-auth_comp componentID] -os_type  
{Windows|Linux|Any|HP-UX|AIX|Solaris} [-wait [timeout]] [-pre] [-post][-ws_user  
username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-package_name {*packagename* | *all*}

Defines the name of the package to deliver.

packagename

Applies the managed status change to the specified package.

all

Apply the managed status change to all packages.

-procedure_name *procedurename*

Defines a name for the procedure.

-computer_host *computerhostname*

Defines the name of the server for deploying the package.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-itcm_server *itcdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-computer_username *computerusername*

Defines the user name of the target host server for the deployment operation.

Windows: Administrator access is required.

UNIX/Linux: Root access is required.

-computer_password *computerpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying. If you do not specify the password, is retrieved from the authorization file.

Note: Use the `dpmutil` CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the `dpmutil set auth` command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-os_type={Windows|Linux|HP-UX|AIX|Solaris|Any}

Defines the type of operating system that is on the computer host. This parameter is required for software delivery agent deployment. Options include the following:

Windows

Specifies the Windows operating system.

Linux

Specifies the Linux operating system.

HP-UX

Specifies the HP UNIX operating system.

AIX

Specifies the AIX UNIX operating system.

Solaris

Specifies the Solaris operating system.

Any

Specifies any type of operating system.

Note: This option is valid for image deployment only.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deliver a Software Package to a Host

This example delivers a package containing a version of Apache software to a Windows host computer.

```
dpmsd deliver -computer_host comphost001 -package_name "Apache HTTP Server 2.2.3 2.2.3" -procedure_name Install -os_type Windows -computer_username administrator
```

Example: Deliver a Software Package to a Host in a Multiple ITCM Domain Manager Environment

This example delivers a package containing a version of Apache software to a Windows host computer. This operation is routed to the sd adapter on ITCM domain manager domainmanager001.

```
dpmsd deliver -computer_host comphost001 -package_name "Apache HTTP Server 2.2.3 2.2.3" -procedure_name Install -os_type Windows -itcm_server domainmanager001 -computer_username administrator
```

dpmsd deploypkggrp Command--Deliver a Package Group

The dpmsd deploypkggrp command delivers a package group to a host computer. A package group contains one or more software packages.

This command has the following format:

```
dpmsd deploypkggrp [-sc sc_url] -package_group_name packagegroupname -computer_host computerhostname [-scalability_server servername] [-itcm_server itcmdomainmanager] -computer_username username [-computer_password password] [-auth_file authorizationfile] [-auth_comp componentID] -os_type {Windows|Linux|HP-UX|AIX|Solaris} [-wait [timeout]] [-pre][-post][-ws_user username -ws_password password][-locale iso629value]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-package_group_name packagegroupname

Defines a name for a predefined software delivery package group.

-computer_host computerhostname

Defines the name of the server for deploying the package.

-scalability_server servername

(Optional) Specifies the CA ITCM scalability server for software distribution.

-itcm_server itcmdomainmanager

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-computer_username computerusername

Defines the user name of the target host server for the deployment operation.

Windows: Administrator access is required.

UNIX/Linux: Root access is required.

-computer_password *computerpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying. If you do not specify the password, is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-os_type={*Windows|Linux|HP-UX|AIX|Solaris|Any*}

Defines the type of operating system that is on the computer host. This parameter is required for software delivery agent deployment. Options include the following:

Windows

Specifies the Windows operating system.

Linux

Specifies the Linux operating system.

HP-UX

Specifies the HP UNIX operating system.

AIX

Specifies the AIX UNIX operating system.

Solaris

Specifies the Solaris operating system.

Any

Specifies any type of operating system.

Note: This option is valid for image deployment only.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deliver a Package Group to a Host

This example delivers a package group to a Windows host computer.

```
dpmsd deploypkggrp -package_group_name Installations -computer_host comphost001  
-computer_username administrator -os_type Windows
```

Example: Deliver a Package Group to a Host in a Multiple ITCM Domain Manager Environment

This example delivers a package group to a Windows host computer. This operation is routed to the sd adapter on ITCM domain manager domainmanager001.

```
dpmsd deploypkggrp -package_group_name Installations -computer_host comphost001  
-itcm_server domainmanager001 -computer_username administrator -os_type Windows
```


dpmsd deployprocgrp Command--Deliver a Procedure Group

The dpmsd deployprocgrp command delivers a procedure group to a host computer.

This command has the following format:

```
dpmsd deployprocgrp [-sc sc_url] -procedure_group_name proceduregroupname
-computer_host computerhostname [-scalability_server servername] [-itcm_server
itcmdomainmanager] -computer_username username [-computer_password password]
[-auth_file authorizationfile] [-auth_comp componentID] -os_type
{Windows|Linux|HP-UX|AIX|Solaris} [-wait [timeout]] [-pre][-post][-ws_user username
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-procedure_group_name *proceduregroupname*

Defines a name for the software delivery procedure group.

-computer_host *computerhostname*

Defines the name of the server for deploying the package.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-itcm_server *itcmdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-computer_username *computerusername*

Defines the user name of the target host server for the deployment operation.

Windows: Administrator access is required.

UNIX/Linux: Root access is required.

-computer_password *computerpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying. If you do not specify the password, is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-os_type={Windows|Linux|HP-UX|AIX|Solaris|Any}

Defines the type of operating system that is on the computer host. This parameter is required for software delivery agent deployment. Options include the following:

Windows

Specifies the Windows operating system.

Linux

Specifies the Linux operating system.

HP-UX

Specifies the HP UNIX operating system.

AIX

Specifies the AIX UNIX operating system.

Solaris

Specifies the Solaris operating system.

Any

Specifies any type of operating system.

Note: This option is valid for image deployment only.

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the `caimgconf.cfg` file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: Deliver a Procedure Group to a Host

This example delivers a procedure group to a Windows host computer.

```
dpmsd deployprocgrp -procedure_group_name Installations -computer_host comphost001
-computer_username administrator -os_type Windows
```

Example: Deliver a Procedure Group to a Host in a Multiple ITCM Domain Manager Environment

This example delivers a procedure group to a Windows host computer. This operation is routed to the `sd` adapter on ITCM domain manager `domainmanager001`.

```
dpmsd deployprocgrp -procedure_group_name Installations -itcm_server
domainmanager001 -computer_host comphost001 -computer_username administrator
-os_type Windows
```

dpmsd image Command--Deploy an Image

The dpmsd image command deploys an image to a server.

This command has the following format:

```
dpmsd image [-sc sc_url] -img_name imagename -target_host targethostname -target_mac  
macaddress -auto_deploy {yes|no} [-itcm_server itcmdomainmanager]  
-computer_username username [-computer_password password] [-auth_file  
authorizationfilename] [-auth_comp componentID] [-deploy_template templatename]  
-boot_server bootserver [-interface interface] [-net_protocol netprotocol]  
[-net_ip_address netipaddress] [-net_def_gateway netdefgateway] [-net_sub_mask  
netsubmask] [-dns_server dnsserver] [-dns_suffix dnssuffix] [-boot_sepical  
bootsepical] [-install_drive installdrive] [-wait [timeout]][-pre][-post][-ws_user  
username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-img_name *imagename*

Defines the name of the image to deploy.

-target_host *targethostname*

Defines the name of the target host server to which you are deploying the image.

-target_mac *macaddress*

Defines the hardware address of the computer to which you are deploying the image.

-auto_deploy {yes | no}

Specifies whether CA Server Automation agents are deployed automatically. Options include the following:

yes

Deploys CA Server Automation agents automatically.

no

Prevents CA Server Automation agents from being deployed automatically.

Default: no

-itcm_server *itcmdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-computer_username *computerusername*

Defines the user name of the target host server for the deployment operation.

Windows: Administrator access is required.

UNIX/Linux: Root access is required.

-computer_password *computerpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying. If you do not specify the password, is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-deploy_template *templatename*

(Optional) Specifies the name of the generic template created in CA Server Automation.

Note: Do not confuse this template with the templates created and managed by VMware vCenter.

-boot_server *bootserver*

Defines the CA ITCM boot server to deliver an image to a server.

-interface *interface*

(Optional; for ESX only) Defines the uplink device for the virtual switch created for the service console for VMware ESX 4.0 (vmnic0, vmnic1, and so on).

Defines an Ethernet device for the installation for VMware ESX 3.5 (eth0, eth1, and so on).

-net_protocol *netprotocol*

(Optional; for ESX only) Defines the use of static IP addresses for VMware ESX installations. Values can be either DHCP or static.

Default: DHCP

-net_ip_address *netipaddress*

(Optional; for ESX only) Defines the IP address of the server to install.

-net_def_gateway *netdefgateway*

(Optional; for ESX only) Defines the default gateway as an IP address.

-net_sub_mask *netsubmask*

(Optional; for ESX only) Defines the subnet mask for the installed system.

Default: 255.255.255.0

-dns_server *dnserver*

(Optional; for ESX only) Defines the primary name server as an IP address. You can add the IP address of a secondary name with a preceding comma.

-dns_suffix *dnssuffix*

(Optional; for ESX only) Defines a search list for host name lookup. Use spaces to separate multiple suffixes.

-boot_special *bootspecial*

(Optional) Defines the Windows PE boot image for the VMware ESX installation. DOSX boot images are not supported.

-install_drive *installdrive*

(Optional) Defines the drive for installing the OS. hda1=IDE sda1=SCSI, RAID. You can specify multiple drives.

Default: sda1

-wait *timeout*

(Optional) Indicates whether to display the status of the job and not return until the operation completes, fails, or the timeout period is met (if you specify a timeout value). If you do not specify this option, the CLI returns without waiting for completion. If there is no response, the optional timeout value defines the timeout period in minutes. If you enter the wait option with no timeout value, the CLI uses the default wait time from the caimgconf.cfg file or defaults to 120 minutes. In addition to any positive integer, the following timeout values are also possible:

""

No timeout value.

0

Wait indefinitely.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Deploy an OS Image on a Host without Deploying Agents

This example installs an operating system image on the target host server001 using boot server bootserver001 and image ghost001 without deploying the agents.

```
dpmsd image -target_host server001 -target_mac 00:00:00:00:00:00 -auto_deploy no  
-boot_server bootserver001 -img_name ghost001 -computer_username administrator  
-computer_password adminpassword
```

Example: Deploy an OS Image on a Host without Deploying Agents in a Multiple ITCM Domain Manager Environment

This example installs an operating system image on the target host server001 using boot server bootserver001 and image ghost001 without deploying the agents. This operation is routed to the sd adapter on ITCM domain manager domainmanager001.

```
dpmsd image -target_host server001 -target_mac 00:00:00:00:00:00 -auto_deploy no
-itcm_server domainmanager001 -boot_server bootserver001 -img_name ghost001
-computer_username administrator -computer_password adminpassword
```

dpmsd imgjobcheck Command--Get OS Imaging Job Status

The dpmsd imgjobcheck command retrieves the image job status of the operating system deployment for a specific CA Server Automation job ID.

This command has the following format:

```
dpmsd imgjobcheck [-sc sc_url] -status jobID [-pre][-post][-ws_user username
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-status={*job ID*}

Defines the job ID used to obtain the job status.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Get the Operating System Image Job Status Using the Job ID

This example retrieves the image job status of the operating system deployment using a CA Server Automation job ID.

```
dpmsd imgjobcheck -status 40
```

dpmsd installSDAgent--Install Software Delivery Agent

The dpmsd installSDAgent command installs the Software Delivery agent on a host computer.

This command has the following format:

```
installSDAgent [-sc sc_url] -computer_host computerhostname [-scalability_server scalability server] [-itcm_server itcm domain manager] -computer_username computerusername [-computer_password computerpassword][-auth_file authorizationfile][-auth_comp componentID] -os_type ostype [-pre][-post][[-ws_user username -ws_password password]][-prompt <yes|no>][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-computer_host *computerhostname*

Defines the name of the server for deploying the package.

-scalability_server *servername*

(Optional) Specifies the CA ITCM scalability server for software distribution.

-itcm_server *itcmdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-computer_username *computerusername*

Defines the user name of the target host server for the deployment operation.

Windows: Administrator access is required.

UNIX/Linux: Root access is required.

-computer_password *computerpassword*

(Optional) Defines the user password used for deploying agents to the target host server to which you are deploying. If you do not specify the password, is retrieved from the authorization file.

Note: Use the dpmutil CLI to set up the authorization file.

-auth_file *authorizationfilename*

(Optional) Specifies the full path name of the authorization file to use to retrieve the encrypted password. This file contains the CA Server Automation administrator credentials that were generated using the dpmutil set auth command. When this option is not specified, the default authorization file is used.

-auth_comp *componentID*

(Optional) Specifies a component ID that you can use to group hosts and users.

-os_type={Windows|Linux|HP-UX|AIX|Solaris|Any}

Defines the type of operating system that is on the computer host. This parameter is required for software delivery agent deployment. Options include the following:

Windows

Specifies the Windows operating system.

Linux

Specifies the Linux operating system.

HP-UX

Specifies the HP UNIX operating system.

AIX

Specifies the AIX UNIX operating system.

Solaris

Specifies the Solaris operating system.

Any

Specifies any type of operating system.

Note: This option is valid for image deployment only.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Install SD Agent on Linux System

This example installs the sd agent on a Linux system.

```
dpmsd installSDAgent -computer_host comphost001 -computer_username administrator  
-computer_password adminpassword -os_type LINUX -pre -post -ws_user wsuser  
-ws_password wsuserpassword
```

Example: Install SD Agent on Linux System in a Multiple ITCM Domain Manager Environment

This example installs the sd agent on a Linux system. This operation is routed to the sd adapter on ITCM domain manager domainmanager001.

```
dpmsd installSDAgent -computer_host comphost001 -itcm_server domainmanager001
-computer_username administrator -computer_password adminpassword -os_type LINUX
-pre -post -ws_user wsuser -ws_password wsuserpassword
```

dpmsd list Command--List Software Packages

The dpmsd list command lists all managed or unmanaged software delivery packages.

This command has the following format:

```
dpmsd list [-sc sc_url] -packages {all|unmanaged|managed} [-itcm_server
itcmdomainmanager ] [-pre][-post][-ws_user username -ws_password password][-locale
iso629value]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-packages={all|unmanaged|managed}

Specifies whether you want a list of all packages, unmanaged, or managed packages.

-itcm_server itcmdomainmanager

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Example: List Software Packages

This example lists all available software packages.

```
dpmsd list -packages all
```

Example: List Software Packages in a Multiple ITCM Domain Manager Environment

This example lists all available software packages. This operation is routed to the `sd` adapter on ITCM domain manager `domainmanager001`.

```
dpmsd list -packages all -itcm_server domainmanager001
```

dpmsd listsd Command--List Installed SD Adapters

The `dpmsd listsd` command lists the systems where the SD Adapter is installed. In this case it is listing the ITCM Domain Managers.

This command has the following format:

```
dpmsd listsd [-sc sc_url] [-pre][-post] [-ws_user username -ws_password  
password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: List Installed SD Adapter

This example lists the systems where the SD Adapter is installed.

```
dpmsd listsd
```

dpmsd pkgjobcheck Command--Get Package Deployment Job Status

The `dpmsd pkgjobcheck` command obtains the operating system deployment job status for a specific CA Server Automation job ID.

This command has the following format:

```
dpmsd pkgjobcheck [-sc sc_url] -status jobID [-itcm_server itcmdomainmanager]  
[-pre][-post][-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(Optional) Defines the URL for the service controller host.

Example: `https://hostname:port/...`

hostname

Defines the fully qualified host where the service controller is installed.

Example: `localhost.ca.com`

port

Defines the listening port for the service controller server.

Example: `80`

-status=*{job ID}*

Defines the job ID used to obtain the job status.

-itcm_server *itcmdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using `caaipsecurity`.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify `"native"`.

Example: Get the Package Deployment Job Status Using the Job ID

This example gets the software package deployment status for a specific CA Server Automation job ID.

```
dpmsd pkgjobcheck -status 47
```

Example: Get the Package Deployment Job Status Using the Job ID in a Multiple ITCM Domain Manager Environment

This example gets the software package deployment status for a specific CA Server Automation job ID. This operation is routed to the sd adapter on ITCM domain manager domainmanager001.

```
dpmsd pkgjobcheck -status 47 -- itcm_server domainmanager001
```

dpmsd setmanagedstatus Command--Set Management Status of Package

The dpmsd setmanagedstatus command sets the status of a specific package to managed or unmanaged.

This command has the following format:

```
dpmsd setmanagedstatus [-sc sc_url] -package_name {packagename|all}  
-managedstatus {1|0} [-itcm_server itcm domain manager] [-pre][-post][-ws_user  
username -ws_password password][-locale iso629value]
```

-sc sc_url

(Optional) Defines the URL for the service controller host.

Example: https://hostname:port/...

hostname

Defines the fully qualified host where the service controller is installed.

Example: localhost.ca.com

port

Defines the listening port for the service controller server.

Example: 80

-package_name {packagename|all}

Defines the name of the package to deliver.

packagename

Applies the managed status change to the specified package.

all

Apply the managed status change to all packages.

-managed_status {0|1}

Defines the managed status of the package.

0

Sets the package to an unmanaged state. The package appears in the Available List in the Software Delivery area of the CA Server Automation user interface, but is not available for deployment.

1

Sets the package to a managed state. The package appears in the Available List in the Software Delivery area of the CA Server Automation user interface, and is available for deployment.

-itcm_server *itcmdomainmanager*

(Optional) Defines the name of the CA ITCM Domain Manager where the software delivery adapter resides. Optional when only one Software Delivery adapter or CA ITCM domain manager is configured.

-pre

(Optional) Specifies whether to generate an event before the operation is performed.

-post

(Optional) Specifies whether to generate an event after the operation is performed.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example: Set Status of Package

This example sets a package to managed.

```
dpmsd setmanagedstatus -package_name "Apache Tomcat" -managedstatus 1
```

Example: Set Status of Package in a Multiple ITCM Domain Manager Environment

This example sets a package to managed. This operation is routed to the sd adapter on ITCM domain manager domainmanager001.

```
dpmsd setmanagedstatus -package_name "Apache Tomcat" -managedstatus 1 - itcm_server domainmanager001
```

Storage Provisioning Manager Command

You can use the `cadpmspm` CLI command to discover storage objects, and to provision, deprovision, and resize storage using the web service. This command also lets you perform operations such as attach storage to a target host, verify the status of a storage provisioning request, and move vFilers. To use `cadpmspm`, you must have access to a storage provider server.

`cadpmspm attach` Command--Attach Storage to Host

Use this command to provision new storage and attach it to a target host.

This command has the following format:

```
cadpmspm -attach
-dataset=DatasetName
-size=StorageSizeMB
[-asynch]
-provxml=ProvisionDataXMLFile
[-target=TargetHostName]
[-attach_user=TargetHostUser -attach_password=TargetHostPassword
-attach_location=MountLocation]
[-stsrv=StorageServer -stplat=StoragePlatform]
[-ws_user=EEMusername -ws_password=EEMpassword]
[-locale iso639value]
[-attach_cifsuser=cifsuser -attach_cifspass=cifspassword]
```

-dataset=DatasetName

Specifies the name of the data set to create and attach. To get the name of the data set, perform one of the following procedures:

- Open the CA Server Automation user interface and click Management, Resources. Select Data Center in the Explore tree, open User Resources, and select Storage. The data set names appear in the Name column.
- Log in the CA Server Automation server with administrator privileges, and start the CA Server Automation Command Prompt from the Start menu. Run the following command:

```
cadpmspm.exe -discover -detail=0 -sttype=9 -stplat=StoragePlatform
```

See the `stplat` parameter description for the possible values.

-size=StorageSizeMB

Specifies the amount of storage to attach in MB.

-asynch

(Optional) Specifies that the attach run asynchronously.

-provxml=*ProvisionDataXMLFile*

Specifies the input file for provision data.

-target=*TargetHostName*

(Optional) Specifies the target to which to attach the newly provisioned storage.

-atch_user=*TargetHostUser*

(Optional) Specifies the attach username for the target host.

atch_password=*TargetHostPassword*

(Optional) Specifies the attach password for the target host.

-atch_location=*MountLocation*

(Optional) Specifies the path to which to mount the provisioned storage.

-stsrv=*StorageServer*

(Optional) Specifies the storage server name or IP address.

-stplat=*StoragePlatform*

(Optional) Specifies the storage platform; possible value is:

1

Specifies NetApp as the storage platform.

2

Specifies EMC as the storage platform.

3

Specifies HP as the storage platform.

4

Specifies IBM as the storage platform.

Default: 1

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-atch_cifsuser=CIFSUser

Specifies the user name of the CIFS storage access.

Note: This parameter is required for CIFS storage attachment.

-atch_cifspass=CIFSPassword

Specifies the password of the CIFS storage access.

Note: This parameter is required for CIFS storage attachment.

cadmpspm clone storage--Clone Storage

Use this command to clone Logical Unit Number (LUN) storage resources.

This command has the following format:

```
cadmpspm -clone  
-lunpath=lunpath  
-luninit=luninitiator  
[-ws_user=username -ws_password=password]  
[-locale iso639value]
```

-lunpath=*lunpath*

Specifies the path of the source LUN.

-luninit=*luninitiator*

The SCSI Initiator ID or World Wide Port Number (WWPN) of the target that is associated with the LUN.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

cadpmspm deprovision Command--Deprovision Existing Storage

Use this command to deprovision an existing NFS, CIFS, iSCSI and FCP storage.

This command has the following format:

```
cadpmspm -deprovision
-dataset=DatasetName
[-stsrv=StorageServer -stplat=StoragePlatform]
[-ws_user=username -ws_password=password]
[-locale iso639value]
```

-dataset=*DatasetName*

Specifies the name of the data set to create and attach. To get the name of the data set, perform one of the following procedures:

- Open the CA Server Automation user interface and click Management, Resources. Select Data Center in the Explore tree, open User Resources, and select Storage. The data set names appear in the Name column.
- Log in the CA Server Automation server with administrator privileges, and start the CA Server Automation Command Prompt from the Start menu. Run the following command:

```
cadpmspm.exe -discover -detail=0 -sttype=9 -stplat=StoragePlatform
```

See the stplat parameter description for the possible values.

-stsrv=*StorageServer*

(Optional) Specifies the storage server name or IP address.

-stplat=*StoragePlatform*

(Optional) Specifies the storage platform; possible value is:

1

Specifies NetApp as the storage platform.

2

Specifies EMC as the storage platform.

3

Specifies HP as the storage platform.

4

Specifies IBM as the storage platform.

Default: 1

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

cadpmspm discover Command--Discover Storage Objects

Use this command to list storage resources.

This command has the following format:

```
cadpmspm -discover
-sttype=DeviceType
[-detail=DiscoveryDetailLevel]
[-stsrv=StorageServer]
[-stplat=StoragePlatform]
[-ws_user=username -ws_password=password]
[-locale iso639value]
```

-sttype=*DeviceType*

Specifies the type of device.

For NetApp storage platform, the possible values are:

- 0**
Specifies data sets
- 1**
Specifies resource pools
- 2**
Specifies provisioning policy
- 3**
Specifies storage services
- 4**
Specifies vFilers
- 5**
Specifies vFiler templates
- 6**
Specifies storage systems
- 7**
Specifies protection policy
- 8**
Specifies host agents
- 9**
Specifies LUN information.
- 10**

Specifies Groups.

11

Specifies VLANs.

12

Specifies IP Spaces.

13

Specifies Masking and Mapping.

For EMC, HP, and IBM storage platforms, the possible values are:

1

Specifies resource pools.

6

Specifies storage systems.

8

Specifies host agents.

9

Specifies LUN information.

13

Specifies Masking and Mapping.

detail=DiscoveryDetailLevel

(Optional) Specifies to turn on detailed discovery; possible values are:

0

Specifies basic detail.

1

Specifies simple detail.

2

Specifies complete detail.

-stsrv=StorageServer

(Optional) Specifies the storage server name or IP address.

-stplat=StoragePlatform

(Optional) Specifies the storage platform; possible value is:

1

Specifies NetApp as the storage platform.

2

Specifies EMC as the storage platform.

3

Specifies HP as the storage platform.

4

Specifies IBM as the storage platform.

Default: 1**-ws_user username -ws_password password**

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

cadmpspm jobstatus Command--Get Job Status

Use this command to get the status of a storage provisioning request.

This command has the following format:

```
cadmpspm -jobstatus
-jsid=JobSetIdentifier
[-ws_user=username -ws_password=password]
[-locale iso639value]
```

-jsid=JobSetIdentifier

The job status identifier for which you want status.

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

cadpmspm move Command--Move Storage

Use this command to move a NetApp LUN and vFiler from source to destination.

Note: This command is supported only on NetApp Storage servers.

This command has the following format:

```
cadpmspm -move
-srcfiler=srcfiler
-desfiler=desfiler
-vlan=vlan
-vfiler=vfiler
-uid=uid
-pwd=pwd
-srcvol=srcvolume
-desvol=desvolume
-mvaction=action
[-stsrv=StorageServer -stplat=StoragePlatform]
[-asynch]
[-ws_user=username -ws_password=password]
[-locale iso639value]
```

-srcfiler=*FilerName*

Specifies the source filer name.

-desfiler=*FilerName*

Specifies the destination filer name.

-vlan=*vlaname*

Specifies the vFiler VLAN interface.

-vfiler=*vFilerName*

Specifies the vFiler name.

-uid=*UserName*

Specifies the source filer username.

-pwd=*Password*

Specifies the source filer password.

-srcvol=*VolumeName*

Specifies the source volume name.

-desvol=*VolumeName*

Specifies the destination filer volume.

-mvaction=*ActionType*

Specifies the move storage command type; possible values are:

0

Complete move.

1

Update LUN.

2

Resynchronize vFiler.

3

Get LUN update status.

4

Get vFiler resynchronization status.

5

Stop vFiler.

6

Offline volume.

7

Configure network interface.

8

Online volume.

9

Activate vFiler.

-stsrv=*StorageServer*

(Optional) Specifies the storage server name or IP address.

-stplat=StoragePlatform

(Optional) Specifies the storage platform; possible value is:

1

Specifies NetApp as the storage platform.

Default: **1**

-asynch

(Optional) Specifies that the attach run asynchronously.

-ws_user username -ws_password password

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale iso639value

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

cadmpspm provision Command--Provision New Storage

Use this command to provision NFS, CIFS, iSCSI, and FCP storage. You can provision storage through a storage service or through a provisioning policy and resource pools.

This command has the following format:

```
cadmpspm -provision  
-provxml=ProvisionDataXMLFile  
[-stsrv=StorageServer -stplat=StoragePlatform]  
[-ws_user=username -ws_password=password]  
[-locale iso639value]
```

-provxml=ProvisionDataXMLFile

Specifies the input file for provision data.

-stsrv=StorageServer

(Optional) Specifies the storage server name or IP address.

-stplat=*StoragePlatform*

(Optional) Specifies the storage platform; possible value is:

1

Specifies NetApp as the storage platform.

2

Specifies EMC as the storage platform.

3

Specifies HP as the storage platform.

4

Specifies IBM as the storage platform.

Default: 1

-ws_user *username* -ws_password *password*

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

cadpmspm resize Command--Resize Existing Storage

Use this command to resize existing NFS, CIFS, iSCSI, or FCP storage.

This command has the following format:

```
cadpmspm -resize
-dataset=DatasetName
-size=StorageSizeMB
-rsvperct=ReservePercentage
[-maxcap=MaxCapacityMB]
[-stsrv=StorageServer -stplat=StoragePlatform]
[-ws_user=username -ws_password=password]
[-locale iso639value]
```

-dataset=*DatasetName*

Specifies the name of the data set to create and attach. To get the name of the data set, perform one of the following procedures:

- Open the CA Server Automation user interface and click Management, Resources. Select Data Center in the Explore tree, open User Resources, and select Storage. The data set names appear in the Name column.
- Log in the CA Server Automation server with administrator privileges, and start the CA Server Automation Command Prompt from the Start menu. Run the following command:

```
cadpmspm.exe -discover -detail=0 -sttype=9 -stplat=StoragePlatform
```

See the stplat parameter description for the possible values.

-size=*StorageSizeMB*

Specifies the amount of storage to attach in MB.

-rsvperct=*ReservePercentage*

Specifies the snapshot reserve percentage; possible values are 1-100.

-maxcap=*MaxCapacityMB*

Specifies the new maximum capacity in MB.

-stsrv=*StorageServer*

(Optional) Specifies the storage server name or IP address.

-stplat=StoragePlatform

(Optional) Specifies the storage platform; possible value is:

1

Specifies NetApp as the storage platform.

2

Specifies EMC as the storage platform.

3

Specifies HP as the storage platform.

4

Specifies IBM as the storage platform.

Default: 1**-ws_user *username* -ws_password *password***

(Optional) Specifies the credentials to use for the web service security check. If you do not include credentials, you are prompted to enter them. Avoid the prompt for credentials by setting up your own session using caaipsecurity.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Log Files

Log files are generated for troubleshooting problems; they are useful for communications with Technical Support. Log files listed are located in the following directories:

Install_Path\CA\productname\apache\Rainier.log

Contains log messages for product components (for example, Imaging).

Install_Path\CA\productname\bin\Rainier*.log

Contains log messages for CLIs.

Install_Path\CA\productname\log\UITrace.log

Contains log messages for the user interface.

Log File Settings

You can configure log files by editing the configuration file. Use the `caaipcfgclient` command to set the following properties that are related to logging:

CONFIG_KEY_TRACE_LEVEL

Sets the level of detail to trace. Valid values are: trace, debug, information, notice, warning, error, critical, fatal.

Example: `caaipcfgclient -setproperty -productAIP -useruser1 -passwordpass -namevalueCONFIG_KEY_TRACE_LEVEL:information`

Note: Setting the overall level to debug or trace can noticeably slow down the product.

CONFIG_KEY_TRACE_FILENAME

Defines the trace file name and path.

Example: `caaipcfgclient -setproperty -productAIP -useruser1 -passwordpass -namevalue CONFIG_KEY_TRACE_FILENAME:./Rainier.log`

CONFIG_KEY_TRACE_INCLUDES

Enables force trace for these IDs. Zero or more regular expressions separated by semicolons.

Example: `caaipcfgclient -setproperty -productAIP -useruser1 -passwordpass -namevalueCONFIG_KEY_TRACE_INCLUDES:AOM-001;CTL-.*`

CONFIG_KEY_TRACE_EXCLUDES

Disables force trace for these IDs. Zero or more regular expressions separated by semicolons.

Chapter 4: Command Line Utilities

This section contains the following topics:

- [dpmkpdb Command--Import or Export Performance Data](#) (see page 1058)
- [dpmutil Syntax](#) (see page 1059)
- [dpmutil -amq Command--Configure the ActiveMQ Message Broker](#) (see page 1060)
- [dpmutil -ccm Command--Configure CA Configuration Automation](#) (see page 1061)
- [dpmutil -dhcp_unix Command--Configure DHCP for UNIX](#) (see page 1062)
- [dpmutil ec2-owner Command--Configure EC2 Owner](#) (see page 1063)
- [dpmutil ec2-private-keypair Command--Configure Private Key Pair for EC2](#) (see page 1065)
- [dpmutil -emc-smi Command--Configure EMC SMI-S Provider Server](#) (see page 1066)
- [dpmutil ecs Command--Configure Windows Scheduler](#) (see page 1066)
- [dpmutil -hd Command--Configure Help Desk](#) (see page 1067)
- [dpmutil -hmc Command--Configure HMC or IVM Server](#) (see page 1068)
- [dpmutil -hyperv Command--Configure Hyper-V Server](#) (see page 1069)
- [dpmutil -itpam-cfg-eem Command--Configure CA Process Automation/CA EEM](#) (see page 1070)
- [dpmutil -itpamserver Command--Configure CA Process Automation](#) (see page 1071)
- [dpmutil -js Command--Configure JumpStart](#) (see page 1072)
- [dpmutil mgmtldb Command--Configure Management Database](#) (see page 1072)
- [dpmutil netapp Command--Configure NetApp DataFabric Manager](#) (see page 1073)
- [dpmutil -nim Command--Configure NIM Server](#) (see page 1074)
- [dpmutil -perfdb Command--Configure the Performance Database](#) (see page 1074)
- [dpmutil -proxy Command--Configure Proxy Server](#) (see page 1075)
- [dpmutil remoteucsaaim Command--Configure Remote Cisco Unified Computing System AIM](#) (see page 1076)
- [dpmutil remotevcaim Command--Configure Remote VC AIM](#) (see page 1077)
- [dpmutil -sc Command--Configure the Service Controller](#) (see page 1077)
- [dpmutil -scvmmserver Command--Configure SCVMM Server](#) (see page 1078)
- [dpmutil -sdscale Command--Configure the Software Delivery Scalability Server](#) (see page 1079)
- [dpmutil -sdserver Command--Configure Software Delivery Server](#) (see page 1080)
- [dpmutil -snmp Command--Configure SNMP](#) (see page 1081)
- [dpmutil -superuser Command--Configure SuperUser Credentials](#) (see page 1083)
- [dpmutil -systemcredential Command--Configure External System Credentials](#) (see page 1084)
- [dpmutil -sysuser Command--Configure Local Credentials Used at Service Startup](#) (see page 1085)
- [dpmutil ucs Command--Configure Cisco Unified Computing System Server](#) (see page 1086)
- [dpmutil upm Command--Configure the CA Patch Manager DB](#) (see page 1087)
- [dpmutil vcdefault Command--Configure Default VMware vCenter Administrator User](#) (see page 1087)
- [dpmutil vcserver Command--Configure VMware vCenter](#) (see page 1088)
- [dpmutil -vioserver Command--Configure VIO Server](#) (see page 1089)

dpmkpdb Command--Import or Export Performance Data

dpmkpdb.exe is a command-line utility that lets you import and export your performance data. This utility lets you preserve performance data when upgrading CA Server Automation to Release 12.8.1. To access dpmkpdb.exe, you must have administrator privileges. The dpmkpdb.exe command-line utility syntax accepts *import*, *export*, or *export_ce* keywords as the first argument.

This command has the following format:

```
dpmkpdb.exe {import | export | export_ce}
-ws_user username -ws_password password
-endpoint endpoint_URI
[-locale iso639value]
{-input input_file_name | -output output_file_name}
```

import

Imports the performance data from the input file.

export

Exports the performance data to the output file.

export_ce

Exports the performance data from the collection engine to the output file.

Important! The Key Performance Database collects the performance data. The Collection Engine that has collected the performance data in previous releases is still available for compatibility reasons.

-ws_user *username* -ws_password *password*

Specifies the credentials that are used to log in the system.

-endpoint *endpoint_URI*

Overrides the endpoint URI.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

-input *input_file_name*

Specifies the input file for importing the performance data.

-output *output_file_name*

Specifies the output file for exporting the performance data.

dpmutil Syntax

dpmutil is a command-line utility that lets you change the configuration of CA Server Automation components. To use dpmutil, Management DB information must be available in the Registry, and the Microsoft SQL client must be installed. To access dpmutil, you must have a user name with administrator privileges. The dpmutil command-line utility syntax accepts either *set*, *get*, or *delete* requests as a first argument, and then lets you set or get information about the components. If you input invalid or no arguments, an error message appears with syntax information and the list of components that you can configure. The dpmutil syntax includes the following:

```
dpmutil {-set|-get|-delete}  
-category|component  
[-username] [-passwordpassword]  
[-help]  
[-validateonly]  
[-locale iso639value]
```

-set

Defines or updates the configuration settings for the component.

-get

Displays the configuration settings for the component.

-delete

Deletes the configuration settings for the component.

-category|component

Defines the category or component from which you want to obtain or change data.

-username

(Optional) Specifies the EEM user. Without this parameter, dpmutil prompts you for valid EEM credentials before allowing access to configuration updates and deletes, or a display of existing configuration information.

-passwordpassword

(Optional) Specifies the EEM password. Without this parameter, dpmutil prompts you for valid EEM credentials before allowing access to configuration updates and deletes, or a display of existing configuration information.

Note: There is no space between the user and password parameters and their value.

-help

Displays the dpmutil help.

-validateonly

(Optional) Validates the information entered when used with set and a valid command as follows:

```
dpmutil -set command -validateonly
```

This command does not create or update information.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Note: Restart the Apache HTTP Server for the changes to take effect.

dpmutil -amq Command--Configure the ActiveMQ Message Broker

The dpmutil set|get|delete amq command lets you configure the ActiveMQ Message Broker for CA Server Automation. Some CA Server Automation components use the Message Broker to pass data between them. This command sets the Message Broker URL.

This command has the following format:

```
dpmutil {-set|-get|-delete} -amq [-locale iso639value]
```

-set

Defines the URL for the ActiveMQ Message Broker.

Example: *tcp://servername:port*

servername

Identifies the name of the server where the ActiveMQ Message Broker is installed.

port

Specifies the ActiveMQ Message Broker listening port.

Default: 61616

-get

Displays the URL configured for the ActiveMQ Message Broker.

-delete

Deletes the URL configured for the ActiveMQ Message Broker.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -ccm Command--Configure CA Configuration Automation

The dpmutil set|get -ccm command configures CA Configuration Automation for CA Server Automation. You can also launch CA Configuration Automation from the CA Server Automation UI in read-only mode without logging in with a key.

This command has the following format:

```
dpmutil {-set|-get|-delete} -ccm [-locale iso639value]
```

-set

Defines the configuration settings for the CCA server and requests key information to launch the server in read-only mode without logging in.

-get

Displays the configuration settings for the CCA server including server name (name), user name (specialist), port (8080), and key (dcam_acm). You are prompted for the CA Server Automation administrator user name and password.

-delete

Deletes CCA servers from CA Server Automation, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-ccm

Defines the change and configuration management database. You are prompted for user credentials for the CCA server. The user role must be *specialist* or *superuser*.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -dhcp_unix Command--Configure DHCP for UNIX

The dpmutil set|get dhcp_unix command configures the Dynamic Host Configuration Protocol (DHCP) servers that assign IP addresses to JumpStart client computers. The command also provides a reference to the JumpStart boot server and boot file for these client computers. Your CA Server Automation administrator user name and password are required to set or get information.

This command has the following format:

```
dpmutil {-set|-get|-delete} -dhcp_unix [-locale iso639value]
```

-set

Configures the server name, user name, and password for the DHCP server.

-get

Lists the configuration settings for the DHCP servers including the assigned server name and user name.

UNIX username: root

-delete

Deletes DHCP servers from CA Server Automation, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-dhcp_unix

Defines the type of operating system of the DHCP server that you are configuring.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil ec2-owner Command--Configure EC2 Owner

The dpmutil set|get EC2 owner command configures CA Server Automation with the EC2 account information that you receive from Amazon after creating your Amazon account. You must have a CA Server Automation administrator role to use this command.

This command has the following format:

```
dpmutil {-set|-get|-delete} -ec2-owner [-locale iso639value]
```

-set

Configures, displays, or deletes the following Amazon EC2 account information.

AWS Account ID

Defines the Amazon Web Services (AWS) account ID.

Example: 495219933132AWS

Account Name

Defines an alias name to the account ID.

Example: Account of Finance Department

Description

Description of the account.

Example: Account used by all full-time employee

Type Y (case-sensitive) to set as default EC2 server, otherwise press enter:

Defines the current account as the default.

EC2 Server URL (default)

Defines the URL for the default and other AWS servers.

Example: http://ec2.amazonaws.com

EC2 region(s) configured for this account

Lists the EC2 regions that you have configured.

x.509 Certificate File (full path)

Defines the path to the Amazon X.509 certificate on the CA Server Automation service controller.

Example: C:\my_ec2_account\cert-U75832H20496JNR3932AER5.pem

EC2 Private Key File (full path)

Defines the path to the Amazon private key on the CA Server Automation service controller.

Example: C:\my_ec2_account\pk-U75832H20496JNR3932AER5.pem

EC2 image list type

Defines whether CA Server Automation displays only private AMIs, or both private and public. The default is private. Valid entries: private or private + public.

Note: Specifying private + public affects product performance because of the volume of AMIs; private is the recommended value.

Example: private

EC2 synchronization frequency in seconds

Defines the frequency in seconds to synchronize records between CA Server Automation and Amazon. The default is 1800 seconds.

Note: Synchronizing records affects product performance; decreasing this value is not recommended.

Example: 1800

-get

Lists the configuration information for the EC2 owner.

-delete

Deletes the configuration information for the EC2 owner.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil ec2-private-keypair Command--Configure Private Key Pair for EC2

The dpmutil set|get ec2-private-keypair command configures an RSA private key for EC2. The key pair private key used in this command, is different from the private key when you set up an account with Amazon. This command stores the key pair name and the private key in CA Server Automation database. Set the location of the private key pair file so CA Server Automation can locate the password for the key pair. You must have a CA Server Automation administrator role to use this command.

This command has the following format:

```
dpmutil {-set|-get|-delete} -ec2-private-keypair [-locale iso639value]
```

-set

Sets the EC2 private key pair file location on the EC2 server. The key pair file contains the unencrypted PEM encoded private key part of the 2048 RSA key pair. The key pair name must exist, or an error is generated. You are prompted for the following information:

EC2 Key Pair Name

Defines the key pair name that is associated with a 2048 RSA key pair. The key pair name is a unique identifier required for users to launch new EC2 instances.

EC2 Key Pair File

Defines the location of the file that contains the unencrypted PEM encoded private key part of the 2048 RSA key pair. The private key is generated when you create a key pair in EC2 using Amazon EC2 management tools.

EC2 Key Pair Name (if multiple)

Defines multiple key pairs in EC2. You can use the Amazon command line utilities to create multiple key pairs after you set up an Amazon account to use EC2. These key pair names are used when launching new EC2 instances.

EC2 Key Pair File (if multiple)

Defines the locations of the files that contain the unencrypted PEM encoded private key part of the 2048 RSA key pair. The private key is generated when you create a key pair in EC2 using Amazon EC2 management tools.

-get

Lists the settings for the private key pair.

-delete

Deletes the settings for the private key pair.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -emc-smi Command--Configure EMC SMI-S Provider Server

The dpmutil set|get|delete -emc-smi command configures the EMC SMI-S Provider Server for use with the CA Server Automation Storage Provisioning Manager.

This command has the following format:

```
dpmutil {-set|-get|-delete} -emc-smi [-locale iso639value]
```

-set

Defines the configuration settings for the EMC SMI-S Provider Server.

-get

Displays the configuration settings for EMC SMI-S Provider Server including server name, user name, port, and protocol.

-delete

Deletes the configuration settings for the EMC SMI-S Provider Server.

-emc-smi

Specifies the EMC SMI-S Provider Server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil ecs Command--Configure Windows Scheduler

The dpmutil set|get ecs command configures the scheduler server.

This command has the following format:

```
dpmutil {-set|-get} -ecs [-locale iso639value]
```

-set

Defines the configuration settings for the scheduler server.

-get

Displays the configuration settings for the scheduler server.

-ecs

Defines the scheduler server to configure. You are prompted to decide whether ECS is the active scheduler (type Y for yes or hit Enter to skip configuration).

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -hd Command--Configure Help Desk

The `dpmutil set|get hd` command configures help desk information for CA Server Automation. You can use this command to configure the help desk system on an existing installation of CA Server Automation.

This command has the following format:

```
dpmutil {-set|-get|-delete} -hd [locale iso639value]
```

-set

Defines the configuration settings for the help desk system.

-get

Displays the configuration settings for the help desk system.

-delete

Deletes help desk servers from CA Server Automation, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-hd

Defines the information required to connect to the help desk server. You are prompted for the help desk server name, the administrator user name and password, and the port.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -hmc Command--Configure HMC or IVM Server

The `dpmutil hmc` command adds, modifies, lists, or deletes the configured HMC or IVM servers. You can use this command to configure HMC or IVM server on an existing installation of CA Server Automation. Your CA Server Automation administrator user name and password and the port is required to set or get information.

Note: Restart the Apache HTTP Server for the changes to take effect.

This command has the following format:

```
dpmutil {-set|-get|-delete} -hmc [-locale iso639value]
```

-set

Adds or changes the configuration settings for HMC or IVM servers including the server name, the user name, and password.

-get

Displays the configuration settings for HMC or IVM servers including the server name, the user name.

-delete

Deletes HMC or IVM servers, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-hmc

Defines the HMC or IVM servers.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmutil -hyperv Command--Configure Hyper-V Server

The `dpmutil set|get -hvserver` command adds, modifies, lists, or deletes the configured Hyper-V servers for virtual machine (VM) provisioning. You can use this command to install the Hyper-V component on an existing installation of CA Server Automation. Your CA Server Automation administrator user name and password and the port is required to set or get information. You do not need to restart Apache after you change this setting.

This command has the following format:

```
dpmutil {-set|-get|-delete} -hyperv [locale iso639value]
```

-set

Adds or changes the configuration settings for Hyper-V servers including the server name, the user name, the port, and the protocol.

-get

Displays the configuration settings for Hyper-V servers including the server name, the user name, the port, and the protocol.

-delete

Deletes Hyper-V servers, but they remain in the Management DB and Performance DB for historical purposes. You will be prompted for each parameter.

-hyperv

Defines the Hyper-V server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmutil -itpam-cfg-eem Command--Configure CA Process Automation/CA EEM

The dpmutil set|get itpam-cfg-eem command configures the single sign-on for the CA Process Automation server that CA Server Automation uses. Single sign-on allows users to provide one set of credentials to access CA Server Automation and CA Process Automation.

This command has the following format:

```
dpmutil {-set|-get|-delete} -itpam-cfg-eem [-locale iso639value]
```

-set

Defines the configuration settings for the CA Process Automation server.

-get

Lists the configuration settings for the CA EEM server. You are prompted for the following information:

CA EEM Backend Server for CA Process Automation integration

Defines the CA EEM server name.

CA EEM Application Instance for CA Process Automation integration [ITPAM]

Defines the application name created for CA Process Automation in the CA EEM server.

CA EEM Certificate File for CA Process Automation integration (full path)

Defines the path to the itpamcert.p12 file that allows CA EEM to log in to CA Process Automation.

CA EEM Certificate Password for CA Process Automation integration

The certificate password for integration.

Default: itpamcertpass

-delete

Deletes the configuration settings for the CA EEM server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -itpamserver Command--Configure CA Process Automation

The dpmutil set|get itpam command configures the CA Process Automation server that CA Server Automation uses. This server orchestrates, manages, and reports on automated processes.

This command has the following format:

```
dpmutil {-set|-get|-delete} -itpamserver [-locale iso639value]
```

-set

Defines the configuration settings for the CA Process Automation server.

-get

Lists the configuration settings for the CA Process Automation server. You are prompted for the CA Process Automation server name, administrator user name and administrator password and port. The default port for CA Process Automation is 8080.

-delete

Deletes CA Process Automation servers from CA Server Automation, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -js Command--Configure JumpStart

The `dpmutil set|get js` command configures Solaris JumpStart servers for provisioning. Set a JumpStart server as a boot server or an installation server that is also a boot server. Use this command to install the JumpStart component on an existing installation of CA Server Automation.

Note: Restart the Apache HTTP Server for the changes to take effect.

This command has the following format:

```
dpmutil {set|get|delete} -js [-locale iso639value]
```

set

Configures the server name, administrator user name, type of image (boot, install, or both), and password for the Solaris JumpStart server. You are prompted for each parameter.

get

Lists the configuration settings for the Solaris JumpStart servers including the server name, type, user name and port.

delete

Deletes Solaris JumpStart servers from CA Server Automation, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

js

Defines the Solaris JumpStart server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

dpmutil mgmtdb Command--Configure Management Database

The `dpmutil set|get -mgmtdb` command configures the Management Database that almost all components in the product suite use as the main database.

This command has the following format:

```
dpmutil {-set|-get} -mgmtdb [-locale iso639value]
```

-set

Defines the configuration settings for the Management Database.

-get

Displays the configuration settings for the Management Database.

-mgmtdb

Defines the Management Database to configure. You are prompted for the database server, version, port, and (if you use SQL authorization) the credentials to use to access the main product tables.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Note: Changing this information requires the dependent product services (CAAIPApache and CAAIPTomcat) to be recycled for the changes to take effect.

dpmutil netapp Command--Configure NetApp DataFabric Manager

The dpmutil set|get|delete -netapp command configures the NetApp DataFabric Manager for use with the CA Server Automation Storage Provisioning Manager.

This command has the following format:

```
dpmutil {-set|-get|-delete} -netapp [-locale iso639value]
```

-set

Defines the configuration settings for the NetApp DataFabric Manager.

-get

Displays the configuration settings for the NetApp DataFabric Manager including server name, user name, port, and protocol.

-delete

Deletes the NetApp DataFabric Manager.

-NetApp

Defines the NetApp DataFabric Manager.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -nim Command--Configure NIM Server

The dpmutil -nim command adds, modifies, lists, or deletes the configured NIM servers. You can use this command to configure the NIM Server on an existing installation of CA Server Automation. Your CA Server Automation administrator user name and password and the port is required to set or get information.

Note: Restart the Apache HTTP Server for the changes to take effect.

This command has the following format:

```
dpmutil {-set|-get|-delete} -nim [-locale iso639value]
```

-set

Adds or changes the configuration settings for NIM servers including the server name, the user name, and password.

-get

Displays the configuration settings for NIM servers including the server name, the user name.

-delete

Deletes NIM servers, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-nim

Defines the NIM servers.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -perfdb Command--Configure the Performance Database

The dpmutil set|get -perfdb command configures the Performance DB for CA Server Automation. This database stores all of the performance information collected by CA Server Automation.

This command has the following format:

```
dpmutil {-set|-get|-delete} -perfdb [-locale iso639value]
```

-set

Defines the configuration settings for the Performance DB.

-get

Displays the configuration settings for the Performance DB.

-delete

Deletes the configuration settings for the Performance DB.

-perfdb

Defines the component you want to configure. You are prompted for the server name, administrator user name and password, database type, database instance, and database port. The Performance DB credentials are typically the same as the Management DB credentials.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Note: Changing this information requires the dependent product services (CAAIPApache and CAIPTomcat) to be recycled for the changes to take effect.

dpmutil -proxy Command--Configure Proxy Server

The dpmutil proxy command configures, displays, or deletes common proxy server information. You must have a CA Server Automation administrator role to use this command.

This command has the following format:

```
dpmutil {-set|-get|-delete} -proxy [-locale iso639value]
```

-set

Configures a common proxy server for network connections to the Internet (for example, behind a corporate firewall) with the following information:

Proxy host

Identifies the host name for the proxy server.

Proxy port

Identifies the listening port for the proxy server.

Proxy user name

Identifies the user name for the proxy server.

Proxy password

Identifies the password for the proxy server.

-get

Lists the configuration information for the proxy server.

-delete

Deletes the configuration information for the proxy server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil remoteucsaim Command--Configure Remote Cisco Unified Computing System AIM

The dpmutil set|get remoteucsaim command configures the remote Cisco Unified Computing System AIM.

This command has the following format:

```
dpmutil {-set|-get|-delete} -remoteucsaim [-locale iso639value]
```

-set

Defines the configuration settings for the remote Cisco Unified Computing System AIM.

-get

Displays the configuration settings for the remote Cisco Unified Computing System AIM.

-delete

Deletes a currently configured remote Cisco Unified Computing System AIM.

-remoteucsaim

Defines the remote Cisco Unified Computing System AIM to configure. You are prompted for the server where UCS AIM is installed, the server where UCS itself is installed, and the SNMP write access string to use to connect to the UCS server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil remotevcaim Command--Configure Remote VC AIM

The dpmutil set|get remotevcaim command configures the remote VC AIM.

This command has the following format:

```
dpmutil {-set|-get|-delete} -remotevcaim [-locale iso639value]
```

-set

Defines the configuration settings for the remote VC AIM.

-get

Displays the configuration settings for the remote VC AIM.

-delete

Deletes a currently configured remote VC AIM.

-remotevcaim

Defines the remote VC AIM to configure. You are prompted for the server where VC AIM is installed, the server where VC itself is installed, and the SNMP write access string to use for connecting to the VC server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -sc Command--Configure the Service Controller

The dpmutil set|get -sc command lets you configure the service controller for CA Server Automation. The service controller is the communications traffic controller that lets the different CA Server Automation components communicate with each other.

This command has the following format:

```
dpmutil {-set|-get|-delete} -sc [-locale iso639value]
```

-set

Defines the configuration settings for the service controller.

-get

Displays the configuration settings for the service controller.

-delete

Deletes the configuration settings for the service controller.

-sc

Defines the service controller for the local server.

Mode

Defines configuration settings for the method used to locate the service controller. The following modes are supported:

SRV automatic lookup

Locates the service controller using a DNS SRV record.

Manual override

Identifies the installed and running service controller based on the specified URL.

Example: `https://hostname:port/dpm/sc`

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -scvmmserver Command--Configure SCVMM Server

The `dpmutil set|get -scvmmserver` command adds, modifies, lists, or deletes the configured SCVMM servers for virtual machine (VM) provisioning. You can use this command to install the SCVMM component on an existing installation of CA Server Automation. Your CA Server Automation administrator user name and password and the port is required to set or get information. You do not need to restart Apache after you change this setting.

This command has the following format:

```
dpmutil {-set|-get|-delete} -scvmmserver [-locale iso639value]
```

-set

Adds or changes the configuration settings for SCVMM servers including the server name, the user name, the port, and the protocol.

-get

Displays the configuration settings for SCVMM servers including the server name, the user name, the port, and the protocol.

-delete

Deletes SCVMM servers, but they remain in the Management DB and Performance DB for historical purposes. You will be prompted for each parameter.

-scvmmserver

Defines the SCVMM servers.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -sdscale Command--Configure the Software Delivery Scalability Server

The dpmutil set|get -sdscale command configures the Software Delivery Scalability server.

This command has the following format:

```
dpmutil {-set|-get|-delete} -sdscale [-locale iso639value]
```

-set

Defines the configuration settings for the Software Delivery Scalability server.

-get

Displays the configuration settings for the Software Delivery Scalability server.

-delete

Deletes a currently configured remote Software Delivery Scalability server.

-sdscale

Defines the Software Delivery Scalability server to configure. You are prompted for the name of the SD scalability server and the user credentials to use for access.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -sdserver Command--Configure Software Delivery Server

The dpmutil set|get -sdserver command configures the Software Delivery server.

This command has the following format:

```
dpmutil {-set|-get|-delete} -sdserver [-locale iso639value]
```

-set

Defines the configuration settings for the Software Delivery server.

-get

Displays the configuration settings for the Software Delivery server.

-delete

Deletes a currently configured remote Software Delivery server.

-sdserver

Defines the remote Software Delivery server to configure. You are prompted for the name of the sd server, and the port, protocol and user credentials to use for access.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -snmp Command--Configure SNMP

The `dpmutil set|get|delete -snmp` command configures data center or server-level SNMP settings for receiving SNMP information.

This command has the following format:

```
dpmutil {-set|-get|-delete} -snmp [-locale iso639value]
```

-set

Adds or changes the configuration settings for the data center or a server, including:

- data center or server level
- server name if it is a server setting
- SNMP name
- timeout
- retries
- write access
- version
- security user
- authorization type
- authorization password
- privacy type
- privacy password
- community string

You are prompted for each parameter. You can configure a server to use both server and data center SNMP configuration or server SNMP configuration only.

-get

Lists the configuration settings of SNMP for the data center or a server, including:

- SNMP name
- version
- port
- timeout
- retries
- write access
- authorization type
- privacy type
- security user
- community string

You can only list a server SNMP configuration even though the server uses the data center default SNMP configuration.

-delete

Deletes a data center or a server SNMP configuration. You are prompted to delete a data center or server SNMP configuration. For server SNMP configuration, you are prompted for the server name and SNMP port. For data center SNMP configuration, you are prompted for SNMP name and port.

-snmp

Defines the SNMP configuration.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -superuser Command--Configure SuperUser Credentials

When Reservation Manager provisions systems for users, it requires Administrator or root credentials for the new systems. The credentials are included in the email messages sent to end users to notify them that the systems are ready for use.

Administrators can provide credentials to the Reservation Manager in one of two ways:

- Use the `dpmutil set superuser` command to define the credentials that are used to access the new systems.
- Let the users select their own passwords when submitting reservation requests for Windows systems or when provisioning Linux systems using the CA OSIM provisioning technology.

In addition to retrieving credentials for inclusion in emails, Reservation Manager uses credentials defined with `dpmutil set superuser` in the following situations when end users are not allowed to select their own Administrator or root password:

- On Windows, Reservation Manager configures the new systems with the superuser password specified with the command `dpmutil set superuser`.
- On all systems, Reservation Manager uses these credentials to connect as a privileged user to perform additional operations such as deploying software to the system.

This command has the following format:

```
dpmutil {-set|-get|-delete} -superuser [-locale iso639value]
```

-set

Defines superuser credentials to store for later use.

-get

Displays superuser credential information. Passwords are not displayed.

-delete

Deletes superuser credentials from CA Server Automation, but they remain in the Management Database for historical purposes.

-superuser

Defines the credentials that to use when deploying software to target systems.

When using the set superuser option, you are prompted for the following data:

- A user name and password with CA Server Automation administrator privileges.
- Information that scopes the usage of the credentials that are entered. Reservation Manager supports defining distinct credentials for the following platforms: Windows, Linux, AIX, and Solaris.
- Credentials with administrator privileges that are used to deploy the Software Delivery agents to the target systems in the Reservation Manager resource pools. These credentials are stored in the Management Database in an encrypted form.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Example

The following command stores the Administrator or root password in the database for later retrieval. Enter this command when users are not allowed to select their own password.

```
dpmutil -set -superuser
```

The command prompts for credentials with administrative privileges.

dpmutil -systemcredential Command--Configure External System Credentials

The dpmutil set|get -systemcredential command configures the external system credentials for both global and system-specific access.

This command has the following format:

```
dpmutil {-set|-get|-delete} -systemcredential [-locale iso639value]
```

-set

Defines the configuration settings for the external system credentials for both global and system-specific access.

-get

Displays the configuration settings for the external system credentials for both global and system-specific access.

-delete

Deletes the external system credentials for both global and system-specific access depending on the credential chosen to delete.

-systemcredential

Defines the external system credentials for both global and system-specific access to configure. You are prompted for the credential level (inherited by the entire data center or only at the server level). Depending on the credential level, you may be prompted to enter a specific server name for which to apply the credentials. In addition, you are prompted for the credential name, the credential information itself (user/password), and optionally, any groups to which the credentials belong.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -sysuser Command--Configure Local Credentials Used at Service Startup

The dpmutil set|get -sysuser command configures the local credentials used at service startup for the main product CAAIPApache service.

This command has the following format:

```
dpmutil {-set|-get} -sysuser [-locale iso639value]
```

-set

Defines the configuration settings for the local credentials used at service startup.

-get

Displays the configuration settings for the local credentials used at service startup.

-sysuser

Defines the local credentials used at service startup to configure. You are prompted for the username and password. This user and password must be valid to log in to the configured EEM Server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

Note: Changing this information requires the dependent product services (CAAIPApache and CAAIPTomcat) to be recycled for the changes to take effect.

dpmutil ucs Command--Configure Cisco Unified Computing System Server

The dpmutil set|get ucs command configures the Cisco Unified Computing System server.

This command has the following format:

```
dpmutil {-set|-get|-delete} -ucs [-locale iso639value]
```

-set

Defines the configuration settings for the Cisco Unified Computing System server.

-get

Displays the configuration settings for the Cisco Unified Computing System server.

-delete

Deletes a currently configured Cisco Unified Computing System server.

-ucs

Defines the Cisco Unified Computing System server to configure. You are prompted for the server where UCS resides, the user credentials, and the port to use for access to the server.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil upm Command--Configure the CA Patch Manager DB

The dpmutil set|get upm command lets you configure the CA Patch Manager DB server for CA Server Automation. This DB stores all the information about the managed resources.

This command has the following format:

```
dpmutil {-set|-get|-delete} -upm [-locale iso639value]
```

-set

Defines the configuration settings for the CA Patch Manager DB.

-get

Displays the configuration settings for the CA Patch Manager DB.

-delete

Deletes CA Patch Manager DB servers from CA Server Automation, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-upm

Defines the configuration settings for the CA Patch Manager DB on the local server. You are prompted for the DB server name, the administrator user name and password, the DB type, instance, and port.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil vcdefault Command--Configure Default VMware vCenter Administrator User

The dpmutil set|get|delete vcdefault command lets you configure the default VMware vCenter administrator user for CA Server Automation. CA Server Automation uses the default VMware vCenter administrator user (vcdefault) when a VMware vCenter server is discovered. This command sets the vcdefault.

This command has the following format:

```
dpmutil {-set|-get|-delete} -vcdefault [-locale iso639value]
```

-set

Defines a vcdefault user for VMware vCenter.

-get

Displays the default user or users configured for VMware vCenter discovery.

-delete

Deletes the default user or users configured for VMware vCenter discovery.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil vcserver Command--Configure VMware vCenter

The dpmutil set|get vcserver command adds, modifies, lists, or deletes the configured VMware vCenter servers for virtual machine (VM) provisioning. You can use this command to install the VMware VirtualCenter component on an existing installation of CA Server Automation. Your CA Server Automation administrator user name and password and the port is required to set or get information. You do not need to restart Apache after you change this setting.

This command has the following format:

```
dpmutil {-set|-get|-delete} -vcserver [-locale iso639value]
```

-set

Adds or changes the configuration settings for VMware vCenter servers including the server name, the user name, the port, and the protocol.

-get

Displays the configuration settings for VMware vCenter servers including the server name, the user name, the port, and the protocol.

-delete

Deletes VMware vCenter servers, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-vcserver

Defines the VMware vCenter servers.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, fr_FR for French. To use the locale of the command prompt, specify "native".

dpmutil -vioserver Command--Configure VIO Server

The `dpmutil -vioserver` command adds, modifies, lists, or deletes the configured VIO servers. You can use this command to configure VIO server on an existing installation of CA Server Automation. Your CA Server Automation administrator user name and password and the port is required to set or get information.

Note: Restart the Apache HTTP Server for the changes to take effect.

This command has the following format:

```
dpmutil {-set|-get|-delete} -vioserver [-locale iso639value]
```

-set

Adds or changes the configuration settings for VIO servers including the server name, the user name, and password.

-get

Displays the configuration settings for VIO servers including the server name, the user name.

-delete

Deletes VIO servers, but they remain in the Management DB and Performance DB for historical purposes. You are prompted for each parameter.

-vioserver

Defines the VIO servers.

-locale *iso639value*

(Optional) Specifies an ISO 639_3166 combination to override the default English output, for example, `fr_FR` for French. To use the locale of the command prompt, specify "native".

Index

!

- ! Command--Invoke Command or Child Process (Cmdlet) • 51
- !! Command--Invoke Command or Child Process and Auto-capture Output (Cmdlet) • 50

?

- ? Command--Write Output to stdout (Cmdlet) • 52
- ?? Command--Write Output in a List to stdout (Cmdlet) • 52

A

- About AutoShell • 21
- accept Command--Read String From stdin and Assign it to a Variable (Cmdlet) • 53
- Accessing AutoShell • 24
- Accessing the Operating System • 36
- Advanced SNMP Parameters • 833
- Amazon Elastic Compute Cloud Commands • 384
- arrdump Command--Display an Array (Cmdlet) • 54
- ASSERT Command--Compare Mathematical Expressions (Funclet) • 55
- AutoShell • 21
- AutoShell Classes • 135
- AutoShell Core Reference • 46
- AutoShell Functions • 111
- AutoShell Loadable Modules' Command Reference • 160

B

- base64Decode--Decode a base64 Encoded String (Function) • 111
- base64Encode--Perform a base64 Encoding of a String (Function) • 112
- Batch Mode • 31

C

- CA Cisco UCS Commands • 395
- CA Hyper-V CLI Commands • 436
- CA IBM LPAR AutoShell Commands • 300
- CA IBM LPAR CLI Commands • 508
- CA Microsoft Cluster Server CLI Commands • 547
- CA Microsoft Hyper-V AutoShell Commands • 200

- CA Oracle Solaris Zones AutoShell Commands • 327
- CA Process Automation Commands • 821
- CA Solaris Zones CLI Commands • 567
- CA Technologies Product References • 3
- CA VMware vCenter Server AutoShell Commands • 160
- CA VMware vCenter Server CLI Commands • 596
- caaipaomwsclient Command--Deleting Objects • 797
- caaipaomwsclient Command--Display Class Information • 801
- caaipaomwsclient Command--Display Detailed Status of a Web Service • 802
- caaipaomwsclient Command--Display Propagated State • 801
- caaipaomwsclient Command--Enumerating Objects • 795
- caaipaomwsclient Command--List Associated Objects • 798
- caaipaomwsclient Command--List Available Associations • 800
- caaipaomwsclient Command--Summarize Class Use • 799
- caaipaomwsclient Command--Summarize Indications • 800
- caaipaomwsclient Command--Summarize Object Associations • 799
- caaipaomwsclient Command--Updating Objects • 796
- caaipsecurity Command--Specifies Credentials for Web Service Security Check • 686
- cadpmspm attach Command--Attach Storage to Host • 1042
- cadpmspm clone storage--Clone Storage • 1044
- cadpmspm deprovision Command--Deprovision Existing Storage • 1045
- cadpmspm discover Command--Discover Storage Objects • 1047
- cadpmspm jobstatus Command--Get Job Status • 1049
- cadpmspm move Command--Move Storage • 1050
- cadpmspm provision Command--Provision New Storage • 1052
- cadpmspm resize Command--Resize Existing Storage • 1054

caismutility Command -- Add Functionalities to Policy Configuration UI • 819

cat, type Commands--Display Text Files (Cmdlets) • 56

cd, chdir Commands--Change Directory (Cmdlet, Funclet) • 57

Collection Engine Commands • 738

Command Line Scripting • 383

Command Line Utilities • 1057

Command-Line Instructions (CLIs) • 383

Commands, Functions, and Classes According to Categories • 46

Common Information Model (CIM) Objects • 24

Contact CA Technologies • 4

Conventions • 18

copy, cp Commands--Copy Files and Directories (Funclets) • 58

curDir--Retrieves the Current Directory (Function) • 112

D

DBG_PROMPT Command--Break Script Execution for Inspection (Cmdlet) • 60

DBG_PROMPT_LINE Command--Break Script Execution for Inspection With Line Numbers (Cmdlet) • 61

del, rm Commands--Delete Files (Funclets) • 62

Deployment and Policy Configuration AutoShell Commands • 347

dir, ls Commands--Get File and Directory Information (Funclets) • 64

Discovery Commands • 733

dpmccm addacmserver Command--Add a Server Instance • 687

dpmccm addcompliancejob Command--Create a Compliance Job • 689

dpmccm addremediationjob Command--Add a CA Configuration Automation Remediation Job • 690

dpmccm compare Command--Compare Systems • 691

dpmccm createsnapshot Command--Create a Snapshot • 692

dpmccm deleteacmserver Command--Delete a CCA Server • 693

dpmccm deletecompliancejob Command--Delete a Compliance Job • 694

dpmccm deletesnapshot Command--Delete a Snapshot • 695

dpmccm detect Command--Detect Server Configuration Changes • 696

dpmccm getaccessprofiles Command--Get Access Profiles • 697

dpmccm getacmagentinfo Command--Get Agent Information • 698

dpmccm getallcompliancejobs Command--Get All Compliance Jobs from CCA Server • 699

dpmccm getallcomponents Command--Get Server Components • 700

dpmccm getallremediationjobs Command--Get All CA Configuration Automation Remediation Jobs • 701

dpmccm getallremediationprofiles Command--Get All CA Configuration Automation Remediation Profiles • 702

dpmccm getallrulegroups Command--Get All Compliance Rule Groups from CCA Server • 703

dpmccm getallsnapshot Command--Get all Snapshots on a Server • 704

dpmccm getsources Command--Get Servers • 705

dpmccm getallsystems Command--Get Server Names from CA Configuration Automation Server • 706

dpmccm getcompliancejobbyname Command--Get a Compliance Job By Job Name • 707

dpmccm getcompliancejobhistory Command--Get Compliance Job History for a Specific Job • 707

dpmccm getcurrentactivity Command--Get Server Activity • 708

dpmccm getmanagementprofiles Command--Get Management Profiles • 709

dpmccm getremediationjobhistory Command--Get CA Configuration Automation Remediation Job History • 710

dpmccm getremediationjobinfo Command--Get CA Configuration Automation Remediation Job Details • 711

dpmccm getremediationprofile Command--Get CA Configuration Automation Remediation Profile • 712

dpmccm getsystemaccessprofile Command--Get Access Profile • 713

dpmccm getsystemdefaultsource Command--Get Assigned Source • 714

dpmccm getsystemmanagementprofile Command--Get System Management Profile • 715

dpmccm redoremediationjob Command--Redo a CA Configuration Automation Remediation Job • 716

dpmccm removesystemdefaultsource Command--Remove Assigned Sources • 716

dpmccm runacmdiscovery Command--Run Discovery • 717

dpmccm runcompliancejob Command--Run a Compliance Job • 718

dpmccm runremediationjob Command--Run a CA Configuration Automation Remediation Job • 719

dpmccm runsystemmanagementprofile Command--Run a Management Profile • 720

dpmccm setacmdefaultmanagementprofile Command--Set the Default Management Profile • 722

dpmccm setdefaultacmserver Command—Set the Default CA Configuration Automation Server • 723

dpmccm setsnapshottype Command--Set Snapshot Type • 724

dpmccm setsystemaccessprofile Command--Set Access Profile • 725

dpmccm setsystemdefaultsource Command--Add a Server Source • 726

dpmccm setsystemmanagementprofile Command--Set System Management Profile • 727

dpmccm syncacmserver Command—Synchronize a CA Configuration Automation Server • 728

dpmccm testsystemagent Command--Ping Agent • 729

dpmccm undoremediationjob Command--Undo a CA Configuration Automation Remediation Job • 730

dpmccm updatecompliancejob Command--Update a Compliance Job • 731

dpmccm updateremediationjob Command--Update a CA Configuration Automation Remediation Job • 732

dpmccm validateacminfo Command--Validate CCA Server Credentials • 733

dpmcda cancelnetworkscan Command--Cancel a Network Discovery • 734

dpmcda deletenetwork Command--Delete a Network • 734

dpmcda discoverhost Command--Discover a Host • 735

dpmcda discovernetwork Command--Discover a Network • 737

dpmce ceconfig -configuremetric Command--Configure Metric • 738

dpmce ceconfig -configuremetricfilter Command--Configure Metric Filter • 741

dpmce ceconfig -disablemetric Command--Disable a Metric • 743

dpmce ceconfig -disablemetricforoverall Command--Disable and Exclude a Metric From Overall Calculation • 744

dpmce ceconfig -enablemetric Command--Enable a Metric • 745

dpmce ceconfig -enablemetricforoverall Command--Enable and Include a Metric for Overall Calculation • 746

dpmce ceconfig -getglobal Command--Retrieve Data Center Intervals • 747

dpmce ceconfig -getmetriclist Command--Retrieve Metric List • 749

dpmce ceconfig -getnode Command--Retrieve Host Collection Intervals • 748

dpmce ceconfig -refreshmetriclist Command--Retry Metric Collection • 750

dpmce ceconfig -setalias Command--Set Alias • 751

dpmce ceconfig -setglobal Command--Set Data Center Intervals • 753

dpmce ceconfig -setnode Command--Set Intervals for a Host • 755

dpmce ceconfig -updatemetric Command--Update a Metric • 756

dpmcmd run Command--Run Scripts • 686

dpmec2 createImage--Create AMI Image • 384

dpmec2 ec2jobcheck Command--Get Amazon EC2 Job Status • 385

dpmec2 getimages Command--Get Amazon EC2 Images • 387

dpmec2 getinstances Command--Get Amazon EC2 Instances • 388

dpmec2 getlist Command--Get Amazon EC2 List • 389

dpmec2 imgjobcheck Command--Get Amazon EC2 Imaging Job Status • 390

dpmec2 manage Command--Manage Amazon Elastic Compute Cloud Instances • 392

dpmec2 run Command--Create Amazon EC2 Instance • 393

dpmeventmanager create_event Command--Create an Event • 758

dpmeventmanager get_events Command--Retrieve Events • 759

dpmhd addcomment Command--Update a Ticket with a Comment • 762

dpmhd close Command--Close a Help Desk Ticket • 763

dpmhd getattribute Command--Get an Attribute Value of a Ticket • 764

dpmhd getstatus Command--Get the Status of a Ticket • 765

dpmhd open Command--Open a Help Desk Ticket • 765

dpmhd setattribute Command--Set an Attribute of a Ticket • 767

dpmhd setstatus Command--Set the Status of a Ticket • 768

dpmhv-addVMNic (Cmdlet) Command--Add a Network Interface Controller to a VM • 200

dpmhv-addVMSCSIController (Cmdlet) Command--Add SCSI Controllers to a VM • 202

dpmhv-changeVMState (Cmdlet) Command--Change the State of a VM • 202

dpmhv-createTemplateFromVM (Funclet) Command--Create a Template from a VM • 204

dpmhv-createVirtDisk (Funclet) Command--Create a Virtual Disk • 205

dpmhv-createVirtFloppy (Cmdlet) Command--Create a Virtual Floppy • 206

dpmhv-createVM (Funclet) Command--Create a VM • 207

dpmhv-createVMFromTemplate (Funclet) Command--Create a VM Using a Template • 211

dpmhv-createVMFromTemplateEx (Funclet) Command--Create a VM Using a Template • 217

dpmhv-destroyTemplate (Cmdlet) Command--Delete a Template • 225

dpmhv-destroyVM (Cmdlet) Command--Delete VM • 226

dpmhv-expandVirtDisk (Funclet) Command--Expand Virtual Disk • 227

dpmhv-exportVM (Funclet) Command--Export a VM • 228

dpmhv-exportVMEx (Funclet) Command--Export a VM • 229

dpmhv-getHostSwitches (Funclet) Command--Show Host Switches on a Hyper-V Host • 231

dpmhv-getJobInfo (Funclet) Command--Show Job Information • 234

dpmhv-getJobStatus (Funclet) Command--Show Job Status • 239

dpmhv-getPhysDisks (Funclet) Command--Show all Physical Disks • 239

dpmhv-getVersion (Funclet) Command--Show the Web Service Version • 243

dpmhv-getVMInfo (Funclet) Command--Show all VMs • 244

dpmhv-getVMProperties (Funclet) Command--Show VM Properties • 248

dpmhv-getVMState (Funclet) Command--Show the VM State • 249

dpmhv-importVM (Funclet) Command--Import a VM • 251

dpmhv-importVMEx (Funclet) Command--Copy and Import VM • 252

dpmhv-setSysprepProperties (Cmdlet) Command--Create a Virtual Disk • 254

dpmhv-setVMClustered (Cmdlet) Command--Set a VM in a Cluster • 259

dpmhv-setVMDisk (Funclet) Command--Create a Drive or Disk on a Drive Controller • 260

dpmhv-setVMProperties (Cmdlet) Command--Set VM Properties • 262

dpmhv-SetVMVirtFloppy (Cmdlet) Command--Connect a Virtual Floppy Image to the VM Floppy Drive • 265

dpmhv-ShowClusterSharedVolumes (Funclet) Command--Show all Cluster Shared Volumes • 266

dpmhv-ShowDirectories (Funclet) Command--Show all Directories • 266

dpmhv-ShowHosts (Funclet) Command--Show all Hyper-V Servers • 269

dpmhv-ShowLogicalDisks (Funclet) Command--Show all Logical Disks • 276

dpmhv-ShowSCVMMHardwareProfiles (Funclet) Command--Show all SCVMM Hardware Profiles • 279

dpmhv-ShowSCVMMOSProfiles (Funclet) Command--Show all Guest OS Profiles • 284

dpmhv-ShowTemplates (Funclet) Command--Show Templates on a Hyper-V Host • 288

dpmhv-ShowVMDisks (Funclet) Command--Show all Virtual Disks • 290

dpmhv-showVMFloppy (Funclet) Command--Show the Properties of Floppy Drive • 292

dpmhv-showVMNICs (Funclet) Command--Show all Network Interface Cards • 294

dpmhv-showVMSCSI (Funclet) Command--Show all SCSI Controllers • 296

dpmhyperv AddVMNic Command--Add a Network Adapter Controller to a VM • 436

dpmhyperv AddVMSCSIController Command--Create a SCSI Controller for a VM • 438

dpmhyperv AgentGet Command--Return an Agent Value • 439

dpmhyperv AgentSet Command--Set an Agent Value • 440

dpmhyperv ChangeVMState Command--Change the State of a VM • 442

dpmhyperv CreateTemplateFromVM Command--Create a Template from a VM • 444

dpmhyperv CreateVirtDisk Command--Create a Virtual Disk Image • 445

dpmhyperv CreateVirtFloppy Command--Create a Virtual Floppy Disk Image • 447

dpmhyperv CreateVM Command--Create a VM • 448

dpmhyperv CreateVMFromTemplate Command--Create a VM from a SCVMM Server Template • 452

dpmhyperv CreateVMFromTemplate Command--Create a VM from a Template • 456

dpmhyperv DeleteTemplate Command--Deletes a Template • 507

dpmhyperv DeleteVM Command--Delete a VM • 463

dpmhyperv ExpandVirtDisk Command--Expand the Size of a Virtual Disk • 464

dpmhyperv ExportVM Command--Export a VM • 466

dpmhyperv ExportVMEx Command--Export a VM • 467

dpmhyperv GetHostSwitches Command--Get Switches of a Host • 469

dpmhyperv GetJobErrorInfo Command--Get the Error Information of a Job • 470

dpmhyperv GetJobInfo Command--Get Information of a Job • 471

dpmhyperv GetJobState Command--Get the State of a Job • 473

dpmhyperv GetServerInfo Command--Get all Managed Servers • 474

dpmhyperv GetVersion Command--Get the PMM Version • 475

dpmhyperv GetVMProperties Command--Get VM Properties • 476

dpmhyperv GetVMState Command--Get a VM State • 477

dpmhyperv ImportVM Command--Import a VM • 478

dpmhyperv ImportVMEx Command--Copy and Import a VM • 480

dpmhyperv SetClustered Command--Add or Remove a VM from a Cluster • 481

dpmhyperv SetSysPrepProperties Command--Sets Sysprep Properties • 483

dpmhyperv SetVMDisk Command--Assign a Drive or Disk to a Drive Controller • 486

dpmhyperv SetVMProperties Command--Set Properties of a VM • 488

dpmhyperv SetVMVirtFloppy Command--Assign a Virtual Floppy Image to a VM Floppy Drive • 491

dpmhyperv ShowClusterSharedVolumes Command--Show all Cluster Shared Volumes • 492

dpmhyperv ShowDirectories Command--Show Directory Properties on a Hyper-V Server • 493

dpmhyperv ShowLogicalDisks Command--Show all Logical Disks • 494

dpmhyperv ShowPhysDisks Command--Show Physical Disks of a Host • 495

dpmhyperv ShowSCVMMGuestOSProfiles Command--Show Guest Operating System Profiles • 497

dpmhyperv ShowSCVMMHardwareProfiles Command--Show all SCVMM Hardware Profiles • 496

dpmhyperv ShowTemplates Command--Show all VM Templates • 498

dpmhyperv ShowVirtFloppy Command--Show the List of Virtual Floppy Drives • 499

dpmhyperv ShowVMDisks Command--Show all Virtual Disks • 501

dpmhyperv ShowVMNics Command--Show all Network Interface Cards • 502

dpmhyperv ShowVMs (Funclet) Command--Get all VMs • 299

dpmhyperv ShowVMs Command--Show all VMs • 503

dpmhyperv ShowVMSCSIControllers Command--Show all SCSI Controllers • 504

dpmhyperv ValidateAgentSNMPAccess Command--Validate SNMP Access to an Agent • 505

dpmhyperv ValidateServerAccess Command--Check Server Access • 506

dpmimg image Command--Deploy a Solaris Image • 770

dpmimg imgjobcheck Command--Get Solaris Imaging Job Status • 774

dpmimg refresh_images Command--Refresh Boot and Install Images from Solaris JumpStart Servers • 776

dpmitpamadapter checkstatus Command--Check Status • 822

dpmitpamadapter getform Command--Get Form • 823

dpmitpamadapter listforms Command--List Forms • 824

dpmitpamadapter startprocess Command--Start Process • 825

dpmkpdb Command--Import or Export Performance Data • 1058

dpmlpar cycle Command--Cycle a Logical Partition • 509

dpmlpar delete Command--Delete a Logical Partition • 513

dpmlpar getresources Command--Get LPAR Resources • 515

dpmlpar image Command--Create an IBM AIX LPAR • 517

dpmlpar imgjobcheck Command--Retrieve Status of IBM AIX Imaging Job • 529

dpmlpar list Command--List HMC Resources • 530

dpmlpar setresources Command--Add Memory Resources • 533

dpmlpar setresources Command--Add Processor Resources • 535

dpmlpar setresources Command--Subtract Memory Resources • 538

dpmlpar setresources Command--Subtract Processor Resources • 540

dpmlpar setresources Command--Update Memory Resources • 543

dpmlpar setresources Command--Update Processor Resources • 544

dpmlpar-cycle Command--Cycle a Logical Partition (Funclet) • 300

dpmlpar-delete Command--Delete a Logical Partition (Funclet) • 304

dpmlpar-getresources Command--Get LPAR Resources (Funclet) • 305

dpmlpar-image Command--Create an IBM AIX LPAR (Funclet) • 307

dpmlpar-imgjobcheck Command--Retrieve Status of IBM AIX Imaging Job (Funclet) • 316

dpmlpar-list Command--List HMC Resources (Funclet) • 317

dpmlpar-setresources Command--Add Memory Resources (Funclet) • 319

dpmlpar-setresources Command--Add Processor Resources (Funclet) • 321

dpmlpar-setresources Command--Subtract Memory Resources (Funclet) • 323

dpmlpar-setresources Command--Subtract Processor Resources (Funclet) • 325

dpmmscs bringgrouponline Command--Bring a Resource Group Online • 559

dpmmscs bringresourceonline Command--Bring a Resource Online • 560

dpmmscs getclusterinfo Command--Get Information of a Cluster • 547

dpmmscs getgroupresourcesinfo Command--Get Resource Information of a Cluster group • 548

dpmmscs getnetworkinterfacesinfo Command--Get Network Interface Information • 550

dpmmscs getnetworksinfo Command--Get Network Information • 551

dpmmscs getnodesinfo Command--Get Cluster Nodes Information • 552

dpmmscs getresourcegroupsinfo Command--Get Cluster Resource Groups Information • 553

dpmmscs getresourcesdependenciesinfo Command--Get Dependency Information of Resources • 554

dpmmscs getresourcetypesinfo Command--Get Cluster Resource Types Information • 555

dpmmscs getservicesinfo Command--Get Cluster Services Information • 556

dpmmscs movegroup Command--Move a Resource Group to another Node • 557

dpmmscs pauseservice Command--Pause a Cluster Service • 558

dpmmscs resumeservice Command--Resume a Cluster Service • 562

dpmmscs startservice Command--Start a Cluster Service • 563

dpmmscs stopservice Command--Stop a Cluster Service • 564

dpmmscs takegroupoffline Command--Take a Resource Group Offline • 565

dpmmscs takeresourceoffline Command--Take a Resource Offline • 566

dpmnim image Command—Deploy an IBM AIX Image to an Existing LPAR Using a Resource Group • 790

dpmnim image Command—Deploy an IBM AIX Image to an Existing LPAR Using Individual Resources • 785

dpmnim image Command—Deploy an IBM AIX Image Using a Resource Group • 781

dpmnim image Command—Deploy an IBM AIX Image Using Individual Resources • 777

dpmnim imgjobcheck Command—Retrieve NIM IBM AIX Imaging Job Status • 793

dpmovf import Command--Import an OVF Package • 684

dmpolicy createdefaultactions Command--Create Default Actions • 803

dmpolicy exportaction Command--Export Actions to a File • 804

dmpolicy exportrule Command--Export Rules to a File • 805

dmpolicy findbestmachine Command--Find the Best Server • 806

dmpolicy findcandidatemachine Command--Find the Candidate Servers • 807

dmpolicy getrulerecount Command--Retrieve rule execution count • 809

dmpolicy importaction Command--Import Actions from a File • 810

dmpolicy importrule Command--Import Rules from a File • 811

dmpolicy resetrulerecount Command--Reset Retries to Run a Rule • 812

dmpolicy runaction Command--Run an Action • 814

dmpolicy setrulereclimit Command--Set Limits for a Rule • 816

dpmresourcemgr addbaremetal Command--Add a Physical Computer • 847

dpmresourcemgr addcomputertonetwork Command--Add Server to Network • 848

dpmresourcemgr addipaddresspool Command--Add IP Address Pools • 849

dpmresourcemgr addmachinestosome Command--Add Servers to a Service • 851

dpmresourcemgr addtocontainer Command--Add Content to a Container • 852

dpmresourcemgr createcontainer Command--Create a Container • 853

dpmresourcemgr createdefaultservice Command--Create Default Service • 854

dpmresourcemgr createipnetwork Command--Create Network • 855

dpmresourcemgr createresourcegroup Command--Create Resource Group • 857

dpmresourcemgr createresourcetemplate Command--Create Server Group Template • 858

dpmresourcemgr createservice Command--Create a Service • 859

dpmresourcemgr deletebaremetal Command--Delete a Physical Computer • 861

dpmresourcemgr deletecontainer Command--Delete a Container • 862

dpmresourcemgr deleteipnetworks Command--Delete a Network • 863

dpmresourcemgr deletesystem Command--Delete a System • 864

dpmresourcemgr getdiscoverednetworks Command--Get Discovered Networks • 865

dpmresourcemgr getipaddresspools Command--List IP Address Pools • 866

dpmresourcemgr getipnetworks Command--List Networks • 867

dpmresourcemgr getserverlist Command--Get a List of Servers • 868

dpmresourcemgr getservicelist Command--Get List of Services • 869

dpmresourcemgr getsubnetdefaults Command--List Subnet Defaults • 870

dpmresourcemgr removecomputerfromipnetwork Command--Remove Server From Network • 871

dpmresourcemgr removefromcontainer Command--Remove Content from a Container • 872

dpmresourcemgr removeipaddresspool Command--Remove IP Address Pool • 873

dpmresourcemgr removemachinesfromservice Command--Remove Servers From a Service • 874

dpmresourcemgr removeresourcegroup Command--Remove Resource Group • 875

dpmresourcemgr removeservice Command--Remove a Service • 876

dpmresourcemgr removesourcetemplate Command--Remove Resource Template • 877

dpmresourcemgr retrieveresourcegroup Command--Retrieve Resource Group • 878

dpmresourcemgr retrieveresourcetemplate Command--Retrieve Resource Template • 879

dpmresourcemgr setcontainername Command--Set Container Name • 880

dpmresourcemgr setcontainerservice Command--Set Container to a Service • 881

dpmresourcemgr setmanagementip Command--Set the Management IP for a Host • 883

dpmresourcemgr setmanagementstatus Command--Set the Management Status for a Host • 884

dpmresourcemgr setserviceproperties
Command--Set Service Properties • 885

dpmresourcemgr updateipaddresspool
Command--Update IP Address Pool • 893

dpmresourcemgr updateipnetwork
Command--Update a Network • 890

dpmresourcemgr updateresourcegroup
Command--Update Resource Group • 886

dpmresourcemgr updateresourcetemplate
Command--Update Resource Template • 887

dpmresourcemgr updateservice Command--Update
a Service • 889

dpmresourcemgr updatesubnetdefaults
Command--Update Network Defaults • 891

dpmmr cancel Command--Cancel a Reservation •
835

dpmmr checkreservation and updatereservation
Commands • 836

dpmmr createfromtemplate Command--Create
Reservation • 837

dpmmr createsoftwaregroup Command • 839

dpmmr extend Command--Extend a Reservation •
841

dpmmr importvirtualmachines Command • 843

dpmmr listsoftwaregroups Command • 844

dpmmr listtemplates Command--Display List
Reservation Templates • 846

dpmsd cancelimage Command--Cancel a Pending
Image • 1017

dpmsd deliver Command--Deliver a Package • 1018

dpmsd deploypkggrp Command--Deliver a Package
Group • 1022

dpmsd deployprocgrp Command--Deliver a
Procedure Group • 1025

dpmsd image Command--Deploy an Image • 1028

dpmsd imgjobcheck Command--Get OS Imaging Job
Status • 1032

dpmsd installSDAgent--Install Software Delivery
Agent • 1033

dpmsd list Command--List Software Packages • 1036

dpmsd listsd Command--List Installed SD Adapters •
1037

dpmsd pkgjobcheck Command--Get Package
Deployment Job Status • 1038

dpmsd setmanagedstatus Command--Set
Management Status of Package • 1040

dpmucs associateserviceprofile--Associate Service
Profile With a Blade • 397

dpmucs cloneserviceprofile--Clone a Service Profile •
398

dpmucs createmacpool Command--Create MAC
Address Pool • 399

dpmucs createorg Command--Create Organization •
401

dpmucs createserverpool Command--Create Server
Pool • 402

dpmucs createserviceprofile Command--Create a
Service Profile • 404

dpmucs createtemplatefromserviceprofile
Command--Create a Template from a Service
Profile • 405

dpmucs createuuidpool Command--Create UUID
Pool • 407

dpmucs createwwpool Command--Create WWNN
Pool • 408

dpmucs deletemacpool Command--Delete MAC
Address Pool • 410

dpmucs deleteorg Command--Delete Organization •
411

dpmucs deleteserverpool Command--Delete Server
Pool • 413

dpmucs deleteserviceprofile--Delete a Service Profile
• 414

dpmucs deleteuuidpool Command--Delete UUID
Pool • 415

dpmucs deletewwpool Command--Delete WWNN
Pool • 416

dpmucs deployimage Command--Deploy a Cisco UCS
Image • 418

dpmucs disassociateserviceprofile
Command--Disassociate Service Profile • 419

dpmucs failoverserviceprofile Command--Associate a
Service Profile With a Blade for Failover • 421

dpmucs getblades Command--Get Blade Information
• 422

dpmucs getchassis Command--Get Chassis
Information • 423

dpmucs getpolicy Command--Gets Policy from a UCS
Manager • 425

dpmucs getserviceprofilestatus Command--Get
Service Profile Status • 426

dpmucs getswitches Command--Get Switch
Information • 427

dpmucs gettopsystem Command--Get System
Information • 428

dpmucs listserviceprofiles Command--List Service
Profiles • 429

dpmucs powercycle Command--Power Cycle Blades • 431

dpmucs renamepool Command--Rename Pool • 433

dpmucs serviceprofileupdate Command--Execute Service Profile Operations • 434

dpmutil -amq Command--Configure the ActiveMQ Message Broker • 1060

dpmutil -ccm Command--Configure CA Configuration Automation • 1061

dpmutil -dhcp_unix Command--Configure DHCP for UNIX • 1062

dpmutil ec2-owner Command--Configure EC2 Owner • 1063

dpmutil ec2-private-keypair Command--Configure Private Key Pair for EC2 • 1065

dpmutil ecs Command--Configure Windows Scheduler • 1066

dpmutil -emc-smi Command--Configure EMC SMI-S Provider Server • 1066

dpmutil -hd Command--Configure Help Desk • 1067

dpmutil -hmc Command--Configure HMC or IVM Server • 1068

dpmutil -hyperv Command--Configure Hyper-V Server • 1069

dpmutil -itpam-cfg-eem Command--Configure CA Process Automation/CA EEM • 1070

dpmutil -itpamserver Command--Configure CA Process Automation • 1071

dpmutil -js Command--Configure JumpStart • 1072

dpmutil mgmtadb Command--Configure Management Database • 1072

dpmutil netapp Command--Configure NetApp DataFabric Manager • 1073

dpmutil -nim Command--Configure NIM Server • 1074

dpmutil -perfadb Command--Configure the Performance Database • 1074

dpmutil -proxy Command--Configure Proxy Server • 1075

dpmutil remoteucsaim Command--Configure Remote Cisco Unified Computing System AIM • 1076

dpmutil remotevcaim Command--Configure Remote VC AIM • 1077

dpmutil -sc Command--Configure the Service Controller • 1077

dpmutil -scvmmserver Command--Configure SCVMM Server • 1078

dpmutil -sdscale Command--Configure the Software Delivery Scalability Server • 1079

dpmutil -sdserver Command--Configure Software Delivery Server • 1080

dpmutil -snmp Command--Configure SNMP • 1081

dpmutil -superuser Command--Configure SuperUser Credentials • 1083

dpmutil Syntax • 1059

dpmutil -systemcredential Command--Configure External System Credentials • 1084

dpmutil -sysuser Command--Configure Local Credentials Used at Service Startup • 1085

dpmutil ucs Command--Configure Cisco Unified Computing System Server • 1086

dpmutil upm Command--Configure the CA Patch Manager DB • 1087

dpmutil vcdefault Command--Configure Default VMware vCenter Administrator User • 1087

dpmutil vcserver Command--Configure VMware vCenter • 1088

dpmutil -vioserver Command--Configure VIO Server • 1089

dpmvc addesxhost Command--Add an ESX Host • 597

dpmvc addvmvdisk Command--Add Virtual Disk • 599

dpmvc addvmvnic Command--Add Virtual NIC • 601

dpmvc clone Command--Clone a VM • 604

dpmvc createdatastore Command--Create a Datastore • 612

dpmvc cycle Command--Cycle a VM • 614

dpmvc datastore Command--Get Data Store Properties • 618

dpmvc delete Command--Destroy a VM • 620

dpmvc distributedswitch Command--Manage Virtual Distributed Switches • 623

dpmvc entermaintenancemode Command--Set to the Maintenance Mode • 628

dpmvc exitmaintenancemode Command--Exit the Maintenance Mode • 630

dpmvc faulttolerance Command--Specify Fault Tolerant Operations • 632

dpmvc getavailablescsidisks Command--Get Available SCSI Disks • 634

dpmvc gethosthba Command--Get Host Bus Adapters • 635

dpmvc getresources Command--Get VM Resources • 636

dpmvc getshares Command--Get VM Shares • 639

dpmvc getsnapshots Command--Get VM Snapshots • 641

dpmvc imgjobcheck Command--Get Clone Job Status • 643

dpmvc migrate Command--Migrate a VM • 646

dpmvc removedatastore Command--Remove a Datastore • 649

dpmvc removevmvdisk Command--Remove Virtual Disk • 650

dpmvc removevmvnic Command--Remove Virtual NIC • 652

dpmvc setcpu Command-- Set CPUs for a VM • 654

dpmvc setmemory Command-- Set memory size for a VM • 656

dpmvc setresources Command--Set VM Resources • 658

dpmvc setshares Command--Set VM Shares • 661

dpmvc snapshot Command--Manage Snapshots • 665

dpmvc templatetovm Command--Convert Template to VM • 670

dpmvc vapp Command--Manage vApp • 194, 673

dpmvc virtualswitch Command--Manage Virtual Switches • 677

dpmvc vmtotemplate Command--Convert VM to Template • 681

dpmvc addesxhost Command--Add an ESX Host (Funclet) • 160

dpmvc addvmvdisk Command--Add Virtual Disk (Funclet) • 162

dpmvc addvmvnic Command--Add Virtual NIC (Funclet) • 163

dpmvc-clone Command--Clone a VM (Funclet) • 164

dpmvc-createdatastore Command--Create a Datastore (Funclet) • 171

dpmvc-cycle Command--Cycle a VM (Funclet) • 172

dpmvc-datastore Command--Get Data Store Properties (Funclet) • 173

dpmvc-delete Command--Destroy a VM (Funclet) • 174

dpmvc-distributedswitch Command--Manage Virtual Distributed Switches (Funclet) • 175

dpmvc-entermaintenancemode Command--Set to the Maintenance Mode (Funclet) • 178

dpmvc-exitmaintenancemode Command--Exit the Maintenance Mode (Funclet) • 179

dpmvc-faulttolerance Command--Specify Fault Tolerant Operations (Funclet) • 180

dpmvc-getavailablescsidisks Command--Get Available SCSI Disks (Funclet) • 181

dpmvc-gethosthba Command--Get Host Bus Adapters (Funclet) • 182

dpmvc-getresources Command--Get VM Resources (Funclet) • 182

dpmvc-getshares Command--Get VM Shares (Funclet) • 183

dpmvc-getsnapshots Command--List Snapshots of a VM (Funclet) • 184

dpmvc-imgjobcheck Command--Get Clone Job Status (Funclet) • 185

dpmvc-migrate Command--Migrate a VM (Funclet) • 185

dpmvc-removedatastore Command--Remove a Datastore (Funclet) • 186

dpmvc-removevmvdisk Command--Remove Virtual Disk (Funclet) • 187

dpmvc-removevmvnic Command--Remove Virtual NIC (Funclet) • 188

dpmvc-setresources Command--Set VM Resources (Funclet) • 188

dpmvc-setshares Command--Set VM Shares (Funclet) • 189

dpmvc-snapshot Command--Manage Snapshots (Funclet) • 191

dpmvc-templatetovm Command--Convert Template to VM (Funclet) • 192

dpmvc-virtualswitch Command--Manage Virtual Switches (Funclet) • 197

dpmvc-vmtotemplate Command--Convert VM to Template (Funclet) • 199

dpmvc-wait Command--Time to Wait for a Job (Funclet) • 199

dpmzone associateproject Command--Associate a Task with a Project • 567

dpmzone clonezone Command--Clone a Zone • 568

dpmzone createandinstallzone Command--Create and Install a Zone • 569

dpmzone createpool Command--Create Resource Pool • 572

dpmzone createproject Command--Create a Project • 574

dpmzone createzone Command--Create a Zone • 575

dpmzone customcommand Command--Run a Custom Command • 578

dpmzone deleteproject Command--Delete a Project • 579

dpmzone deletezone Command--Delete a Zone • 580

dpmzone installzone Command--Install a Zone • 581

dpmzone movezone Command--Move a Zone • 582

dpmzone rebootzone Command--Reboot a Zone • 583

dpmzone renamezone Command--Rename a Zone • 584

dpmzone setpoolscheduler Command--Set the Pool Scheduler • 585

dpmzone setprocessorset Command--Set the Maximum Number of CPUs for a Processor Set • 586

dpmzone showinterfaces Command--Show Network Interfaces • 587

dpmzone showpools Command--Show Resource Pools • 588

dpmzone showprojects Command--List the Properties of a Project • 589

dpmzone showresourceSet Command--Show Resource Sets • 589

dpmzone showtasks Command--List the Tasks of a Project • 590

dpmzone showzoneinterfaces Command--Show Network Interfaces of a Zone • 591

dpmzone showzoneprocess Command--Show Processes of a Zone • 592

dpmzone showzoneprojects Command--Show Projects of a Zone • 592

dpmzone showzones Command--Show Zones • 593

dpmzone startzone Command--Start a Zone • 594

dpmzone stopzone Command--Stop a Zone • 595

dpmzone uninstallzone Command--Uninstall a Zone • 596

dpmzone-associateproject Command--Associate a Task with a Project (Funclet) • 327

dpmzone-clonezone Command--Clone a Zone (Funclet) • 328

dpmzone-createandinstallzone Command--Create and Install a Zone • 329

dpmzone-createpool Command--Create a Resource Pool (Funclet) • 332

dpmzone-createproject Command--Create a Project (Funclet) • 333

dpmzone-createzone Command--Create a Zone (Funclet) • 334

dpmzone-customcommand Command--Run a Custom Command (Funclet) • 336

dpmzone-deleteproject Command--Delete a Project (Funclet) • 336

dpmzone-deletezone Command--Delete a Zone (Funclet) • 337

dpmzone-installzone Command--Install a Zone (Funclet) • 337

dpmzone-movezone Command--Move a Zone (Funclet) • 338

dpmzone-rebootzone Command--Reboot a Zone (Funclet) • 339

dpmzone-renamezone Command--Rename a Zone (Funclet) • 339

dpmzone-setpoolscheduler Command--Set the Pool Scheduler (Funclet) • 340

dpmzone-setprocessorset Command--Set the Maximum Number of CPUs for a Processor Set (Funclet) • 341

dpmzone-showinterfaces Command--Show Network Interfaces (Funclet) • 341

dpmzone-showpools Command--Show Resource Pools (Funclet) • 342

dpmzone-showprojects Command--List the Properties of a Project (Funclet) • 342

dpmzone-showresourceSet Command--Show Resource Sets (Funclet) • 343

dpmzone-showtasks Command--List the Tasks of a Project (Funclet) • 343

dpmzone-showzoneinterfaces Command--Show Network Interfaces of a Zone (Funclet) • 344

dpmzone-showzoneprocess Command--Show Processes of a Zone (Funclet) • 344

dpmzone-showzoneprojects Command--Show Projects of a Zone (Funclet) • 345

dpmzone-showzones Command--Show Zones (Funclet) • 345

dpmzone-startzone Command--Start a Zone (Funclet) • 346

dpmzone-stopzone Command--Stop a Zone (Funclet) • 346

dpmzone-uninstallzone Command--Uninstall a Zone (Funclet) • 347

E

Event Commands • 758

exit, quit Commands--Terminate an AutoShell Session (Cmdlets) • 67

external Command--Declare a Native External Function (Cmdlet) • 67

G

- General AutoShell Commands • 49
- General Shell Commands • 685
- get-autoShellClassInfo Command--Display Methods and Properties for Autoshell Native Classes (Cmdlet) • 68
- gete--Get Environment Variable (Function) • 113
- get-help Command--Get Help Information (Funclet) • 69
- get-remoteResult Command--Get Result from a Remote Target (Funclet) • 70
- get-webServiceInfo Command--Display WSDL Info (Cmdlet) • 71

H

- help Command--Request Help Information (Cmdlet) • 71
- Help Desk Commands • 762

I

- Introduction • 17
- Invoking AutoShell • 28
- ism-applyPolicies Command--Apply Policies to a Group of Hosts (Funclet) • 348
- ism-applyPoliciesAndTemplates Command--Apply Policies and Templates to a Group of Hosts (Funclet) • 351
- ism-applyPoliciesAndTemplatesForService Command--Apply Policies and Templates to all Systems (Funclet) • 354
- ism-applyPoliciesForService Command--Apply Policies to all Systems (Funclet) • 352
- ism-applyPolicyForService Command--Apply a Policy to all Systems (Funclet) • 347
- ism-applyPolicyTemplates Command--Apply Policy Templates to a Group of Hosts (Funclet) • 350
- ism-applyPolicyTemplatesForService Command--Apply Policy Templates to all Systems (Funclet) • 353
- ism-applyPolicyWait Command--Apply a Policy and Wait (Funclet) • 349
- ism-cancelJob Command--Cancel a Job (Funclet) • 355
- ism-copyPolicy Command--Copy a Policy (Funclet) • 356
- ism-copyPolicyTemplate Command--Copy a Policy Template (Funclet) • 376

- ism-createJobTemplate Command--Create a Job Template (Funclet) • 356
- ism-deleteJobTemplate Command--Delete a Job Template (Funclet) • 357
- ism-deletePolicy Command--Delete a Policy (Funclet) • 357
- ism-deletePolicyTemplate Command--Delete a Policy Template (Funclet) • 377
- ism-deployPackageToHost Command--Deploy a Package to a Host (Funclet) • 358
- ism-deployTemplate Command--Deploy Packages from a Template (Funclet) • 359
- ism-deployTemplateForService Command--Deploy Template Packages in a Service (Funclet) • 361
- ism-deployTemplateToHost Command--Deploy Template Packages on a Host (Funclet) • 362
- ism-encryptString Command--Encrypt a String (Funclet) • 363
- ism-listComputerSystems Command--Displays all Computer Systems (Funclet) • 364
- ism-listDeployedSystems Command--List all Hosts With a Package Deployed (Funclet) • 364
- ism-listJobs Command--List all Jobs (Funclet) • 365
- ism-listJobStatus Command--List a Job Status (Funclet) • 365
- ism-listJobTemplates Command--List all Job Templates (Funclet) • 366
- ism-listPackages Command--List all Packages (Funclet) • 366
- ism-listPolicies Command--List all Policies (Funclet) • 366
- ism-listPolicyStatus Command--List a Policy Status (Funclet) • 367
- ism-listPolicyTemplate Command--List all Policy Templates (Funclet) • 381
- ism-listPolicyTemplateStatus Command--List a Policy Template Status (Funclet) • 377
- ism-listServiceSystems Command--List all Computer Systems in Services (Funclet) • 367
- ism-listSysEdgeConfig Command--List the SystemEDGE Configuration (Funclet) • 368
- ism-listSystemPolicies Command--List all Policies (Funclet) • 370
- ism-listSystemPoliciesForService Command--List all Policies in a Service (Funclet) • 368
- ism-listTemplateEulas Command--Display EULAs in a Template (Funclet) • 369
- ism-removePolicyTemplate Command--Remove a Policy Template (Funclet) • 378

ism-renamePolicy Command--Rename a Policy (Funclet) • 369

ism-renamePolicyTemplate Command--Rename a Policy Template (Funclet) • 379

ism-setMaintenanceMode Command--Set the Maintenance Mode of a Host (Funclet) • 370

ism-setMaintenanceModeForService Command--Set the Maintenance Mode in a Service (Funclet) • 371

ism-setSysEdgeLogLevel Command--Set the SystemEDGE Log Level (Funclet) • 372

ism-setSysEdgeLogLevelForService Command--Set the SystemEDGE Log Level in a Service (Funclet) • 372

ism-updateAllExceptionSystems Command--Apply the Latest Policy (Funclet) • 373

ism-updateAllOldPolicies Command--Update all Old Policies (Funclet) • 373

ism-updateAllPolicies Command--Update all Policies (Funclet) • 374

ism-updateAllUpToDateSystems Command--Update all Up-To-Date Systems (Funclet) • 374

ism-updateExceptionSystems Command--Apply the Latest Policy (Funclet) • 374

ism-updateExceptionSystemsTemplate Command--Apply the Latest Policy Template (Funclet) • 379

ism-updateOldSystems Command--Update an old Policy (Funclet) • 375

ism-updateOldSystemsTemplate Command--Update all Old System Templates (Funclet) • 380

ism-updatePolicies Command--Update Configuration of a Policy (Funclet) • 375

ism-updateSystemsTemplate Command--Apply a System Template (Funclet) • 380

ism-updateUpToDateSystems Command--Reapply a Policy (Funclet) • 376

ism-updateUpToDateSystemsTemplate Command--Reapply a System Template (Funclet) • 381

L

Log File Settings • 1056

Log Files • 1055

M

Manager Shell and Client Shell • 23

memoRead--Read a Text File Into a String (Function) • 114

memoWrit--Write a String to a File (Function) • 115

mkdir Command--Create a Directory (Cmdlet) • 72

mv, ren Commands--Move Files and Directories (Funclets) • 74

N

Network Installation Management Commands • 777

new-comObject Command--Instantiate a COM Object (Funclet) • 76

new-SSHSession Command--Create a new Secure SSH Session (Funclet) • 78

new-webService Command--Create a Web Service Object (Funclet) • 79

O

objdump Command--Display Enumerable Properties of an Object (Cmdlet) • 80

Object Model Utility Commands • 795

OSRedirect Class • 136

OSRedirect.clear Method • 137

OSRedirect.errorOccurred Method • 138

OSRedirect.errout Method • 139

OSRedirect.hasCompleted Method • 140

OSRedirect.onCompleted Method • 141

OSRedirect.onError Method • 141

OSRedirect.OSRedirect Constructor • 137

OSRedirect.output Method • 142

OSRedirect.receivedErrOutput Method • 143

OSRedirect.receivedOutput Method • 143

OSRedirect.receivedResult Method • 144

OSRedirect.result Method • 145

P

platform--Query Operating System Type (Function) • 115

Policy Commands • 803

Policy Configuration • 818

push-client Command--Install the Autossh Client on a Remote Windows System (Funclet) • 81

push-winRemote Command--Copy and Execute on Multiple Remote Windows Systems (Funclet) • 83

pute--Set an Environment Variable (Function) • 116

PW_GET Command--Input Hidden Data (Funclet) • 85

pwd Command--Print Name of Working Directory (Cmdlet) • 86

Q

- qout--Write Output Followed by Linefeed to stdout (Function) • 117
- qqout--Write Output to stdout (Function) • 118
- query-service Command--Query the Status of a Windows Service (Funclet) • 87

R

- Reading Data from the Keyboard (stdin) • 35
- regCreateKey--Create a Registry Key (Function) • 119
- regCreateSubkeys--Create Subkeys From an Array (Function) • 120
- regDeleteKey--Delete a Registry Key or a Key Hierarchy (Function) • 121
- regDeleteVal--Delete a Registry Value (Function) • 122
- regGetKeyValues--Get Registry Key Value Information (Function) • 123
- regGetSubKey--Retrieve Sub Keys of a Registry Key (Function) • 125
- regGetVal--Get Registry Value (Function) • 127
- regIsKey--Check the Existence of a Registry Key (Function) • 128
- regIsVal--Check the Existence of a Registry Value (Function) • 129
- regSetKeyValues--Set Registry Key Values From an Array (Function) • 130
- regSetVal--Set Registry Value (Function) • 131
- Related Publications • 17
- rem-client Command--Remove the Autossh Client from a Remote Windows System (Funclet) • 88
- Remote Batch Mode • 32
- Remote Monitoring Commands • 826
- RemoteTarget Class • 146
- RemoteTarget.abort Method • 148
- RemoteTarget.createRemoteContext Method • 149
- RemoteTarget.destroyRemoteContext Method • 150
- RemoteTarget.errorOccurred Method • 151
- RemoteTarget.execute Method • 152
- RemoteTarget.getError Method • 152
- RemoteTarget.getHost Name Method • 154
- RemoteTarget.hasCompleted Method • 155
- RemoteTarget.onError Method • 156
- RemoteTarget.output Method • 156
- RemoteTarget.receivedOutput Method • 157
- RemoteTarget.receivedResult Method • 157
- RemoteTarget.RemoteTarget Constructor • 147
- RemoteTarget.result Method • 158

- RemoteTarget.wasAborted Method • 159
- Reservation Manager Commands • 835
- Resource Manager Commands • 847
- Retrieve Antecedents and Dependents From CIM Objects • 25
- Retrieve CIM Classes • 24
- Retrieve CIM Objects • 26
- rmdir Command--Remove Directory (Funclet) • 90
- rmonwatch add Command--Add Rows to a MIB Table • 826
- rmonwatch del Command--Delete Rows from a MIB Table • 828
- rmonwatch getTable Command--View rmonWbem MIB Tables • 830
- rmonwatch getVersion Command--View Version Information • 832
- run-client Command--Run the Autossh Client on a Remote Windows System (Funclet) • 92
- run-local Command--Execute a Script on the Local System (Funclet) • 93
- Running Scripts • 37
- run-remote Command--Execute a Script on Remote Systems (Funclet) • 94
- run-SSHCommand Command--Run a Remote Command Through SSH (Funclet) • 98
- run-SSHShell Command--Run an Interactive SSH Command Line (Funclet) • 101
- run-winRemote Command--Execute a Command on Multiple Remote Windows Systems (Funclet) • 103

S

- Scope of This Guide • 17
- Service Response Monitor CLI Commands • 894
- set alternate to Command--Set Alternate Output File (Cmdlet) • 105
- set console Command--Suppress Console Output (Cmdlet) • 106
- set result display on/off Command--Turn Automatic Result Display On or Off (Cmdlet) • 107
- setProcExitCode--Set AutoShell Exit Code (Function) • 132
- shellType--Query AutoShell Type (Function) • 133
- Software Delivery Commands • 1016
- Solaris Imaging Commands • 769
- start-java Command--Start Java Virtual Machine (Cmdlet) • 109

start-service Command--Start a Windows Service on a Remote or Local System (Funclet) • 107

stop-service Command--Stop a Windows Service on a Remote or Local System (Funclet) • 108

Storage Provisioning Manager Command • 1042

Stringification • 33

svcwatch add adir Command--Add an Active Directory Test • 895

svcwatch add custom Command--Add a Custom Test • 900

svcwatch add dhcp Command--Add a DHCP Test • 905

svcwatch add dns Command--Add a DNS Test • 910

svcwatch add fileio Command--Add a File IO Test • 915

svcwatch add ftp Command--Add an FTP Test • 920

svcwatch add http | https Command--Add an HTTP or HTTPS Test • 925

svcwatch add imap Command--Add an IMAP Test • 931

svcwatch add ldap Command--Add an LDAP Test • 936

svcwatch add mapi Command--Add a MAPI Test • 941

svcwatch add nis Command--Add a NIS Test • 947

svcwatch add nntp Command--Add an NNTP Test • 952

svcwatch add ping Command--Add a PING Test • 957

svcwatch add pop3 Command--Add a POP3 Test • 962

svcwatch add rtemail Command--Add a Round Trip Email Test • 967

svcwatch add smtp Command--Add an SMTP Test • 973

svcwatch add snmp Command--Add an SNMP Test • 978

svcwatch add sql Command--Add an SQL Test • 984

svcwatch add tcpconnect Command--Add a TCP Connect Test • 990

svcwatch add tftp Command--Add a TFTP Test • 995

svcwatch add vuser Command--Add a Virtual User Test • 1000

svcwatch delete Command--Delete a Test • 1005

svcwatch list Command--View Test Information • 1008

svcwatch setstatus Command--Change the Status of a Test • 1011

svcwatch version Command--View SRM Version Information • 1014

T

typeName--Get the Type Name of an Expression (Function) • 134

U

Use Interactive Mode • 30

Use Remote Interactive Mode • 31

Using the OSRedirect Class • 36

Using the RemoteTarget Class • 41

Using the run-local Command • 38

Using the run-remote Command • 40

V

Valid AutoShell User • 23

Verification Commands • 687

W

wait Command--Wait for a Key Press (Cmdlet) • 110

weak external Command--Declare Native External Function (Cmdlet) • 110

Writing Data to the Console (stdout) • 34