

CA IT Client Manager

Web Services Reference Guide

Release 12.8



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 is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

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 documentation set references to the following CA products:

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA Asset Portfolio Management (CA APM)
- CA Common Services™
- CA Desktop Migration Manager (CA DMM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Process Automation
- CA Business Intelligence
- CA Service Desk Manager
- CA WorldView™
- CleverPath™ Reporter

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: Web Services Tutorial Writing Client Applications 23

Generating Client Stubs.....	24
Writing DSM Web Service Clients	24
Sessions	24
Handling UUIDs	24
Properties Architecture Implementation	25
Properties Architecture Concepts	25
Complex Attribute Types	26
Handling Errors	26
Creating and Maintaining a Connection	27
Retrieving Attributes	29
Finding a DSM Object	30
Creating a Software Package.....	32
Creating a Software Package Volume	34
Creating a Software Package Procedure	35
Sealing a Software Package.....	37
Choosing a Target.....	37
Creating a Software Container	38
Creating a Software Job	39
Sealing and Activating a Job	40
Closing the Session	41
Conclusions	42

Chapter 2: Web Services Reference 43

DSM Web Services End Point for UTF-8 Strings	45
Automation Web Services End Point	45
Enumerations	46
AdhocTargetEvaluation	48
ArchivePath	48
BootServerProperty	48
BootImageProperty.....	49
computerClassID	50
PackageFormat.....	64
SoftwarePackageProperty.....	65
ComputerProperty	66
ComputerStatusInfo.....	67

CompRequiresAuthentication	68
DeploymentStatusCode	68
DiscoveredSoftwareSourceType	70
DiscoveredSoftwareType	70
DistributionContainerOrderType	70
DistributionOrderStatus	71
DistributionStatus	71
DomainType	71
DomainProperty	72
DownloadMethod	74
ElementType	74
GroupFlag	74
GroupInheritance	75
GroupSystem	75
Inheritance Level	76
LinkedObjectTypeRequired	76
operatingSystemType	76
PackageExportType	82
QueryFragment	82
QueryFragment2	82
QueryFragmentType	84
QueryFragmentType2	85
QueryType	85
RACPolicy	86
SecurityAreaLinkage	86
SecurityClassType	86
SecurityGroup	90
SecurityProfilePermission	90
SecurityProfileType	91
SoftwareGroupType	91
SoftwareJobContainerScope	92
SoftwareJobContainerState	92
SoftwareJobRenewState	93
SoftwareJobStartupTime	93
SoftwareJobState	93
SoftwareJobTask	94
SoftwarePackageClass	95
SoftwarePackageGroupScope	95
SoftwarePackageProcedureFiledBy	95
SoftwarePackageProcedureGroupScope	96
SoftwarePackageProcedureRelation	96
SoftwarePackageProcedureType	97

SoftwarePackageProcedureTask	97
SoftwarePackageGroupProperty	98
SoftwarePolicyEvaluationPeriodType	98
SoftwareJobType.....	98
SoftwarePackageFiledBy	98
SoftwarePackageType	99
SoftwarePackagingType	99
SoftwarePolicyEvaluationType.....	100
SoftwarePolicyScope	100
SourceMedium	101
StringValueType	101
SystemGroup.....	103
UnitProperty	104
UnitSoftwareJobState	106
UnitType.....	107
UserAccountProperty.....	107
UserProfileProperty	108
SoftwareJobLinkageType	109
WeekDay	109
PERIODTYPE	109
CO_TRIGGER_UNIT_TYPE.....	110
UnitObjectType	110
AMExtraCmdParam	110
CFComponentCode	110
WS_OSIM_DEPLOY.....	111
WS_OSIM_CONF_TYPE	111
WS_OSIM_IR_STATUS	111
WS_OSIM_CONFIG_STATE	112
Property_Type.....	112
FILTERCONDITION	113
OSIMAGEPROPERTY	114
OSIMTARGETPROPERTY	115
ComputerProperty2	115
ComputerFilterCondition	118
AMAgentStatus	119
SDAgentStatus.....	119
RCAgentStatus.....	120
ProtectionAgentStatus	121
ASPAgentStatus.....	122
AVAgentStatus	122
HIPStatus	123
AgentRestrictions	123

ObjectPreference	124
QueryTargetType	124
QueryFilterCondition	125
HWMODEL-FILTER-CONDITION.....	125
COMPONENTS.....	126
COMPONENT-STATUS	126
ConfigurationParams	126
WS_OSIM_CONFIG_STATE.....	127
Enumerations for Automation Services	128
Methods	134
Authentication and Web Services Management	135
Health Monitoring.....	156
Deployment.....	158
Queries	165
Security	178
Software	210
OSIM.....	448
Software Groups	491
Units and Groups	498
Enterprise Manager Distribution Functionalities	575
Enterprise Manager Domain and Domain Group Functionalities	597
Asset Script Job Methods	610
Sequences	619
ComputerProperties2	626
QueryFilter	629
QueryProperties2.....	629
QueryProperty	630
QueryFolderProperties.....	630
HWModelFilter.....	630
AttributeMask	631
AuthenticationData.....	631
CreateDistributionContainerProperties	631
CreateDomainGroupProperties	632
CreateInstallSoftwareJobOrderPolicyProperties	632
CreateInstallSoftwareJobOrderPolicyProperties3	634
CreateInstallSoftwareJobOrderProperties	636
CreateInstallSoftwareJobOrderProperties3	637
CreateQueryProperties	640
CreateQueryProperties2	641
CreateSecurityAreaProperties	641
CreateSecurityProfileProperties	642
CreateSoftwareGroupProperties	643

CreateSoftwareJobContainerProperties	643
CreateSoftwarePackageProcedureDependencyProperties	644
CreateSoftwarePackageProcedureGroupProperties	644
CreateSoftwarePackageProcedureProperties.....	644
CreateSoftwarePackageProcedureProperties2.....	646
CreateSoftwarePackageProcedureProperties3.....	650
CommonSoftwarePackageProcedureProperties3	651
SoftwarePackageProcedureOperationMask3	654
computerClassID	655
ComputerProperties	669
ComputerPropertiesRequired	670
ComputerPropertyFilter	671
ComputerStatusInfoRequired	673
CreateAgentComponents.....	673
CreateAgentComponents.....	674
CreateCatalogSoftwarePackageGroupProperties	674
CreateComputerProperties.....	675
CreateComputerProperties2.....	678
CreateSoftwareJobContainerProperties2	679
CreateSoftwareJobContainerProperties3	680
CreateSoftwareJobOrderPolicyProperties	682
CreateSoftwareJobOrderPolicyProperties3	683
CreateSoftwareJobOrderProperties.....	685
CreateSoftwareJobOrderProperties3.....	687
CreateSoftwarePackageGroupProperties	689
CreateSoftwarePackageProperties	690
CreateSoftwarePackageVolumeProperties.....	690
DistributionContainerProperties.....	691
DistributionContainerPropertiesRequired	692
DistributionPermissionMask	692
DistributionContainerOrderPropertiesRequired	692
CreateSoftwarePolicyProperties2	693
CreateUnitGroupPropertiesByName	695
CreateUserProfileProperties	696
DateTime.....	697
DeploymentPackageProperties.....	697
DeploymentPackagePropertiesRequired	698
DeploymentInfo	699
DeploymentPackageInfo	700
DiscoveredSoftwareProperties	700
DiscoveredSoftwarePropertiesRequired.....	701
DistributionContainerOrderProperties	701

DistributionContainerOrderProperties3	702
DistributionDomainProperties	702
DistributionDomainPropertiesRequired	703
DomainGroupProperties	704
DomainGroupPropertiesRequired	704
DomainProperties	705
MachineCredentials	706
NoOfDistributionDomains.....	707
ParameterInfo	707
ParameterValue	708
QueryFragment	708
QueryResultProperties.....	708
QueryResultPropertiesRequired	709
ResumeDeliveryDistributionContainerProperties.....	709
ExportSoftwarePkgProperties	709
FetchItemJobOrderProperties	710
FetchItemOrderProperties	710
FileSystemElementPath	710
GetSecurityAreaConfiguration	710
GetSecurityGroupPermissions	711
GetSecurityObjectsPermissions	711
ImportSoftwarePackageProperties	711
ImportSoftwarePackageProperties2	712
ImportSoftwarePackagePropertiesByName	713
QueryFragment2	714
QueryProperties.....	716
QueryProperties3	717
QueryPropertiesRequired	718
QueryPropertiesRequired3	718
ScanPackageInfo	719
ScanParameters	719
SDAgentProperties	721
SecurityAreaProperties	722
SecurityAreaPropertiesRequired	722
SecurityClassPermissions	722
SecurityClassPermissionsForSecurityProfile	723
SecurityClassPermissionProperties	723
SecurityClassProperties.....	723
SecurityClassPropertiesRequired	723
SecurityGroupPermissionsForAllSecurityProfiles.....	724
SecurityGroupProfilePermission	724
SecurityObjectLinkageProperties	724

SecurityObjectProfilePermission.....	724
SecurityObjectSecurityAreaLinkages	725
SecurityObjectOwner	725
SecurityObjectPermissions.....	725
SecurityAreaByName	725
SecurityAreaConfiguration	726
SecurityAreaConfigurationByName	726
SecurityObjectPermissionsForAllSecurityProfiles	726
SecurityObjectsSecurityAreaProperties	726
SecurityProfileProperties	726
SecurityProfilePropertiesRequired	727
SetSecurityProfileProperties	727
SetSecurityGroupPermissions	728
SetSecurityObjectPermissions	728
SecurityProfileSpecialPermissionFlags	728
SetCatalogSoftwarePackageGroupProperties.....	729
SetComputerProperties	730
SetDomainGroupProperties	731
SetSecurityClassProperties.....	731
SetSoftwareJobContainerProperties2	731
SetSoftwarePolicyProperties2.....	732
SetSoftwareGroupProperties	734
SetQueryProperties.....	734
SetQueryProperties3.....	735
SetSoftwarePackageProcedureGroupProperties	735
SetSoftwarePackageProcedureProperties3	736
SoftwarePackageProcedureProperties3	737
SetSoftwareJobContainerProperties.....	739
SetSoftwareJobContainerProperties2.....	739
SetSoftwareJobContainerProperties3.....	740
SoftwareJobContainerPropertiesRequired2	741
SetSoftwareJobProperties.....	742
SetSoftwareJobProperties2.....	743
SetSoftwareJobProperties3.....	744
Structure CreateUnitGroupProperties2 CreateUnitGroupProperties2	746
CreateSoftwareJobLinkage.....	746
DomainPropertyFilter	747
DomainPropertiesRequired	748
ElementTypeMask.....	750
EvaluationTypeMask	750
SecurityArea.....	750
SetDistributionContainerProperties.....	751

SetSecurityAreaProperties	751
SetSoftwarePackageGroupProperties.....	751
SetSoftwarePackageProcedureProperties	752
SetSoftwarePackageProcedureProperties2	754
SetSoftwarePackageProperties	757
SetSoftwarePackageVolumeProperties	757
SetSoftwarePolicyProperties.....	758
SetSoftwarePolicyProperties3.....	759
SetSoftwarePolicyPropertiesByName	761
SetSoftwarePolicyPropertiesByName3	763
SetTimeoutProperties	766
SetUnitGroupProperties.....	767
SetUserProfileProperties.....	767
SoftwareJobPermissionMask	768
SoftwareJobProperties.....	768
SoftwareJobProperties2.....	771
SoftwareJobProperties3.....	774
SoftwareJobPropertiesRequired	777
SoftwareJobStatusCounters.....	778
SoftwarePackageGroupPropertiesRequired	778
SoftwarePackageMask	779
SoftwarePackageMaskType	779
CreateSoftwarePolicyProperties	779
CreateSoftwarePolicyProperties3	781
CreateUnitGroupProperties	783
SoftwareJobBootMask (bootAfterMask)	784
SoftwareJobBootMask3 (bootAfterMask)	784
SoftwareJobBootMask (bootBeforeMask)	785
SoftwareJobBootMask3 (bootBeforeMask).....	786
SoftwareJobContainerProperties	786
CreateAssetScriptJobResponse	788
AssetScriptJobProperties	788
SchedulingOptions	789
Scheduling	789
Conditions	790
Dependency	790
PreJob.....	791
PostJob	792
Miscellaneous	792
DateTimeRestrictions.....	792
FileDependency.....	793
JobDependency	793

PeriodicDependency	794
JobExecProperties	794
NetWorkDrive	795
PostJobProperties	796
AssetScriptJobPropertiesRequired	796
Dsm:OpenAssetScriptJobListResponse	796
GetAssetScriptJobResponse	797
DayOfTheWeekDependency	797
GetAssetScriptJobsProperties	797
TriggerInfo	798
OSIMProperties	798
OSIMPropertiesWithDesc	799
OSIMparameterDesc	799
ScalabilityServerProperties	799
ScalabilityServerPropertiesRequired	800
OSImageFilter	801
OSImageProperties	802
OSIMTargetFilter	802
OSIMTargetProperties	803
ComputerPropertyFilter2	803
ComputerPropertiesRequired2	806
DateTime2	809
ComputerProperties2	809
CreateOSIMComputerProperties	812
OSImageProperties	815
SoftwareJobContainerProperties3	816
SoftwareJobContainerPropertiesRequired	818
SoftwareJobContainerPropertiesRequired3	819
SoftwareJobContainerPropertyMask	820
SoftwareJobOperationMask	821
SoftwareJobOperationMask2	822
SoftwareJobOrderProperties	823
SoftwareJobOrderProperties3	825
SoftwarePackageGroupProperties	828
SoftwarePackageProcedureBootMask (bootAfterMask)	829
SoftwarePackageProcedureBootMask2 (bootAfterMask)	830
SoftwarePackageProcedureBootMask	830
SoftwarePackageProcedureBootMask2 (bootBeforeMask)	831
FindSoftwarePackageProcedure	832
SoftwarePackageProcedurePropertiesRequired	833
SoftwarePackageProcedureTypeMask	834
SoftwarePackagePropertiesRequired	835

SoftwarePackageVolumeElementPropertiesRequired	836
SoftwarePackageVolumeProperties	836
SoftwarePackageVolumePropertiesRequired	837
SoftwarePackageProcedureDependencyProperties	837
SoftwarePackageProcedureDependencyPropertiesRequired	838
SoftwarePackageProcedureGroupProperties	838
SoftwarePackageProcedureGroupPropertiesRequired	839
SoftwarePackageProcedureManagementTypeMask	840
SoftwarePackageProcedureOperationMask	841
SoftwarePackageProcedureOperationMask2	842
SoftwarePackageProcedureProperties	843
SoftwarePackageProcedureProperties2	845
SoftwarePackageProperties	847
SoftwarePackageProperties2	849
SoftwarePackageVolumeElementProperties	850
SoftwarePolicyExceptionProperties	851
SoftwarePolicyProperties	851
SoftwarePolicyProperties3	854
SoftwarePolicyPropertiesRequired	856
SoftwarePolicyPropertiesRequired2	857
SoftwarePolicyPropertiesRequired3	858
SoftwareJobContainerProperties2	860
SoftwareJobContainerPropertiesRequired2	862
SoftwarePolicyState	862
SoftwareJobContainerOrder	863
SoftwarePolicyOrder	863
SoftwarePackageProcedureOrder	863
SoftwareByName	864
SoftwareJobLinkage	864
SoftwareJobLinkage	864
SoftwarePackage	865
SoftwarePackageOrder	865
SoftwarePackageProcedure	865
SoftwarePolicyProperties2	865
SSPackageInfo	867
SSPackageInfoRequired	867
StagingCredentials	868
UnitInventoryComponentProperties	868
UnitInventoryComponentPropertiesRequired	868
UnitInventoryGroupProperties	869
UnitInventoryGroupPropertiesRequired	869
UnitInventoryItemProperties	869

UnitInventoryItemPropertiesRequired	870
UnitGroupProperties.....	870
SetUnitGroupProperties2.....	874
UnitGroupPropertiesForSoftwarePolicy	874
UnitGroupPropertiesRequired	876
UnitProperties	877
UnitPropertiesRequired	877
UnitPropertyFilter	877
UnitQueryPropertiesRequired	879
UnitQueryProperties	880
UnitSoftwareJobProperties	880
UnitSoftwareJobProperties2	883
UnitSoftwareJobPropertiesRequired	886
UnitSoftwareJobTaskMask	887
UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroupName	887
UsageListMask	889
UserAccountProperties	890
UserAccountPropertiesRequired	891
UserAccountPropertyFilter	891
UserProfileProperties.....	892
UserProfilePropertiesRequired	896
UserProfilePropertyFilter	897
VolumeElementAttributes	899
CreateSoftwarePackageProperties3	900
SetSoftwarePackageProperties3.....	901
SoftwarePackageProperties3	902
SoftwarePackagePropertiesRequired3	906
SoftwarePackageProcedureProperties4	907
SoftwarePackageProcedurePropertiesRequired2.....	910
UnitSoftwareJobProperties3	911
WSChallengeResponse.....	914
WSAuthenticateResponse.....	914
Item	915
UnitGroupProperties2.....	915
CreateUnitGroupProperties3	917
createAssessmentTemplate	918
updateAssessmentTemplate.....	918
deleteAssessmentTemplates	919
getAssessmentTemplate	919
getAssessmentTemplateList	920
getAssessmentTemplateListDetailed	921
createAssessmentTask	922

executeAssessmentTask	922
getAssessmentStatus	923
getAssessmentResult	923
getAssessmentTask	924
getAssessmentTaskList	924
deleteAssessmentTasks	925
createOSIMTemplate	926
updateOSIMTemplate	926
deleteOSIMTemplates	926
getOSIMTemplate	927
getOSIMTemplateList	927
getOsimTemplateListDetailed	928
createSoftwareDeploymentTemplate	929
updateSoftwareDeploymentTemplate	930
deleteSoftwareDeploymentTemplates	930
getSoftwareDeploymentTemplate	931
getSoftwareDeploymentTemplateList	931
getSoftwareDeploymentTemplateListDetailed	932
createAutomationProcess	933
deleteAutomationProcesses	934
executeAutomationProcess	934
getAutomationProcessList	935
getAutomationProcessListDetailed	936
getAutomationResult	937
getAutomationResult2	937
getAutomationProcessTargetStatus	938
getAutomationProcessStatus	939
runAutomationSoftwareJobContainer	940
runAutomationOSInstallJob	941
runAutomationGenericJobContainer	941
getActivityLog	942
getActivityLog2	943
setActivationStatus	944
getSupportedOSForAssessment	944
SDTemplateProperties	945
SoftwarePackageFilter	945
SoftwarePackageProperties2	946
SDProperties	947
ManagerProperties	948
ManagerVersion	948
CreateSoftwareJobContainerProperties4	949
BootImageFilter	950

BootImageProperties	950
BootServerProperties.....	951
OSIMTargetProperties2	952
OSIMTargetProperties	952
BootServerFilter	953
ComputerProperties2	953
QueryFilter	956
QueryProperties2	956
QueryProperty	957
QueryFolderProperties.....	957
HWMModelFilter.....	957
Health Monitoring Sequences.....	958
Sequences for Automation Web Services	958
Array of Elements.....	980
ArrayOfSoftwareJob	983
ArrayOfSoftwareJobProperties3	983
ArrayOfTargetProperties2.....	983
ArrayOfSoftwarePackageProcedureProperties2.....	984
ArrayOfBootServerFilter	984
ArrayOfBootImageProperties	984
ArrayOfBootServerProperties	984
ArrayOfUnitSoftwareJobProperties3	984
ArrayOfBootImageFilter	985
ArrayOfSoftwarePackageProperties2	985
ArrayOfSoftwarePackageProcedureProperties4.....	985
ArrayOfSoftwarePackageFilter3.....	986
ArrayOfSoftwarePackageProperties3	986
ArrayOfUnitSoftwareJobProperties2	987
ArrayOfSecurityAreaByName.....	987
ArrayOfComponents	987
ArrayOfComponentStatus.....	987
ArrayOfConfigurationParams.....	988
ArrayOfConfigurationParamValues.....	988
ArrayOfComputerProperties2	988
ArrayOfQueryFilter.....	988
ArrayOfHWMModelFilter	989
ArrayOfQueries	989
ArrayOfQueryFolderProperties	989
ArrayOfSecurityClassPermissions.....	989
ArrayOfSecurityProfilesSecurityGroupsPermissions	989
ArrayOfComponents	990
ArrayOfComponentStatus.....	990

ArrayOfConfigurationParams	990
ArrayOfConfigurationParamValues	990
ArrayOfComputerProperties2	990
ArrayOfQueryFilter	991
ArrayOfQueries	991
ArrayOfQueryFolderProperties	991
ArrayOfHWMModelFilter	991
ArrayOfSDTemplateProperties	992
ArrayOfSoftwarePackageFilter	992
ArrayOfSoftwarePackageProperties2	992
ArrayOfOSIMProperties	992
ArrayOfOSIMTargetProperties	992
ArrayOfOSIMTargetInstallationRequestsProperties	993
ArrayOfGetAssetScriptJobsProperties	993
ArrayOfDayOfTheWeekDependency	993
ArrayOfTriggerInfo	993
ArrayOfAMExtraCmdParam	993
ArrayOfOSIMparameterDesc	994
ArrayOfOSIMPropertiesWithDesc	994
ArrayOfOSIMIRParametersWithDesc	994
ArrayOfOSIMTargetPropertiesWithDesc	994
ArrayOfOSIMImageProperties	994
ArrayOfServerProperties	995
ArrayOfBootServerProperties	995
ArrayOfOSImageFilter	995
ArrayOfOSImageProperties	995
ArrayOfOSIMTargetFilter	995
ArrayOfTargetProperties	995
ArrayOfComputerPropertyFilter2	996
ArrayOfComputerProperties2	996
ArrayOfOSImageProperties	996
ArrayOfParameterInfo arrayOfParamInfo	996
ArrayOfItems	996
ArrayOfstring	997
Limitations	997
Configuration Parameters	997
SSL Configuration	998
Passing System Defined Groups as Parameters	998
Chapter 3: Web Services Messages	999
AWS00052	1068

AWS00053.....	1068
AWS00054.....	1069
AWS00055.....	1069
AWS00103.....	1069
AWS00104.....	1070
AWS00105.....	1070
AWS00201.....	1070
AWS00202.....	1071
AWS00203.....	1071
AWS00204.....	1071
AWS00205.....	1071
AWS00206.....	1072
AWS00207.....	1072
AWS00208.....	1072
AWS00209.....	1073
AWS00210.....	1073
AWS00211.....	1073
AWS00213.....	1074
AWS00214.....	1074
AWS00215.....	1074
AWS00216.....	1074
AWS00217.....	1075
AWS00218.....	1075
AWS00219.....	1075
AWS00220.....	1076
AWS00221.....	1076
AWS00222.....	1076
AWS00223.....	1077
AWS00301.....	1077
AWS00302.....	1077
AWS00303.....	1077
AWS00304.....	1078
AWS00305.....	1078
AWS00306.....	1078
AWS00307.....	1079
AWS00308.....	1079
AWS00309.....	1079
AWS00310.....	1079
AWS00311.....	1080
AWS00312.....	1080
AWS00316.....	1080
AWS00313.....	1081

AWS00314.....	1081
AWS00401.....	1081
AWS00402.....	1081
AWS00404.....	1082
AWS00406.....	1082
AWS00407.....	1082
AWS00408.....	1082
AWS00409.....	1083
AWS00410.....	1083
AWS00411.....	1083
AWS00412.....	1084
AWS00414.....	1084
AWS00601.....	1084
AWS00602.....	1084
AWS00603.....	1085
AWS00604.....	1085
AWS00605.....	1085
AWS00607.....	1085
AWS00608.....	1086
AWS00609.....	1086
AWS00610.....	1086
AWS00611.....	1087
AWS00612.....	1087
AWS00613.....	1087
AWS00701.....	1087
AWS00702.....	1088
AWS00703.....	1088
AWS00704.....	1088
HLA000259.....	1090
HLA000260.....	1090
WSA000408.....	1090
WSA000409.....	1091
WSA000410.....	1091
WSA000400.....	1091
WSA000401.....	1092
WSA000402.....	1092
WSA000403.....	1092
WSA000404.....	1093
WSA000405.....	1093
WSA000406.....	1093

Chapter 1: Web Services Tutorial Writing Client Applications

This tutorial is a primer for users of the DSM web service functions. The tutorial uses code snippets from a Java client application to demonstrate some common tasks that can be achieved with the DSM web service functions.

The code examples demonstrate the following:

- Connection to the web service
- Session handling
- Searching for DSM units
- Creating and delivering a software package to a target

The tutorial also provides some useful tips for developing your own DSM web service client.

The tutorial assumes the following knowledge:

- An understanding of Web Service architecture and of the Web Services Definition Language (WSDL)
- A brief understanding of Java code development
- An understanding of a small number of simple DSM tasks

This section contains the following topics:

[Generating Client Stubs](#) (see page 24)
[Writing DSM Web Service Clients](#) (see page 24)
[Retrieving Attributes](#) (see page 29)
[Finding a DSM Object](#) (see page 30)
[Creating a Software Package](#) (see page 32)
[Creating a Software Package Volume](#) (see page 34)
[Creating a Software Package Procedure](#) (see page 35)
[Sealing a Software Package](#) (see page 37)
[Choosing a Target](#) (see page 37)
[Creating a Software Container](#) (see page 38)
[Creating a Software Job](#) (see page 39)
[Sealing and Activating a Job](#) (see page 40)
[Closing the Session](#) (see page 41)
[Conclusions](#) (see page 42)

Generating Client Stubs

The example code in this tutorial is taken from a test client application written in Java. The DSM web service functions are exposed by a language-independent WSDL interface that follows Web Service standard. You can write your web service clients in any language that lets you generate web service client stubs from the CA DSM Web Service WSDL file (webserviceapi.wsdl).

Web Service client stubs are a series of code files that are generated from the WSDL to represent the methods and objects exposed by the web service. There are a number of third-party tools that can perform this stub-generation for you.

The Apache Axis (<http://ws.apache.org/axis/>) tool, wsdl2java, was used to generate the Java client stubs for the test client application. The stubs were archived into a jar file that can be referenced on the classpath of the client application.

Writing DSM Web Service Clients

This section provides general tips for writing DSM web service clients.

Sessions

To call any DSM Web Service function, you need to have created a connection to the web service. You also need to have logged into the web service using the credentials of a valid DSM user. On login, you are returned a string session ID. This ID is used in all further calls to the web service. If the session remains inactive for longer than the web service configured timeout, the session ID will become invalid. On completion of your session, if the session ID is still valid, log out of the DSM web service to close and clean up the session. For further details, see [Creating and Maintaining a Connection](#) (see page 27) and [Closing the Session](#) (see page 41).

Handling UUIDs

Many of the web service functions take as input, or return as output, a string ID that represents a DSM object. The IDs equate to the UUID of the objects in the CA-MDB. The UUIDs are important as they are often required as input to further web service functions.

Properties Architecture Implementation

When creating an object you usually need to indicate which of the creation properties you have supplied by marking the corresponding 'supplied' boolean to true. For example, if you are creating a `SoftwareJobContainer` and you want to supply a value for the comment property you should do both of the following:

```
props.setComment("MyComment");  
props.setCommentSupplied(true);
```

Similarly, if you are retrieving the properties of a DSM object you will need to indicate which properties you would like the web service to return to you. This is achieved through the provision of a separate `propertiesRequired` structure as an input parameter. For example, if you are retrieving the properties of a computer unit and you want to return the computer's UUID, you should pass the following structure to the `findComputer` function:

```
ComputerPropertiesRequired props = new ComputerPropertiesRequired();  
props.setComputerUUIDRequired(true);
```

Properties Architecture Concepts

The DSM Web Service has implemented a distinct method for getting and setting object properties.

When you are creating or retrieving the properties of a DSM object using the web service, you need to make clear to the web service, those properties of the object with which you are currently concerned.

If you are creating or updating a new object, you provide as input a structure that contains a sequence of elements representing object properties whose values you need to populate. The sequence also contains a series of boolean values that indicate the property values you have supplied. Note that if you do not set a property's matching supplied Boolean flag to true, the DSM web service will ignore the property value that you supplied.

If you are retrieving the properties of a structure, a similar rule applies. You must provide as input to the DSM function a `propertiesRequired` structure. This structure contains a sequence of boolean flags corresponding to each of the property values contained in the return or output properties structure of the method. For each property whose value you wish to be returned in the output structure, you must set the corresponding boolean flag in the `propertiesRequired` structure to true. Note that if you do not set a flag to true in the `propertiesRequired` input structure, the property's value will not be returned in the output properties structure.

Complex Attribute Types

Some of the object types described in the WSDL may be implemented by your client stubs in a different way than that described in this tutorial. This is because the way in which types such as Arrays, Enumerations and return objects are handled, varies among different SOAP and web service tools. When you read the tutorial, bear in mind that how return values and enumerated lists are handled may not be the same for your implementation. For example, our stubs have implemented the long return type of many methods as an object of type `javax.xml.rpc.holders.LongWrapperHolder`.

Handling Errors

The way that the DSM web service notifies you about errors that it has experienced depends greatly on the language in which you implement your client application. The DSM web service, if it experiences an error, returns a SOAP fault. In many cases, this fault is identified by your client implementation, and in the specific example of Axis-based Java applications, an Axis SOAP exception will be thrown to be caught by the client application. The fault will usually contain an error code that can be looked up in the DSM help documentation.

Creating and Maintaining a Connection

In order to create a web service connection, and log into the DSM web service you will need to initialize both a web service locator, and a binding to the web service using SOAP. You will need to know the URL of the web service. In a standard DSM installation, this is likely to be in the format

`http://<machinename>/udsm_r11_webservice/mod_gsoap.dll` on Windows and
`http://<machinename>/UDSM_R11_WebService` on Linux.

For Post R12.5 release, use the following web service end point, in addition to the previous URL:

Windows

`http://<machinename>/DSM_Webservice/mod_gsoap.dll`

Linux

`http://<machinename>/DSM_WebService`

You must initialize the web service locator (a) and then get a connection to the web service API using the location URL of the web service (b).

Example 1: Setting up the required connection variables

```
//This is the session ID used in all web service calls
//It is set up when you log into the web service
private String sessionId = "";
//These are the UDSM Web Service objects you need to make
//web service calls
//(a)
private DSMWebServiceAPIService wsAPIService = null;
//(b)
public DSMWebServiceAPISoapBindingStub wsAPIConnector = null;
private String serviceLocation="";
private java.net.URL endpoint = new java.net.URL(serviceLocation);
```

Note: This example is generated using Apache Axis version 1.2 and JDK 1.5, and it does not endorse any best practice in writing web service client code.

Example 2: Creating the Web Service Connection

```
//create the Service Locator object -- this is required.
wsAPIService = new WebServiceAPIServiceLocator();
//if the service locator object was created then carry on
if (wsAPIService != null)
{
    try
    {
        //try to create the binding object to bind to the web
        //service using the URI of the web service location
```

```
        wsAPIconnector= (WebServiceAPISoapBindingStub)
        wsAPIService.getWebServiceAPIService(endpoint);
    }
    catch(Exception e)
    {
        throw e;
    }
}
else
{
    sessionId = "";
    return WEB_SERVICE_COULD_NOT_BE_INITIALISED;
}
```

Remember that java web service methods can throw web service exceptions that should be caught and handled.

Once a connection has been made, it is important to log into the DSM web service as you would to the DSM explorer. The Login method should be called with the credentials of a valid DSM user.

Example 3: Logging in to the DSM Web Service

```
//now try to log in to the new web service connection
//using the credentials from a config object (not described here)
try
{
    sessionId = wsAPIconnector.Login(
    cfg.getUserName(),cfg.getPassword(),cfg.getManager() );
}
catch (Exception e)
{
    sessionId = "";
    throw e;
}

//if the session Id is blank it means no session could be established
if(sessionId.equals(""))
{
    return SESSION_COULD_NOT_BE_ESTABLISHED;
}
else
{
    return RESULT_OK;
}
```

Retrieving Attributes

If you have successfully retrieved a CA ITCM object from a list or search function you will have been returned a structure containing the properties for that object.

In the examples, the output structures described in the WSDL have been mapped to holder classes. For example, if you had retrieved a structure of type `dsm:ComputerProperties` from the `findComputer` function, you will have retrieved a java object of type `ComputerPropertiesHolder`. To obtain the UUID from this object you should do the following:

```
ComputerPropertiesHolder methodOutput =  
new ComputerPropertiesHolder();  
//find the computer  
...  
String computerId = methodOutput.value.getComputerUUID();
```

Finding a DSM Object

To request information or perform tasks on a DSM object you need to know its CA_MDB UUID. The DSM web service provides find functions that simplify this task, returning to you the string UUID of an object. Many object types such as users and computers can be found using just their display name. Other objects may need further information to enable the web service to perform the search.

A simple example of a search can be demonstrated through use of the findComputer function.

Example 4: Finding a Computer by Name

```
String computerId = "";

//This method performs a search for the given computer name

private int findTargetComputer(String computerName) throws Exception
{
    //Set the requested property (we are only interested in the UUID)

    ComputerPropertiesRequired props =
        new ComputerPropertiesRequired();

    props.setComputerUUIDRequired(true);

    //set up the output parameter

    LongWrapperHolder numFound = new LongWrapperHolder();

    ArrayOfComputerPropertiesHolder arrOfCompPropsHolder = new
        ArrayOfComputerPropertiesHolder();

    //this may throw an axis fault

    wsAPIconnector.findComputer(service.getSessionId(),computerName,
        props, arrOfCompPropsHolder, numFound );

    if(numFound.value.intValue() > 0)
    {
        computerId =
            arrOfCompPropsHolder.value.getComputerPropertiesArray(0).
            getComputerUUID();
    }
}
```

```
        }  
    else  
    {  
        computerId = "";  
        return UDSMWebService.NO_MATCHING_COMPUTER;  
    }  
    return UDSMWebService.RESULT_OK;  
}
```

This method demonstrates the request of required properties, and the retrieval of requested properties. The findComputer method, as with all DSM web service methods, may throw a SOAP/Axis exception.

Creating a Software Package

In order to deliver a software package, you will need to know the following UUIDs.

```
private String packageID = "";
private String volumeID = "";
private String procID = ""; //procedure ID
```

First, you must create the software package itself. In this example, all the input values have been taken from an external 'cfg' config object that was responsible for reading the values in from a java properties file on application startup.

Example 5: The Steps for Creating a Software Package

```
protected int createSoftwarePackage() throws Exception
{
    try
    {
        //set up the properties that you are going to
        //create for the software package
        CreateSoftwarePackageProperties props =
        new CreateSoftwarePackageProperties();
        props.setSoftwarePackageName(cfg.getPackageName());
        props.setSoftwarePackageNameSupplied(true);
        props.setSoftwarePackageVersion(cfg.getPackageVersion());
        props.setSoftwarePackageVersionSupplied(true);
        props.setSupplier(cfg.getPackageVendor());
        props.setSupplierSupplied(true);
        props.setComment(cfg.getPackageComment());
        props.setCommentSupplied(true);
        //create the software package mask to indicate
        //whether a checksum is required
        SoftwarePackageMask mask = new SoftwarePackageMask();
        mask.setChecksum(cfg.getPackageChecksum());
        props.setSoftwarePackageMask(mask);
        props.setSoftwarePackageMaskSupplied(true);
        //Create the software package (c)
        packageID =wsAPIconnector.createSoftwarePackage (getSessionId(),
        props);

        if(packageID.equals(""))
        {
            return UDSMWebService.PACKAGE_CREATION_ERROR;
        }
        System.out.println("Successfully created software package"+ "with
        ID "+packageID);
        //if the package was successfully created, carry on
        int retVal = 0;
        //create a software volume for this package
        retVal = createSimpleSoftwareVolume();
    }
}
```



```
        if(retVal != UDSMWebService.RESULT_OK)
            return retVal;
        System.out.println("Successfully created software volume"+
            "with ID "+volumeID);
        //create a software procedure for this package
        retVal = createSimpleSoftwareProcedure();
        if(retVal != UDSMWebService.RESULT_OK)
        {
            return retVal;
        }
        System.out.println("Successfully created software procedure"+ "with
ID                                "+procID);
        //seal the package now the volume and procedure are defined
        retVal = sealSoftwarePackage();
        if(retVal != UDSMWebService.RESULT_OK)
            return retVal;
        System.out.println("Successfully sealed complete software"+
"package ready for                delivery.");
    }
    catch(Exception e)
    {
        throw e;
    }
    return UDSMWebService.RESULT_OK;
}
```

Creating a Software Package Volume

If the call to `createSoftwarePackage` was successful, you can see that the code continues. You then need to create one or more software package volumes associated with your new package. This will normally require you to copy file elements into the volume for each file or directory that the volume should contain. When you create your volume you should associate it with the package ID of the package that you recently created.

Example 6: Creating a simple `SoftwarePackageVolume`

```
//This method creates a simple software volume containing one file
private int createSimpleSoftwareVolume() throws Exception
{
    CreateSoftwarePackageVolumeProperties props =
        new CreateSoftwarePackageVolumeProperties();
    //set all the properties to create a software volume
    props.setVolumeName(cfg.getVolumeName());
    props.setVolumeNameSupplied(true);
    //the source is on the local hard disk
    props.setVolumeType(SourceMedium.HARDDISK);
    props.setVolumeTypeSupplied(true);
    props.setSoftwarePackageId(packageID);
    props.setSoftwarePackageIdSupplied(true);
    //create the software volume
    String volID = wsAPIconnector.createSoftwarePackageVolume (getSessionId(),
        props);

    if(volID.equals(""))
        return UDSMWebService.PACKAGE_CREATION_ERROR;
    //assign the return value
    volumeID = volID;
    // now add some elements to the volume
    ArrayOfFileSystemElementPath array =
        new ArrayOfFileSystemElementPath();
    FileSystemElementPath path[] = new FileSystemElementPath[1];
    path[0] = new FileSystemElementPath();
    path[0].setElementName(cfg.getVolumeElementName());
    path[0].setElementPath(cfg.getVolumeElementPath());
    array.setFileSystemElementPath(path);
    Integer retVal = wsAPIconnector.copyElementsToSoftwarePackageVolume
        (getSessionId(), volumeID, "", array);
    if(retVal.intValue() != 0)
        return UDSMWebService.ELEMENT_ADDITION_ERROR;
    else
        return UDSMWebService.RESULT_OK;
}
```

Creating a Software Package Procedure

Following this, you will need to create a software package procedure (install, configure, activate or uninstall) associated with the software package and its volume. There are a lot of properties that need to be set for a procedure to execute correctly. For full descriptions of each property, see the Unicenter DSM help.

Example 7: Creating a Simple Software Package Procedure

```
//This method creates a simple software installation procedure
//using the one file from the volume
//If you were defining a non-install software job, you would also need to // define
an install procedure as well as your non-install procedure
private int createSimpleSoftwareProcedure() throws Exception
{
    CreateSoftwarePackageProcedureProperties props =
        new CreateSoftwarePackageProcedureProperties();

    props.setComment(cfg.getProcedureToRun());
    props.setCommentSupplied(true);
    props.setSoftwarePackageProcedureName(cfg.getProcedureToRun());
    props.setSoftwarePackageProcedureNameSupplied(true);
    SoftwarePackageProcedureBootMask mask =
        new SoftwarePackageProcedureBootMask();
    mask.setNoReboot(true);
    props.setBootAfterMask(mask);
    props.setBootAfterMaskSupplied(true);
    props.setBootBeforeMask(mask);
    props.setBootBeforeMaskSupplied(true);
    //set the management mask
    SoftwarePackageProcedureManagementTypeMask mgmtMsk =
        new SoftwarePackageProcedureManagementTypeMask();
    mgmtMsk.setSd(true);
    mgmtMsk.setSm(true);
    props.setManagementTypeMask(mgmtMsk);
    //set the operation mask
    SoftwarePackageProcedureOperationMask opMsk =
        new SoftwarePackageProcedureOperationMask();
    opMsk.setAutoDeliverItem(true);
    opMsk.setBackground(false);
    opMsk.setOffline(false);
    props.setOperationMask(opMsk);
    props.setOperationMaskSupplied(true);
    //set this to be an install task
    props.setSoftwarePackageProcedureTask
        (SoftwarePackageProcedureTask.INSTALL);
    props.setSoftwarePackageProcedureTaskSupplied(true);
    //set this to show that the task is an executable file
```

```
        props.setSoftwarePackageProcedureType
(SoftwarePackageProcedureType.EXEFILE);
        props.setSoftwarePackageProcedureTypeSupplied(true);
        //link to the volume and the package
        props.setSoftwarePackageVolumeId(volumeID);
        props.setSoftwarePackageVolumeIdSupplied(true);
        props.setSoftwarePackageId(packageID);
        props.setSoftwarePackageIdSupplied(true);

        //values that represent the procedure itself
        props.setProcToRun(cfg.getProcedureToRun());
        props.setProcToRunSupplied(true);
        props.setSoftwarePackageProcedureName(cfg.getProcedureName());
        props.setSoftwarePackageProcedureNameSupplied(true);
        props.setComment(cfg.getProcedureComment());
        props.setCommentSupplied(true);
        //additional parameters
        //the procedure can run on any OS
        // for the set of possible values see operatingSystemType
        props.setOperatingSystemType(196608);
        props.setOperatingSystemTypeSupplied(true);
        //create the procedure
        procID = wsAPIconnector.createSoftwarePackageProcedure (
        getSessionId(), props);
        if(procID.equals(""))
            return UDSMWebService.PROCEDURE_CREATION_ERROR;
        else
            return UDSMWebService.RESULT_OK;
    }
```

Sealing a Software Package

After these steps have been completed successfully, you can seal your software package, ready for delivery.

Example 8: Sealing a Software Package

```
//This method seals the software package ready for delivery
private int sealSoftwarePackage() throws Exception
{
    String [] packageIDs = new String[1];
    Integer i = wsAPIconnector.sealSoftwarePackages (
        service.getSessionId(), packageIDs);
    System.out.println("FYI: The return code from"
        + " SealSoftwarePackages was "+ i.intValue());
    if(i.intValue() != 0)
        return i.intValue();
    else
        return UDSMWebService.RESULT_OK;
}
```

Choosing a Target

In order to deliver a software package to a target unit or units you need the uuid of the unit or of a unit group. You can find the uuid easily using the find and list functions exposed by the DSM web service. For a simple example, see [Finding a DSM Object](#) (see page 30).

Creating a Software Container

To deliver a software package you must create a job container that contains a delivery job for the package. The first step is to create an empty job container.

Example 9: Creating an Empty Software Job Container

```
//This method creates an empty software job container
private int createContainer() throws Exception
{
    CreateSoftwareJobContainerProperties props =
        new CreateSoftwareJobContainerProperties();
    props.setComment(cfg.getContainerComment());
    props.setCommentSupplied(true);
    props.setName(cfg.getContainerName());
    props.setNameSupplied(true);
    //set the container property mask elements
    SoftwareJobContainerPropertyMask mask =
        new SoftwareJobContainerPropertyMask();
    mask.setBuildOnly(false);
    mask.setCascade(true);
    mask.setLinkageBatch(true);
    mask.setLinkageNone(false);
    mask.setLinkageSequence(false);
    mask.setRenewed(true);
    mask.setRollback(true);
    props.setSoftwareJobContainerPropertyMask(mask);
    props.setSoftwareJobContainerPropertyMaskSupplied(true);
    containerId=wsAPIconnector.createSoftwareJobContainer(getSessionId(),prop
s);
    if(containerId.equals(""))
        return UDSMWebService.CONTAINER_CREATION_ERROR;
    return UDSMWebService.RESULT_OK;
}
```

Creating a Software Job

You are now ready to create a software job designed to deliver your software package. You need to know the UUID of the new container, the UUID of the package's procedure, and the UUIDs of all the units and groups to which you wish to deliver the job.

Example 10: Creating an Installation Job

```
//This method sets up the software job for the package's procedure in
//the newly created container, ready to go!
private int createJob() throws Exception
{
    CreateSoftwareJobOrderProperties props = new
CreateSoftwareJobOrderProperties();

    //set the boot mask for the job
    SoftwareJobBootMask mask = new SoftwareJobBootMask();
    mask.setNoReboot(true);
    props.setBootAfterMask(mask);
    props.setBootBeforeMask(mask);
    props.setBootAfterMaskSupplied(true);
    props.setBootBeforeMaskSupplied(true);

    //set the job's own properties
    props.setJobName(cfg.getJobName());
    props.setJobNameSupplied(true);

    //set the operation mask
    SoftwareJobOperationMask opMsk = new SoftwareJobOperationMask();
    opMsk.setAutoDeliverItem(true);
    opMsk.setOffline(false);
    props.setOperationMask(opMsk);
    props.setOperationMaskSupplied(true);

    //additional set-up properties mainly to do with execution time
    props.setPromptTimeoutHours(211);
    props.setPromptTimeoutHoursSupplied(true);
    props.setStartWhenSupplied(true);
    props.setStartWhen(SoftwareJobStartupTime.value2);//local-time
    props.setTimeoutHours(211);
    props.setTimeoutHoursSupplied(true);
    props.setUserParametersSupplied(false);
    props.setDeliveryDateTime(getADateTime());
    props.setDeliveryDateTimeSupplied(true);
    DateTime dt = getADateTime();
    dt.setMinute(dt.getMinute()+10);
    props.setExecutionDate(dt);
    props.setExecutionDateSupplied(true);
    props.setUserParametersSupplied(false);
```

```
String unitIds[] = new String[1];
String groupIds[] = new String[1];
String instIds[] = { "" };

// target is a single unit
unitIds[0] = computerId;
groupIds[0] = "";

jobId = wsAPIconnector.createInstallSoftwareJob(
    getSessionId(),
    packager.getProcedureID(),
    props,
    containerId,
    unitIds,
    groupIds);

if(jobId.equals(""))
    return UDSMWebService.JOB_CREATION_ERROR;

return UDSMWebService.RESULT_OK;
}
```

Sealing and Activating a Job

Now all that remains is to activate your newly created software job. To do this, all you need is the UUID of the new container for the software job.

Example 11: Sealing and Activating a Software Job Container to Deliver Software

```
//seal and activate the job container
Integer retVal =
wsAPIconnector.sealAndActivateSoftwareJobContainer(getSessionId(), containerId);

if(retVal.intValue() != 0)
    return UDSMWebService.SEAL_AND_ACTIVATE_ERROR;
```

It is important that once the job is activated, you check back on the status of the container and/or the job to ensure that it completed successfully. This can be done by retrieving the status property using a call to `GetSoftwareJobContainer` or `GetSoftwareJob`.

Closing the Session

When your client application has completed its work, it should log out of the DSM web service and close the session cleanly by using the Logout function.

Example 12: Logging out of the DSM Web Service

```
if(!sessionId.equals(""))
{
    int result = 0;
    if (null != wsAPIService && null != wsAPIConnector)
    {
        try
        {
            result = wsAPIConnector.Logout(sessionId).intValue();
        }
        catch (Exception e)
        {
            throw e;
        }
    }
    if (result == 0)
    {
        sessionId = "";
        System.out.println( "Logout successful");
        return RESULT_OK;
    }
    else
    {
        //The session Id given could not be closed
        sessionId = "";
        return LOG_OUT_FAILURE;
    }
}
else
{
    //The user was not logged in
    return LOG_OUT_FAILURE;
}
```

Conclusions

This tutorial demonstrated the use of a very limited subset of the DSM web service functions, using a Java Axis-based test client application. For more details of the usage of the DSM web service functions, see [WebServicesReference](#).

The sample client code for DMDeploy [Java Client] and AM Query [C/C++ Client] related Web Services APIs are available as a part of the Unicenter DSM Manager Installation Kit.

Chapter 2: Web Services Reference

The Web Services Reference describes the methods made available through the CA ITCM Web Service as described in the CA ITCM Web Service WSDL (`webserviceapi.wsdl`), which is in the top level CA ITCM install directory.

The CA ITCM Web Service exposes a series of utility methods that enable you to perform some common CA ITCM tasks. These methods are logically grouped as follows:

- Authentication and Web Services Management
- Deployment
- Enterprise Distribution Functionalities
- Queries
- Software
- Security
- Units and Groups
- OSIM
- Health Monitoring
- Automation Web Service

Each of the documented Web Service methods takes a session identifier as input. This session identifier is created for you when you log in to the CA ITCM web service using the method [Login](#) (see page 143). The session identifier is unique to each web service session and maintains information about an authenticated user.

The Web Service methods may also take as input CA ITCM sequences and enumerations as described in this reference document.

Notes:

- Webservices related to DMdeploy, OSIM, Security Areas, Software jobs, Creation of Units and Automation Services are only applicable at Domain Manager.
- Webservices related to Distributions and Domain Groups are only applicable at Enterprise Manager.
- When you use a non UTF8 end-point and an input parameter string contains non-English characters, the string should be converted to a UTF8 encoded string and then converted to an ISO-8859-1 encoded string before passing the input parameter to the WS API.
- An asterisk (*) indicates that the parameter is mandatory.
- This Webservice API (webserviceapi.wsdl) is compliant with Microsoft® .NET Framework, Java Axis, and Gsoap toolkit.
- The documentation for all Webservice APIs is according to the CA ITCM Web Service WSDL file and not according to code generated by the client tool in use.
- The service end point in WSDL (definitions\service\port\SOAP:address\location) is written as:

"http://<servername>/UDSM_R11_WebService/mod_gsoap.dll" (on Windows)
"http://<servername>/UDSM_R11_WebService" (on Linux)

The path is valid only if Default Web Site is listening on port 80 and SSL is not configured.

If the port number is not 80 or SSL is configured, the administrator has to modify the service end point in WSDL manually to comply with different port and SSL. For example, if port is changed to 5256, the service end point on Windows will be the following:

"http://<servername>:5256/UDSM_R11_WebService/mod_gsoap.dll"

If SSL is configured, use https instead of http.
- All the WS APIs corresponding to CO & AM component, take/return DateTime structure as GMT. All the WS APIs corresponding to SD component, take/return DateTime structure in the time zone of the manager machine.

This section contains the following topics:

[DSM Web Services End Point for UTF-8 Strings](#) (see page 45)

[Automation Web Services End Point](#) (see page 45)

[Enumerations](#) (see page 46)

[Methods](#) (see page 134)

[Sequences](#) (see page 619)

[Array of Elements](#) (see page 980)

[Limitations](#) (see page 997)

[Configuration Parameters](#) (see page 997)

[SSL Configuration](#) (see page 998)

[Passing System Defined Groups as Parameters](#) (see page 998)

DSM Web Services End Point for UTF-8 Strings

CA ITCM provides a WSDL and an end point for UTF-8 strings. The new WSDL and end point are available in addition to the existing WSDL and end point. Following are the new WSDL and end point that enable DSM web services to accept or return UTF-8 strings:

WSDL: webserviceapi_utf8.wsdl

End point: http://<hostname>/DSM_Webservice/mod_gsoap_utf8.dll

You can access the new WSDL that points to the new end point using
http://<hostname>/DSM_Webservice/wsdl

Automation Web Services End Point

You can access the Automation Service URI and WSDL from the following locations:

- AutomationService URI - http://<hostname>:[tomcat port number]/axis/services/automationService
- AutomationService WSDL - http://<hostname>:[tomcat port number]/axis/services/automationService?wsdl

In addition, on the server, the Automation Service WSDL file is available at the following location:

<DSM Installed folder>\webservices\wsdl\addons\AutomationService.wsdl

Enumerations

The following sections describe all the enumerated types (enums) that the CA ITCM Web Services uses. These structures are a mapping of the underlying layer functionality.

This section contains the following topics:

- [AdhocTargetEvaluation](#) (see page 48)
- [ArchivePath](#) (see page 48)
- [BootServerProperty](#) (see page 48)
- [BootImageProperty](#) (see page 49)
- [computerClassID](#) (see page 50)
- [PackageFormat](#) (see page 64)
- [SoftwarePackageProperty](#) (see page 65)
- [ComputerProperty](#) (see page 66)
- [ComputerStatusInfo](#) (see page 67)
- [CompRequiresAuthentication](#) (see page 68)
- [DeploymentStatusCode](#) (see page 68)
- [DiscoveredSoftwareSourceType](#) (see page 70)
- [DiscoveredSoftwareType](#) (see page 70)
- [DistributionContainerOrderType](#) (see page 70)
- [DistributionOrderStatus](#) (see page 71)
- [DistributionStatus](#) (see page 71)
- [DomainType](#) (see page 71)
- [DomainProperty](#) (see page 72)
- [DownloadMethod](#) (see page 74)
- [ElementType](#) (see page 74)
- [GroupFlag](#) (see page 74)
- [GroupInheritance](#) (see page 75)
- [GroupSystem](#) (see page 75)
- [Inheritance Level](#) (see page 76)
- [LinkedObjectTypeRequired](#) (see page 76)
- [operatingSystemType](#) (see page 76)
- [PackageExportType](#) (see page 82)
- [QueryFragment](#) (see page 82)
- [QueryFragment2](#) (see page 82)
- [QueryFragmentType](#) (see page 84)
- [QueryFragmentType2](#) (see page 85)
- [QueryType](#) (see page 85)
- [RACPolicy](#) (see page 86)
- [SecurityAreaLinkage](#) (see page 86)
- [SecurityClassType](#) (see page 86)
- [SecurityGroup](#) (see page 90)
- [SecurityProfilePermission](#) (see page 90)
- [SecurityProfileType](#) (see page 91)
- [SoftwareGroupType](#) (see page 91)
- [SoftwareJobContainerScope](#) (see page 92)
- [SoftwareJobContainerState](#) (see page 92)
- [SoftwareJobRenewState](#) (see page 93)

[SoftwareJobStartupTime](#) (see page 93)
[SoftwareJobState](#) (see page 93)
[SoftwareJobTask](#) (see page 94)
[SoftwarePackageClass](#) (see page 95)
[SoftwarePackageGroupScope](#) (see page 95)
[SoftwarePackageProcedureFiledBy](#) (see page 95)
[SoftwarePackageProcedureGroupScope](#) (see page 96)
[SoftwarePackageProcedureRelation](#) (see page 96)
[SoftwarePackageProcedureType](#) (see page 97)
[SoftwarePackageProcedureTask](#) (see page 97)
[SoftwarePackageGroupProperty](#) (see page 98)
[SoftwarePolicyEvaluationPeriodType](#) (see page 98)
[SoftwareJobType](#) (see page 98)
[SoftwarePackageFiledBy](#) (see page 98)
[SoftwarePackageType](#) (see page 99)
[SoftwarePackagingType](#) (see page 99)
[SoftwarePolicyEvaluationType](#) (see page 100)
[SoftwarePolicyScope](#) (see page 100)
[SourceMedium](#) (see page 101)
[StringValueType](#) (see page 101)
[SystemGroup](#) (see page 103)
[UnitProperty](#) (see page 104)
[UnitSoftwareJobState](#) (see page 106)
[UnitType](#) (see page 107)
[UserAccountProperty](#) (see page 107)
[UserProfileProperty](#) (see page 108)
[SoftwareJobLinkageType](#) (see page 109)
[WeekDay](#) (see page 109)
[PERIODTYPE](#) (see page 109)
[CO_TRIGGER_UNIT_TYPE](#) (see page 110)
[UnitObjectType](#) (see page 110)
[AMExtraCmdParam](#) (see page 110)
[CFComponentCode](#) (see page 110)
[WS_OSIM_DEPLOY](#) (see page 111)
[WS_OSIM_CONF_TYPE](#) (see page 111)
[WS_OSIM_IR_STATUS](#) (see page 111)
[WS_OSIM_CONFIG_STATE](#) (see page 112)
[Property_Type](#) (see page 112)
[FILTERCONDITION](#) (see page 113)
[OSIMAGEPROPERTY](#) (see page 114)
[OSIMTARGETPROPERTY](#) (see page 115)
[ComputerProperty2](#) (see page 115)
[ComputerFilterCondition](#) (see page 118)
[AMAgentStatus](#) (see page 119)
[SDAgentStatus](#) (see page 119)
[RCAgentStatus](#) (see page 120)
[ProtectionAgentStatus](#) (see page 121)
[ASPAgentStatus](#) (see page 122)
[AVAgentStatus](#) (see page 122)

[HIPStatus](#) (see page 123)
[AgentRestrictions](#) (see page 123)
[ObjectPreference](#) (see page 124)
[QueryTargetType](#) (see page 124)
[QueryFilterCondition](#) (see page 125)
[HWMODEL-FILTER-CONDITION](#) (see page 125)
[COMPONENTS](#) (see page 126)
[COMPONENT-STATUS](#) (see page 126)
[ConfigurationParams](#) (see page 126)
[WS_OSIM_CONFIG_STATE](#) (see page 127)
[Enumerations for Automation Services](#) (see page 128)

AdhocTargetEvaluation

NO_ADHOC

Adhoc evaluation mode not changed.

ADHOC_ONLY

Adhoc evaluation is enabled only when new or reinstalled targets report their inventory.

ADHOC_ALWAYS

Adhoc evaluation is enabled always when targets report their inventory.

ArchivePath

An enumerated list of values. The selected value indicates whether or not the path that is referred to can be found at the server or enterprise manager.

You can select one of the following xsd:string types:

- SERVER
- ENTERPRISESERVER

BootServerProperty

Specifies the boot server property on which you want to apply the sort and filter conditions.

BS-NAME

Refers to boot server name.

BS-STATUS

Refers to boot server status.

BS-PLATFORM

Refers to operating system of the boot server.

BS-NFS-SUPPORT

Refers to the NFS support.

BS-ACCESS-METHOD

Refers to the access method.

BS-LASTREPORT

Refers to the date and time of last boot server report.

BS-RESPONSE

Refers to the date and time of last boot server response.

BS-LASTREPORT-REQ

Refers to the date and time when the last boots server was requested.

BS-PXE-INTERFACE

Refers to the configured PXE interface.

BS-SD-PKG-WIN

Refers to the software delivery package name for Windows.

BS-SD-PKG-LINUX

Refers to the software delivery package name for Linux.

BS-SD-PKG-R11

Refers to the software delivery package for r11 version.

BootImageProperty

Specifies the boot image property on which you want to apply the sort and filter conditions.

BI-IMAGE-NAME

Refers to the boot image name.

BI-TYPE-INDICATOR

Refers to the boot image type.

BI-PLATFORM

Refers to the boot image operating system.

BI-REGISTERED-DATETIME

Refers to the date and time of boot image registration.

BI-SD-PACKAGE-NAME

Refers to the software delivery package associated with the boot image.

BI-SD-PACKAGE-VERSION

Refers to the version of the software delivery package associated with the boot image.

BI-SD-PACKAGE-COMMENT

Refers to the comment included in the software delivery package associated with the boot image.

BI-IMAGE-FLAGS

Refers to the FIPS image flag associated with the boot image.

computerClassID

Note: For the latest list of supported operating systems, visit CA Support.

The possible value for computerClassID is one of the following:

4	Windows Server 2003 Web
5	Windows Server 2003 Ent
6	Windows Server 2003 DC
8	Windows 2000 Professional
9	Windows 2000 Advanced Server
10	Windows 2000 DC Server
11	Windows 2000 Server
13	Windows 95
14	Windows 98
15	Windows 98 SE

16	Windows ME
18	Windows NT Workstation 3.51
19	Windows NT workstation 4.0
20	Windows NT Server 4.0
21	Windows NT Server 4.0 Enterprise
23	Windows XP Home
24	Windows XP Professional
25	Windows XP Media Center Edition
26	Windows XP Tablet PC Edition
27	Windows XP Professional x64 Edition
29	Windows 3.0
30	Windows 3.1
31	Windows 3.11
33	Windows CE 2.00
34	Windows CE 2.01
35	Windows CE 2.11
36	Windows CE 3.00
37	Windows Mobile 2003
38	Palm OS
39	Symbian
41	DG_UX
44	DEC 3.0
45	DEC 3.2
46	DEC 4.0
48	FUJIUxp
50	HPUX 9
53	ICLUnix
75	NCR SV
76	NCR SV 2x

77	NCR SV 3x
78	NCR SST
79	NCR SST S4
80	NCR SST S4I
82	AIX 3.2
83	AIX 4.1
84	AIX 4.2
85	AIX 4.3
86	AIX 5.0
87	AIX 5.1
88	AIX 5.2
90	Unixware 2.0
91	Unixware 2.1
93	SCO OpenServer 3.2
94	SCO OpenServer 5.2
96	Sinix 5.42
97	Sinix 5.43
98	Sinix 5.44
99	Sinix 5.45
101	Irix 6.2
102	Irix 6.3
103	Irix 6.4
104	Irix 6.5
106	Dynix 4.2.x
107	Dynix 4.4.x
109	Solaris 2.3
110	Solaris 2.4
111	Solaris 2.5
112	Solaris 2.6
113	Solaris 7
114	Solaris 8
115	Solaris 9

116	SunOS 4.1.4
117	OS/2
118	AS400
119	OS390
120	DOS
122	Macintosh 10.0
123	Macintosh 10.1
124	Macintosh 10.2
125	Macintosh 10.3
126	Macintosh 10.4
129	OpenVMS 5.5
130	OpenVMS 6.0
131	OpenVMS 6.1
132	OpenVMS 6.2
133	OpenVMS 7.0
134	OpenVMS 7.1
135	OpenVMS 7.2
136	Windows NT Server 3.51
137	Windows XP Embedded
138	Windows Server 2003 Standard
139	Windows Small Business Server 2003
140	Windows Server 2003 Enterprise x64
141	Windows Server 2003 Enterprise 64-Bit Itanium

202	Caldera Linux 3.0
203	Caldera Linux 3.1
204	SCO Linux 4.0
206	Connectiva Linux 8
207	Connectiva Linux 9
208	Connectiva Linux 10
210	Debian GNU/Linux 2.0
211	Debian GNU/Linux 2.1
212	Debian GNU/Linux 2.2
213	Debian GNU/Linux 3.0
215	Fedora Core 1
216	Fedora Core 2
218	Miracle Linux 2.1
219	Miracle Linux 3.0
221	Red Flag Linux 3.0
222	Red Flag Linux 3.2
223	Red Flag Linux 4.0
224	Red Flag Linux 4.1
226	Red Hat Linux 7.0
227	Red Hat Linux 7.1
228	Red Hat Linux 7.2
229	Red Hat Linux 7.3
230	Red Hat Linux 8.0
231	Red Hat Enterprise Linux ES 2.1

232	Red Hat Enterprise Linux AS 2.1
233	Red Hat Enterprise Linux WS 2.1
234	Red Hat Enterprise Linux ES 3
235	Red Hat Enterprise Linux AS 3
236	Red Hat Enterprise Linux WS 3
237	Red Hat Enterprise Linux ES 4.0
238	Red Hat Enterprise Linux AS 4.0
239	Red Hat Enterprise Linux WS 4.0
240	Red Hat Desktop
242	Sun Java Desktop System 2003
243	Sun Java Desktop System 2
245	SuSE Linux 7.0
246	SuSE Linux 7.1
247	SuSE Linux 7.3
248	SuSE Linux 8.0
249	SuSE Linux 8.1
250	SuSE Linux 8.2
251	SuSE Linux Professional 9
252	SuSE Linux Professional 9.1
253	SuSE Linux Personal 9
254	SuSE Linux Personal 9.1
255	SuSE Linux Enterprise Server 7
256	SuSE Linux Enterprise Server 8
257	SuSE Linux Enterprise Server 9
258	SuSE Standard Server 8
259	SuSE Desktop 1
260	SuSE Desktop 2
262	TurboLinux Server 8
263	TurboLinux Workstation 8
264	TurboLinux Enterprise 8
265	TurboLinux 10
267	Xandros Desktop 1.0
268	Xandros Desktop 1.1

269	Xandros Desktop 2.0
270	OpenVMS 7.3
271	Unixware 7.1.4
273	SCO OpenServer 5.7
276	Nokia AdminSuite 1.0
277	OpenVMS 8.0
278	OpenVMS 8.2
279	Solaris 10
280	AIX 5.3
281	NCR MP-RAS 3.02
282	HPUX 11.10
283	HPUX 11.22
284	HPUX 11.23
285	HPUX 11.00
286	HP Tru64 5.1
287	HP Tru64 5.0
288	HP Tru64 4.0F
289	HP Tru64 4.0G
290	HPUX 10.00
291	HPUX 10.10
292	HPUX 10.20
293	HPUX 10.30
294	HPUX 11.20
298	Windows Vista Business

299	Windows Vista Enterprise
300	Windows Vista Home Premium
301	Windows Vista Home Basic
302	Windows Vista Ultimate
303	Windows Vista Business x64 Edition
304	Windows Vista Enterprise x64 Edition
305	Windows Vista Home Premium x64 Edition
306	Windows Vista Home Basic x64 Edition
307	Windows Vista Ultimate x64 Edition
309	Windows Mobile 5.0
5029	Macintosh 10.x or later
5082	SuSE Linux Enterprise Server 10
5083	SuSE Linux Enterprise Server 10.1
5084	SuSE Linux Enterprise Desktop 10
5085	SuSE Linux Enterprise Desktop 10.1
5088	Windows Server 2003 Standard x64
5089	Windows Server 2003 Datacenter x64
5090	Unixware 7.1.3
5091	Macintosh 10.5
5092	Red Hat Enterprise Linux Server 5
5093	Red Hat Enterprise Linux Desktop 5
5094	Red Hat Enterprise Linux 5 Advanced Platform
5096	Sun Servers
5097	Sun Fire E25K
5098	Sun Fire E20K
5099	Sun Fire 15K
5100	Sun Fire 12K
5101	Sun Enterprise 10000
5102	Sun Enterprise 6500
5103	Sun Enterprise 5500
5104	Sun Enterprise 4500
5105	Sun Enterprise 3500

5107	Windows Embedded for Point of Service Version 1
5110	Windows Embedded for Point of Service 1.0
5111	HPUX 11.11
5112	HPUX 11.31
5115	Sun RTOS
5116	Red Hat Enterprise Linux Server 5.1
5117	Red Hat Enterprise Linux Desktop 5.1
5118	Red Hat Enterprise Linux 5.1 Advanced Platform
5120	Windows Server 2008 Standard Edition
5121	Windows Server 2008 Standard Edition x64 Edition
5122	Windows Server 2008 Enterprise Edition
5123	Windows Server 2008 Enterprise Edition x64 Edition
5124	Windows Server 2008 Datacenter Edition
5125	Windows Server 2008 Datacenter Edition x64 Edition
5126	Windows Web Server 2008

5127	Windows Web Server 2008 x64 Edition
5128	Windows Storage Server 2008
5129	Windows Storage Server 2008 x64 Edition
5130	Windows Small Business Server 2008
5131	Windows Essential Business Server 2008
5132	Windows Server 2008 IA-64 Edition
5137	Windows Mobile 6.0
5139	AIX 6.1
5148	Windows Server 2008 Hyper-V
5149	VMWare ESX Server 3.5
5150	VMWare ESXi 3.5
5153	Virtual Host Machine
5154	HP Chassis
5155	IBM Server
5157	Windows 7 Enterprise
5158	Windows 7 Enterprise x64 Edition
5159	Windows 7 Home Premium
5160	Windows 7 Home Premium x64 Edition
5161	Windows 7 Professional
5162	Windows 7 Professional x64 Edition
5163	Windows 7 Starter

5164	Windows 7 Home Basic
5165	Windows 7 Home Basic x64 Edition
5166	Windows 7 Ultimate
5167	Windows 7 Ultimate x64 Edition
5168	Macintosh 10.6
5169	SUSE Linux Enterprise Server 11
5189	Windows Server 2008 R2 Datacenter x64 Edition
5190	Windows Server 2008 R2 Enterprise x64 Edition
5191	Windows Server 2008 R2 IA-64 Edition
5192	Windows Server 2008 R2 Standard x64 Edition
5193	Windows Server 2008 Server Core Standard Edition
5194	Windows Server 2008 Server Core Standard x64 Edition
5195	Windows Server 2008 R2 Server Core Standard x64 Edition
5196	Windows Server 2008 Server Core Enterprise Edition
5197	Windows Server 2008 Server Core Enterprise x64 Edition
5198	Windows Server 2008 R2 Server Core Enterprise x64 Edition
5199	Windows Server 2008 Server Core Datacenter Edition
5200	Windows Server 2008 Server Core Datacenter x64 Edition
5201	Windows Server 2008 R2 Server Core Datacenter x64 Edition
5202	Windows Web Server 2008 R2 x64 Edition
5203	Windows Web Server 2008 Server Core Edition
5204	Windows Web Server 2008 Server Core x64 Edition
5205	SuSE Linux Enterprise Desktop 11
5210	Windows Server 2008 Server Core Installation

5211	Windows Server 2008 (x64) Server Core Installation
5221	openSUSE 11.0
5222	openSUSE 11.1
5223	Sun eXtended System Control Facility (XSCF)
5224	Windows Mobile 6.1
5226	VMWare ESX Server 4.0
5229	Windows Embedded POSReady 2009
5230	Windows Storage Server 2008 Standard Edition
5231	Windows Storage Server 2008 Basic Edition
5232	Windows Storage Server 2008 Enterprise Edition
5233	Windows Storage Server 2008 Workgroup Edition
5234	Windows Storage Server 2008 Standard X64 Edition
5235	Windows Storage Server 2008 Basic X64 Edition
5236	Windows Storage Server 2008 Enterprise X64 Edition
5237	Windows Storage Server 2008 Workgroup X64 Edition
5238	VMWare ESX Server 4.1
5239	Windows Mobile 6.5
5240	VMware ESXi 4.0
5241	VMware ESXi 4.1
5243	Citrix XenServer 4.0
5244	Citrix XenServer 4.1
5245	Citrix XenServer 5.0
5246	Citrix XenServer 5.5
5247	Citrix XenServer 5.6
5248	Windows Server 2008 R2 Hyper-V
5249	Hyper-V Server 2008
5250	Hyper-V Server 2008 R2
5254	Windows Embedded Standard 7 Edition
5255	Windows Embedded Standard 7 X64 Edition
5257	AIX 7.1
5258	Red Hat Enterprise Linux Server 6.0 x86

5259	Red Hat Enterprise Linux Server 6.0 x86_64
5260	Debian GNU/Linux 5.0
5261	Debian GNU/Linux 6.0
5262	Debian GNU/Linux 7.0
5264	Ubuntu Linux 10.04
5265	Ubuntu Linux 10.10
5266	Ubuntu Linux 11.04
5267	Ubuntu Linux 11.10
5268	Ubuntu Linux 12.04
5269	Macintosh 10.7
5270	Red Hat Enterprise Linux Server 6.1 x86
5271	Red Hat Enterprise Linux Server 6.1 x86_64
5272	openSUSE 11.2 x86
5273	openSUSE 11.2 x86_64
5274	openSUSE 11.3 x86
5275	openSUSE 11.3 x86_64
5276	openSUSE 11.4 x86
5277	openSUSE 11.4 x86_64
5278	Red Hat Enterprise Linux Server 6.2 x86
5279	Red Hat Enterprise Linux Server 6.2 x86_64
5296	Windows CE 6.00
5297	Windows CE 5.00
5298	Windows CE .NET 4.0
5299	Windows CE .NET 4.1
5300	Windows CE .NET 4.2
5302	Oracle Enterprise Linux Server 5 x86
5303	Oracle Enterprise Linux Server 5 x86_64

5304	Oracle Linux Server 6 x86
5305	Oracle Linux Server 6 x86_64
5309	Red Hat Enterprise Linux Server 6.3 x86
5310	Red Hat Enterprise Linux Server 6.3 x86_64
5311	Windows Embedded POSReady 7
5313	Windows 8 Pro x86
5314	Windows 8 Pro x64
5315	Windows 8 Enterprise x86
5316	Windows 8 Enterprise x64
5317	Windows Embedded Standard 8 x86
5318	Windows Embedded Standard 8 x64
5319	Windows 8 x86
5320	Windows 8 x64
5322	Windows Server 2012 Foundation x64
5323	Windows Server 2012 Essentials x64
5324	Windows Server 2012 Standard x64
5325	Windows Server 2012 Datacenter x64
5332	Windows Storage Server 2012 Workgroup x64
5333	Windows Storage Server 2012 Standard x64
5334	Solaris 11
5337	Macintosh 10.8
5338	VMware ESXi 5.1
5339	Red Hat Enterprise Linux 5.5 Advanced Platform
5340	Red Hat Enterprise Linux 5.6 Advanced Platform
5341	Red Hat Enterprise Linux 5.7 Advanced Platform
5342	Red Hat Enterprise Linux 5.8 Advanced Platform
5343	Red Hat Enterprise Linux 5.9 Advanced Platform

5344	Red Hat Enterprise Linux Desktop 5.5
5345	Red Hat Enterprise Linux Desktop 5.6
5346	Red Hat Enterprise Linux Desktop 5.7
5347	Red Hat Enterprise Linux Desktop 5.8
5348	Red Hat Enterprise Linux Desktop 5.9
5349	Red Hat Enterprise Linux Server 5.5
5350	Red Hat Enterprise Linux Server 5.6
5351	Red Hat Enterprise Linux Server 5.7
5352	Red Hat Enterprise Linux Server 5.8
5353	Red Hat Enterprise Linux Server 5.9
5354	Red Hat Enterprise Linux Server 6.4 x86
5355	Red Hat Enterprise Linux Server 6.4 x86_64
5356	Hyper-V Server 2012
5357	Windows Server 2012 Hyper-V
5402	openSUSE Linux 12
5403	Citrix XenServer 6.0
5405	Citrix XenServer 6.1

PackageFormat

An enumerated list of possible software package formats. You can select one of the following xsd:string types:

REGULAR

A software package that contains a regular application

VIRTUAL

A software package that contains a virtual application

SoftwarePackageProperty

Specifies the software package properties on which you want to apply the sort and filter conditions.

SDPKG-NAME

Refers to the name of the software package.

SDPKG-VERSION

Refers to the version of the software package.

SDPKG-TYPE

Refers to the type of software package.

SDPKG-BASEDONPKG-NAME

Refers to the name of the software package based on which this software package is created or from which this software package is imported. A software package can be created or imported based on another software package.

SDPKG-BASEDONPKG-VERSION

Refers to the version of the software package based on which this software package is created or from which this software package is imported. A software package can be created or imported based on another software package.

SDPKG-REGISTERED-DATETIME

Refers to the date and time the software package was created.

SDPKG-FORMAT

Refers to the package format.

SDPKG-SIZE

Refers to the package size.

SDPKG-CHANGED-DATETIME

Refers to the date and time the software package was last changed.

SDPKG-SUPPLIER

Refers to the vendor of the software package.

SDPKG-FILEDBY

Refers to the user who created the software package.

SDPKG-CHECKSUM

Refers to the software package checksum.

SDPKG-USERSELECTABLE

Specifies whether the package is selectable from the SD Catalog.

SDPKG-COMMENT

Refers to the comment for the software package.

SDPKG-ARCHIVED-DATETIME

Specifies the date and time when the package was archived.

SDPKG-ARCHIVEDBY

Refers to the user who performed the archive.

SDPKG-ARCHIVED-AT

Specifies whether the archived path is on the manager or on the DSM Web Services computer.

SDPKG-ARCHIVED-PATH

Refers to the archive path of the package.

SDPKG-ARCHIVED-COMMENT

Refers to the archive comment.

ComputerProperty

An enumerated list of possible computer properties. You can select one of the following xsd:string types:

- COMPUTER-UUID
- COMPUTER-LABEL
- COMPUTER-PRIMARY-NETWORK-ADDRESS
- COMPUTER-HOST-NAME
- COMPUTER-SERIAL-NUMBER
- COMPUTER-DISC-SERIAL-NUMBER
- COMPUTER-PRIMARY-MAC-ADDRESS
- COMPUTER-ASSET-TAG

- COMPUTER-CLASSID
- COMPUTER-USAGE-LIST
- COMPUTER-URI
- COMPUTER-SYSTEM-ID
- COMPUTER-VENDOR-NAME
- COMPUTER-DOMAIN-LABEL
- COMPUTER-CREATION-USER
- COMPUTER-CREATION-DATE
- COMPUTER-LAST-UPDATED-USER
- COMPUTER-LAST-UPDATED-DATE
- COMPUTER-VERSION-NUMBER

ComputerStatusInfo

xsd:string computerName

Name of the computer

xsd:string computerType

Type of the computer (OS type)

xsd:string computerDescription

Description of the computer

xsd:string IPAddress

Ip Address of the computer.

dsm:DeploymentStatusCode computerStatusCode;

Status code returned during scan or deployment.

xsd:int computerStatusCodeEx

Extended Status code. Contains the percentage of package transfer during deployment.

xsd:string computerStatusMsg

Status message if any.

xsd:bool bDmPrimerExists

DmPrimer Exists or not.

xsd:bool bCredentialsNecessary

Returns whether credentials are required or not for deployment.

CompRequiresAuthentication

CO_REQUIRES_AUTHENTICATION

Specifies that the common object (CO) component requires authentication.

AM_REQUIRES_AUTHENTICATION

Specifies that the asset management (AM) component requires authentication.

SD_REQUIRES_AUTHENTICATION

Specifies that the software delivery (SD) component requires authentication.

DeploymentStatusCode

MACHINE_UP

The computer has responded.

READY_TO_DEPLOY

Computer has responded; the package is not already installed in any version. Check the credentials field to establish the need for a username and password

PACKAGE_DEPLOYED

Computer has responded; the package at the same version is already installed.

PACKAGE_OLDER

Computer has responded; the package at a previous version is already installed.

PACKAGE_NEWER

Computer has responded; the package at a later version is already installed.

PACKAGE_DEPLOYING

Computer has responded; the package is already being installed.

PACKAGE_POSTED

Package has being submitted for deployment to the computer. The computerStatusCodeEx field contains the time (in seconds) since the package was posted.

PACKAGE_IN_TRANSIT

Package is being transmitted to the computer. The computerStatusCodeEx contains the percentage of the file that has been transmitted.

PACKAGE_DELIVERED

Package has arrived at the computer. The computerStatusCodeEx field contains the time (in seconds) since the package was posted.

PACKAGE_INSTALLING

Package has started to install. The computerStatusCodeEx field contains the time (in seconds) since the package was posted.

PACKAGE_INSTALLED

Package has successfully installed. The computerStatusCodeEx field contains the time (in seconds) since the package was posted.

PRIMER_DEPLOYING

The primer is being deployed.

PRIMER_INSTALLED

The primer has successfully installed.

MACHINE_DOWN

The computer has not responded.

DELIVERY_FAILED

The package has failed to be delivered because of an internal error. The computerStatusCodeEx field contains a further code elucidating the reason and the computerStatusMsg contains a textual reason.

INSTALL_FAILED

The package has failed to install because of an internal error. The computerStatusCodeEx field contains a further code elucidating the reason and the computerStatusMsg contains a textual reason.

PRIMER_FAILED

The primer has failed to install.

PACKAGE_INCOMPATIBLE

Package incompatible with target platform.

UNKNOWN_TARGET

Target was not resolved properly.

NO_DOMAIN_SCANNING

Domain scanning not supported by the manager. Unix managers do not support domain scanning.

PACKAGE_STAGED

Package is already staged. Returned when scanning for staged packages on scalability servers.

UNKNOWN_STATUS

Unknown status.

DiscoveredSoftwareSourceType

An enumerated list of possible source types that indicate how a discovered software item was found. You can select one of the following xsd:string types:

- SOURCE-TYPE-CA
- SOURCE-TYPE-CUSTOM
- SOURCE-TYPE-HEURISTIC

DiscoveredSoftwareType

An enumerated list of the possible types for a discovered software item. You can select one of the following xsd:string types:

- SOFTWARE-TYPE-UNDEFINED
- SOFTWARE-TYPE-SOFTWARE
- SOFTWARE-TYPE-RELEASE
- SOFTWARE-TYPE-PATCH
- SOFTWARE-TYPE-CONFIGURATION-SETTING
- SOFTWARE-TYPE-SERVICE
- SOFTWARE-TYPE-PRODUCT

DistributionContainerOrderType

Type of the DistributionContainerOrder. Members are:

xsd:int REGISTER_PACKAGE

xsd:int Deregister_PACKAGE

xsd:int REGISTER_PROCEDURE

xsd:int Deregister_PROCEDURE

xsd:int FETCH_ITEM

xsd:int REGISTER_DIFF_PACKAGE
xsd:int REGISTER_SOFTWARE_POLICY
xsd:int Deregister_SOFTWARE_POLICY
xsd:int REGISTER_JOB_CONTAINER
xsd:int Deregister_JOB_CONTAINER
xsd:int ARCHIVE_PACKAGE
xsd:int RESTORE_PACKAGE

DistributionOrderStatus

An overall order status. Members are:

DIST_ORDER_OK

DIST_ORDER_NO_STATUS

This means that no orders have been executed. The container is probably still transferring or the transfer has failed or is halted.

DIST_ORDER_ERROR

DistributionStatus

DIST_OK,
DIST_WAITING,
DIST_ACTIVE,
DIST_ERRORED,
DIST_HALTINGPROGRESS,
DIST_HALTED,
DIST_TERMINATING,
DIST_BUILDING

DomainType

WS_ENTERPRISE

Represents the enterprise.

WS_DOMAIN

Represents the domain.

DomainProperty

An enumerated list of possible properties of domains. The list encompasses properties from user accounts, user profiles, and domains. You can select one of the following types

DOMAIN_UUID

Represents the domain UUID property.

DOMAIN_PARENT_UUID

Represents the parent domain UUID property.

DOMAIN_LABEL

Represents the domain label property.

DOMAIN_DESCR

Represents the domain description property.

DOMAIN_CREATION_USER

Represents the domain created user property.

DOMAIN_LAST_UPDATED_USER

Represents the last updated user property.

DOMAIN_CONTACT_INFO

Represents the domain contact information property.

DOMAIN_DIS_HW_UUID

Represents the domain discovered UUID property.

DOMAIN_DB_HOST_NAME

Represents the domain database host name property.

DOMAIN_DB_SERVER

Represents the domain database server property.

DOMAIN_DB_TYPE

Represents the domain database type property.

DOMAIN_DB_NAME

Represents the domain database name property.

DOMAIN_DB_USER_NAME

Represents the domain database user name property.

DOMAIN_DB_INSTANCE

Represents the domain database instance property.

DOMAIN_USAGE_LIST_MASK

Represents the domain usage list mask property.

DOMAIN_NUMBER

Represents the domain NUMBER property.

DOMAIN_LAST_UPDATED_DATE

Represents the domain last updated property.

DOMAIN_TYPE

Represents the domain type property.

DOMAIN_CREATION_DATE

Represents the domain creation date property.

DomainPropertyFilter

DomainProperty domainProperty

The identifier for the domain property on which the search is based.

xsd:string searchString

If the property to search on is a string property, this parameter specifies the value of the search filter to be applied.

This field is mandatory if domainProperty is set to one of the following values:

DP_DOMAIN_UUID,
DP_PARENT_DOMAIN_UUID
DP_DOMAIN_LABEL
DP_DOMAIN_DESCR
DP_CREATION_USER
DP_LAST_UPDATED_USER
DP_CONTACT_INFO
DP_DIS_HW_UUID
DP_DB_HOST_NAME
DP_DB_SERVER
DP_DB_TYPE
DP_DB_NAME
DP_DB_USER_NAME
DP_DB_INSTANCE

xsd:long searchValue

If the property to search on is a numeric property, this parameter specifies the value of the search filter to be applied. This field is mandatory if domainProperty is set to one of the following values:

DP_USAGE_LIST_MASK

DP_DOMAIN_NUMBER

DP_LAST_UPDATED_DATE

DP_DOMAIN_TYPE

DP_CREATION_DATE

DownloadMethod

INTERNAL_NOS

Download method for software delivery is internal NOS.

INTERNAL_NOSLESS

Download method for software delivery is Internal NOSLESS

DTS_NOSLESS

Download method for software delivery is DTS.

ElementType

An enumerated list of possible element types. You can select one of the following xsd:string types:

- DSM-FILE
- DSM-DIR

GroupFlag

An enumerated list of possible group types to indicate whether a group is local or global in nature. You can select one of the following xsd:string types:

- LOCAL-GROUP
- GLOBAL-GROUP

GroupInheritance

An enumerated list of options to indicate the nature of permissions inheritance within a group. You can select one of the following xsd:string types:

- FLAT
- SINGLE-PARENT
- MULTIPLE-PARENT

GroupSystem

An enumerated list of the CA ITCM system-defined unit groups. You can select one of the following xsd:string types:

- UNCLASSIFIED-SYSTEM-GROUP
- USER-DEFINED-GROUP
- DIRECTORY-GROUP
- ALL-COMPUTERS
- ALL-USERACCOUNTS
- ALL-USERPROFILES
- ALL-USERPROFILE-GROUPS
- ALL-DOMAINS
- ALL-QUERIES
- ALL-JOBS
- ALL-EVENT-LOGS
- ALL-SERVERS
- ALL-SERVER-GROUPS
- ALL-DOMAIN-GROUPS
- ALL_LEGACY_COMPUTERS
- ALL_LEGACY_USERS
- ALL_EXTERNAL_ASSETS
- ALL_BACKUP_USERS
- ALL_BACKUP_JOBS
- ALL_BACKUP_RESTORE_JOBS

Inheritance Level

SEC_INHERIT_CLASS
SEC_INHERIT_GROUP
SEC_INHERIT_OBJECT
SEC_INHERIT_REVERT

LinkedObjectTypeRequired

CONSTRAINT_NONE

No related objects to unit group are listed

CONSTRAINT_ALL

All related objects to unit group are listed

operatingSystemType

The following table contains the operating systems and their corresponding ClassID's:

Operating System	ClassID
Aix	0x090000
Aix_3_2_Rs_6000	0x090101
Aix_4_1_Rs_6000	0x090201
Aix_4_2_Rs_6000	0x090202
Aix_4_3_Rs_6000	0x090203
Aix_5_0_Rs_6000	0x090301
Aix_5_1_Rs_6000	0x090302
Aix_5_2_Rs_6000	0x090303
Aix_5_3_Rs_6000	0x090304
Aix_5_4_Rs_6000	0x090305
Aix_5_5_Rs_6000	0x090306
Aix_5_6_Rs_6000	0x090307
Aix_5_7_Rs_6000	0x090308
Aix_5_8_Rs_6000	0x090309
Any	0x030000

Compaq_True64_5_X_Alpha	0x0d0401
Compaq_True64_Dec	0x0d0000
Dec_3_0_Alpha	0x0d0101
Dec_3_2_Alpha	0x0d0201
Dec_4_0_Alpha	0x0d0301
Dgux	0x0a0000
Dgux_4_1_Intel	0x0a0201
Dgux_4_1_Mc88	0x0a0101
Dgux_4_2_Intel	0x0a0301
Dos	0x010000
Drs_Nx_7_Sparc	0x160101
Drs_Nx_7mp_Sparc	0x160102
Drs_Nx_Sparc	0x160000
Dynix	0x120000
Dynix_4_2_X_Intel	0x120101
Dynix_4_4_X_Intel	0x120201
Hpux	0x070000
Hpux_10	0x070200
Hpux_10_700	0x070201
Hpux_10_800	0x070202
Hpux_11	0x070300
Hpux_11_700	0x070301
Hpux_11_800	0x070302
Hpux_11_ia64	0x070303
Hpux_11_iv2	0x070304
Hpux_12	0x070400
Hpux_12_700	0x070401
Hpux_12_800	0x070402
Hpux_12_ia64	0x070403
Hpux_12_iv2	0x070404
Hpux_9	0x070100
Hpux_9_700	0x070101

Hpux_9_800	0x070102
Irix	0x130000
Irix_6_2	0x130101
Irix_6_3	0x130301
Irix_6_4	0x130201
Irix_6_5	0x130401
Linux	0x170000
Linux_2_1_Intel	0x170101
Linux_2_2_Intel	0x170201
Linux_2_2_S_390	0x170202
Linux_2_3_Intel	0x170301
Linux_2_3_S_390	0x170302
Linux_2_4_Intel	0x170401
Linux_2_4_S_390	0x170402
Linux_2_5_Intel	0x170501
Linux_2_5_S_390	0x170502
Linux_2_6_Intel	0x170601
Linux_2_6_Intel_64	0x170603
Linux_2_6_S_390	0x170602
Macintosh	0x1b0000
Macintosh_10_0_Ppc	0x1b0101
Macintosh_10_1_Ppc	0x1b0201
Macintosh_10_2_Ppc	0x1b0301
Macintosh_10_3_Ppc	0x1b0401
Macintosh_10_4_Ppc	0x1b0501
Ncr_Sst	0x150000
Ncr_Sst_S4_Intel	0x150101
Ncr_Sst_S4i_Intel	0x150201
Ncr_Sv	0x0c0000
Ncr_Sv_2x_Intel	0x0c0101
Ncr_Sv_3x_Intel	0x0c0201
Netware	0x0b0000

Netware_3_Intel	0x0b0101
Netware_4_Intel	0x0b0201
Netware_5_Intel	0x0b0301
Netware_6_Intel	0x0b0401
Nokia_Adminsuite	0x1a0000
Nokia_Adminsuite_1_0	0x1a0001
Openvms	0x0e0000
Openvms_5_5_Vax	0x0e0101
Openvms_6_0_Alpha	0x0e0202
Openvms_6_0_Vax	0x0e0201
Openvms_6_1_Alpha	0x0e0302
Openvms_6_1_Vax	0x0e0301
Openvms_6_2_Alpha	0x0e0402
Openvms_6_2_Vax	0x0e0401
Openvms_7_0_Alpha	0x0e0502
Openvms_7_0_Vax	0x0e0501
Openvms_7_1_Alpha	0x0e0602
Openvms_7_1_Vax	0x0e0601
Openvms_7_2_Alpha	0x0e0702
Openvms_7_2_Vax	0x0e0701
Os2_16	0x000000
Os2_32	0x040000
Palm_Os	0x190000
Sco	0x0f0000
Sco_3_2_V5_Intel	0x0f0101
Sco_5_2_V5_Intel	0x0f0201
Sinix	0x100000
Sinix_5_42_R4000	0x100101
Sinix_5_43_R4000	0x100201
Sinix_5_44_R4000	0x100301
Sinix_5_45_R4000	0x100401
Solaris	0x080000

Solaris_10_Intel	0x080802
Solaris_10_Sparc	0x080801
Solaris_2_3_Sparc	0x080101
Solaris_2_4_Intel	0x080202
Solaris_2_4_Sparc	0x080201
Solaris_2_5_Intel	0x080302
Solaris_2_5_Sparc	0x080301
Solaris_2_6_Intel	0x080402
Solaris_2_6_Sparc	0x080401
Solaris_7_Intel	0x080502
Solaris_7_Sparc	0x080501
Solaris_8_Intel	0x080602
Solaris_8_Sparc	0x080601
Solaris_9_Intel	0x080702
Solaris_9_Sparc	0x080701
Sunos	0x110000
Sunos_4_1_4_Sparc	0x110101
Unix	0x060000
Unixware	0x140000
Unixware2_0_Intel	0x140101
Unixware2_1_Intel	0x140201
Unixware7_X_Intel	0x140301
Win_2000	0x050100
Win_2000_Intel	0x050101
Win_Ce	0x180000
Win_Ce_2_00_Mips	0x180101
Win_Ce_2_00_Sh3	0x180102
Win_Ce_2_01_Mips	0x180201
Win_Ce_2_01_Sh3	0x180202
Win_Ce_2_11_Arm	0x180301
Win_Ce_2_11_Mips	0x180302
Win_Ce_2_11_Sh3	0x180303

Win_Ce_2_11_Sh4	0x180304
Win_Ce_3_00_Arm	0x180401
Win_Ce_3_00_Mips	0x180402
Win_Ce_3_00_Ppc2002_Arm	0x180501
Win_Ce_3_00_Sh3	0x180403
Win_Ce_3_00_Sh4	0x180404
Win_Ce_4_20_Arm	0x180601
Win_Ce_5_00_Arm	0x180701
Win_Longhorn_Intel	0x050501
Win_Longhorn_Itanium	0x050502
Win_Longhorn_x64	0x050503
Win_Nt	0x050000
Win_Nt_Alpha	0x050002
Win_Nt_Intel	0x050001
Win_Nt_Mips	0x050003
Win_Server_2003	0x050300
Win_Server_2003_Intel	0x050301
Win_Server_2003_Itanium	0x050302
Win_Server_2003_x64	0x050303
Win_Vista_Intel	0x050401
Win_Vista_Itanium	0x050402
Win_Vista_x64	0x050403
Win_Xp	0x050200
Win_Xp_Intel	0x050201
Win_Xp_Itanium	0x050202
Win_Xp_x64	0x050203
Windows_16	0x020000
Windows_32	0x020100
Windows_95	0x020101
Windows_98	0x020102
Windows_Me	0x020103

PackageExportType

SOFTWAREPACKAGE

Exports as a software package

LIBRARYIMAGE

Exports as a library image

QueryFragment

This sequence of elements represents one fragment of a CA ITCM query definition. It describes a search statement.

QueryFragmentType queryFragmentType

The type of the query fragment, is it a database or directory statement?

xsd:string pseudoStatement

A user-defined description of the nature of the query fragment's statement.

xsd:string statement

The underlying SQL-style statement from which the query fragment is formulated.

Remarks

None

QueryFragment2

This sequence of elements represents one fragment of a DSM query definition. It describes a search statement.

QueryFragmentType2 queryFragmentType

Type of the query fragment; specifies whether it is a database or directory statement.

xsd:string pseudoStatement

A user-defined description of the nature of the query fragment's statement.

xsd:string statement

The underlying SQL-style statement from which the query fragment is formulated.

xsd:string linkQueryName

Name of the existing query used to link with this new query. Existing query will be added as an argument of new query.

xsd:string attributeFilter

Used to specify the attribute values based on which the query results can be streamlined.

For example: (assetCn=z*) AND (assetDescription=This is a sample Query).

The above attributeFilter helps to find assets with name starting 'z'.

Other Attributes are as follows:

- assetCn
- assetDescription
- assetName
- c
- company
- dc
- department
- directReports
- displayName
- dnsHostName
- email
- givenName
- groupCn
- groupDescription
- groupMembers
- groupName
- jobTitle
- l
- memberOf
- objectClass
- operatingSystem
- operatingSystemServicePack
- operatingSystemVersion
- postalCode
- streetAddress
- telephoneNumber

- uniqueComputerFields
- uniqueUserFields
- userCn
- userDescription
- userID
- userName
- userPassword
- userSn.

xsd:bool attributeFilterSupplied

Set the value of this parameter to true if 'attributeFilter' is supplied otherwise set it to false.

xsd:bool searchRecursive

Set it true if recursive search is desired.

xsd:bool searchRecursiveSupplied

Set it to true if 'searchRecursive' parameter is set to true otherwise set it to false.

Remarks

The value of 'attributeFilterSupplied' can be set to true or false if value of 'queryFragmentType' is DIRECTORY_***. For all other values 'attributeFilterSupplied' should be set to false.

QueryFragmentType

An enumerated list of possible query types. You can select one of the following xsd:string types:

SQL

Database queries for the MDB

DIRECTORY

Queries for external, configured directories

QueryFragmentType2

An enumerated list of possible query types. You can select one of the following xsd:string types:

SQL_QUERY

Database queries for the MDB

LINK_QUERY

Link queries to create link queries.

DIRECTORY_OU_USERS

Active directory queries for the users in organizational units.

DIRECTORY_GROUP_USERS

Active directory queries for group.

DIRECTORY_OU_COMPUTERS

Active directory queries for the computers in organizational units.

DIRECTORY_GROUP_COMPUTERS

Active directory queries for group computers.

DIRECTORY_DIRECT_REPORTS

Active directory queries for direct reports.

DIRECTORY_LOCALGROUP_USERS

Active directory queries for local group users.

DIRECTORY_LOCALGROUP_COMPUTERS

Active directory queries for local group computers.

QueryType

An enumerated list of possible query types. You can select one of the following xsd:string types:

- AGENT
- DISCOVERED-HARDWARE
- DISCOVERED-USER
- DISCOVERED-COMPUTERUSER

RACPolicy

Specifies the type of Reinstall After Crash policy.

RAC_COMMON

The common Reinstall after Crash (RAC) policy is used for this computer. *(Default)*

RAC_AUTOMATIC

The RAC job container will automatically get created and activated. policy is used for this computer.

RAC_DEFERRED

A RAC job container is created. It can reviewed and modified, since it has to be activated manually.

RAC_DISABLED

The Reinstall after Crash (RAC) policy is disabled for this computer.

SecurityAreaLinkage

SEC_AREA_LINK

SEC_AREA_UNLINK

SecurityClassType

The following are the enumerations for the SecurityClassType, followed by its representation in the DSM Explorer GUI.

SEC_CLASS_POLICY_CONFIGURATION_COMPUTER

Policy - Configuration Computer object in DSM Explorer GUI.

SEC_CLASS_DOMAIN

Domain object in DSM Explorer GUI.

SEC_CLASS_EXTERNAL_ASSET

External Asset object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_PACKAGE

Software Package object in DSM Explorer GUI.

SEC_CLASS_REPORT_SCHEDULING

Report Scheduling object in DSM Explorer GUI.

SEC_CLASS_DOMAIN_GROUP

Domain Group object in DSM Explorer GUI.

SEC_CLASS_CLASS_PERMISSIONS

Class Permissions object in DSM Explorer GUI.

SEC_CLASS_ENGINE

Engine object in DSM Explorer GUI.

SEC_CLASS_SCALABILITY_SERVER_GROUP

Scalability Server Group object in DSM Explorer GUI.

SEC_CLASS_SECURITY_AREA

Security Area object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_JOB

Software Job object in DSM Explorer GUI.

SEC_CLASS_INVENTORY_TASK

Inventory Task object in DSM Explorer GUI.

SEC_CLASS_COMPUTER

Computer object in DSM Explorer GUI.

SEC_CLASS_POLICY_SOFTWARE_BASED

Policy - Software Based object in DSM Explorer GUI.

SEC_CLASS_REPORT_TEMPLATE

Report Template object in DSM Explorer GUI.

SEC_CLASS_COMMON_QUERY

Common Query object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_JOB_CONTAINER

Software Job Container object in DSM Explorer GUI.

SEC_CLASS_ASSET_JOB

Asset Job object in DSM Explorer GUI.

SEC_CLASS_DEPLOYMENT_JOB

Deployment Job object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_USAGE_TASK

Software Usage task object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_DISTRCONTAINER

Software Distribution Container object in DSM Explorer GUI.

SEC_CLASS_OS_INSTALLATION_IMAGE

OS Installation Image object in DSM Explorer GUI.

SEC_CLASS_REMOTE_CONTROL_ACCESS

Remote Control Access object in DSM Explorer GUI.

SEC_CLASS_ENGINE_TASK

Engine Task object in DSM Explorer GUI.

SEC_CLASS_ASSET_GROUP

Asset Group object in DSM Explorer GUI.

SEC_CLASS_USER_ACCOUNT

User Account object in DSM Explorer GUI.

SEC_CLASS_MANAGER

Manager object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_DISTRIBUTIONS

Software Distributions object in DSM Explorer GUI.

SEC_CLASS_CONTROL_PANEL_ACCESS

Control Panel Access object in DSM Explorer GUI.

SEC_CLASS_DATABASE_CREDENTIALS

Database Credentials object in DSM Explorer GUI.

SEC_CLASS_SCALABILITY_SERVER

Scalability Server object in DSM Explorer GUI.

SEC_CLASS_POLICY_EVENT_BASED

Policy - Event Based object in DSM Explorer GUI.

SEC_CLASS_SECURITY_PROFILE

Security Profile object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_CATEGORY

Software Category object in DSM Explorer GUI.

SEC_CLASS_PROCEDURE_GROUP

Procedure Group object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_DEFINITION

Software Definition object in DSM Explorer GUI.

SEC_CLASS_USER_PROFILE

User Profile object in DSM Explorer GUI.

SEC_CLASS_POLICY_QUERY_BASED

Policy - Query Based object in DSM Explorer GUI.

SEC_CLASS_PROCEDURE

Procedure object in DSM Explorer GUI.

SEC_CLASS_CONFIGURED_DIRECTORY

Configured Directory object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_GROUP

Software Group object in DSM Explorer GUI.

SEC_CLASS_SOFTWARE_DISCOVERY_TASK

Software Discovery Task object in DSM Explorer GUI.

SEC_CLASS_TEMPLATE_TASK

Template Task object in DSM Explorer GUI.

SEC_CLASS_BACKUP_USER

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_AGENT_CONFIG

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_ACT_USER_POLICY

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_SERVER

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_SERVER_CONFIG

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_JOB

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_POLICY

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_SET

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_PERFORMED_BU

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_BU_FILE_REV

No object in DSM Explorer GUI.

SEC_CLASS_BACKUP_PERFORMED_RS

No object in DSM Explorer GUI.

SecurityGroup

An enumerated list of options that indicates if a group is a security group. You can select one of the following xsd:string values:

- ORDINARY-GROUP
- SECURITY-GROUP

SecurityProfilePermission

You can select one of the following :

NOACCESS

If selected, it blocks all permissions.

VIEW

If selected, it grants only 'view' permissions.

READ

If selected, it grants "view and read" permissions.

MANAGE

If selected, it grants "read and execute permissions".

CHANGE

If selected, it grants "manager, write, and delete permissions".

FULLCONTROL

If selected, it grants "change, ownership, and create permissions".

SPECIALACCESS

If selected, it creates special permissions. It is constructed using bits selected from 'SecurityProfileSpecialPermissionFlags' structure.

SecurityProfileType

You can select one of the following:

- SEC_PROFILE_OWNER
- SEC_PROFILE_EVERYONE
- SEC_PROFILE_DISTRIBUTIONS
- SEC_PROFILE_LOCALUSER
- SEC_PROFILE_GLOBALUSER
- SEC_PROFILE_LOCALGROUP
- SEC_PROFILE_GLOBALGROUP
- SEC_PROFILE_GLOBALMACHINE
- SEC_PROFILE_SDCATALOG
- SEC_PROFILE_CERT_CSM_AGCT
- SEC_PROFILE_CERT_SERVER_REG
- SEC_PROFILE_CERT_DIR_SYNC
- SEC_PROFILE_CERT_DOMDB_PWD
- SEC_PROFILE_CERT_ENTDB_PWD
- SEC_PROFILE_CERT_REPDB_PWD
- SEC_PROFILE_CERT_SD_AGMOV
- SEC_PROFILE_CERT_SD_CATALOG
- SEC_PROFILE_CERT_BABLD_REG

SoftwareGroupType

- SW_PACKAGE_GROUP
- SW_PACKAGE_PROCEDURE_GROUP

SoftwareJobContainerScope

An enumerated list of possible software job container types.

You can select one of the following xsd:string types:

- NO-PREFERENCE
- MADE-LOCALLY-ONLY
- MADE-LOCALLY-PREFERABLY
- MADE-REMOTELY-ONLY
- MADE-REMOTELY-PREFERABLY

SoftwareJobContainerState

An enumerated list of possible states for software job container. You can select one of the following xsd:string types:

- CREATED
- WAITING
- EVALUATING
- CASCADING
- BUILDING,
- INPROGRESS
- SUCCESS
- FAILURE
- TERMINATING
- BUILD_SUCCESS
- BUILD_SUCCESS_WITH_FAILURES
- SUCCESS_WITH_WARNINGS
- DELIVERY_HALT_INPROGRESS
- DELIVERY_HALTED
- NOT_RENEWED
- PARTIALLY_RENEWED_OK

SoftwareJobRenewState

An enumerated list of possible states for renewed jobs. You can select one of the following xsd:string types:

- WAITING
- INPROGRESS
- OK
- DSM-ERROR
- NOT-RENEWED

SoftwareJobStartupTime

An enumerated list of possible startup times for a software job. You can select one of the following xsd:string types:

BOOT-LOCAL-TIME

The server will not trigger the agent to start the job. The job will be started the first time the agent starts after the start time has occurred. Normally when the agent-machine is re-booted. The start time is based on that of the domain manager.

EXACT-LOCAL-TIME

The server will trigger the agent to start the job as soon as the start time has occurred. The start time is based on that of the domain manager.

BOOT-GLOBAL-TIME

The server will not trigger the agent to start the job. The job will be started the first time the agent starts after the start time has occurred. Normally when the agent-machine is re-booted. The start time is based on that of the enterprise manager.

EXACT-GLOBAL-TIME

The server will trigger the agent to start the job as soon as the start time has occurred. The start time is based on that of the enterprise manager.

SoftwareJobState

An enumerated list of possible states in which a software job can be. You can select one of the following xsd:string types:

- WAITING
- INPROGRESS
- OK

- DSM-ERROR
- BUILDING
- REMOVING
- NOT-AVAILABLE
- CREATED
- OBSOLETE
- HALT-INPROGRESS
- HALTED

SoftwareJobTask

An enumerated list of possible software job tasks. You can select one of the following xsd:string types:

NONE

Initial transient state

INSTALL

This procedure installs its software.

UNINSTALL

This procedure uninstalls its software.

CONFIGURE

This procedure configures its software.

ACTIVATE

This procedure activates (anything else than the above) its software.

DELIVER

This procedure delivers its software to a Scalability Server Library.

UNDELIVER

This procedure removes its software from a Scalability Server Library.

SoftwarePackageClass

An enumerated list of classes to which a software package may belong. You can select one of the following xsd:string types:

- SOFTWARE
- OPEN
- DETECTED
- ARCHIVED

SoftwarePackageGroupScope

An enumerated list of possible software package group types.

You can select one of the following xsd:string types:

- NO-PREFERENCE
- MADE-LOCALLY-ONLY
- MADE-LOCALLY-PREFERABLY
- MADE-REMOTELY-ONLY
- MADE-REMOTELY-PREFERABLY

SoftwarePackageProcedureFiledBy

An enumerated list of possible entities by which a software package procedure can be filed.

You can select one of the following xsd:string types:

LOCAL

The item is filed by a local administrator

CENTRAL

The item is distributed and filed by an enterprise administrator

TD-LOCAL

A Software Delivery item that is filed by a local administrator

TDSWD

A Software Item that is detected and filed by the local Software Detector

TDSWD-CENTRAL

A Software Item that is filed at the Enterprise Manager based on a report from the Domain Manager about a locally detected software item.

TD

An Software Delivery item automatically filed at installation time

SoftwarePackageProcedureGroupScope

An enumerated list of possible software package procedure group types.

You can select one of the following xsd:string types:

- NO-PREFERENCE
- MADE-LOCALLY-ONLY
- MADE-LOCALLY-PREFERABLY
- MADE-REMOTELY-ONLY
- MADE-REMOTELY-PREFERABLY

SoftwarePackageProcedureRelation

An enumerated list of possible classes of procedures. You can select one of the following xsd:string types:

ADDED

A procedure that is an extension to the original package is said to be added if it is defined by a set of files that is not part of the original package and normally has to be added to the package.

EMBEDDED

A procedure that is defined for the original software package is an embedded procedure. It is normally defined by a set of files included in the package.

DETECTED

A procedure that is of type detected is an artificial procedure that has no files associated. The procedure is needed for database consistency reasons.

CONVERTED

A procedure that is an extension to the original package is said to be converted if it is defined by a set of files that is already included in the original package.

SoftwarePackageProcedureType

An enumerated list of possible external file types of software package procedures. You can select one of the following xsd:string types:

- SIP
- CMDFILE
- EXEFILE
- SWDET
- MSI
- SXP
- PKG
- PIF
- IPS
- PALM
- WINCE

SoftwarePackageProcedureTask

An enumerated list of possible software procedure task types. You can select one of the following xsd:string types:

- INSTALL
- UNINSTALL
- CONFIGURE
- ACTIVATE

SoftwarePackageGroupProperty

An enumerated list of possible software package group property types.

You can select one of the following xsd:string types:

NON-CATALOG

The item is not a catalog item

CATALOG

The item is a catalog item

CATALOG-GROUP

The item is a member of a catalog group

SoftwarePolicyEvaluationPeriodType

EPT_ONCE

The software policy is evaluated once.

EPT_PERIODIC

The software policy is evaluated periodically as per the evaluation period set.

SoftwareJobType

An enumerated list of types of a software job that indicates whether the job is mandatory. You can select one of the following xsd:string types:

- MANDATORY
- DSM-OPTIONAL

SoftwarePackageFiledBy

An enumerated list of possible entities by which a software package can be filed.

You can select one of the following xsd:string types:

LOCAL

The item is filed by a local administrator

CENTRAL

The item is distributed and filed by an enterprise administrator

TD-LOCAL

A Software Delivery item that is filed by a local administrator

TDSWD

A Software Item that is detected and filed by the local Software Detector

TDSWD-CENTRAL

A Software Item that is filed at the Enterprise Manager based on a report from the Domain Manager about a locally detected software item.

TD

An Software Delivery item automatically filed at installation time

SoftwarePackageType

An enumerated list of possible software package types. You can select one of the following xsd:string types:

SOFTWARE

A sealed software with its files filed in the Library

OPEN

An unsealed software with its files (if any) filed in the Library

DETECTED

A detected software which has no files in the Library

SoftwarePackagingType

An enumerated list of possible software packaging types. You can select one of the following xsd:string types:

- **GENERIC**

Note: A Generic package is one that requires no special handling by CA ITCM, unlike the other package types.

- **SXP**

- **PIF**

- **PKG**

- **MSI**

- **PALM**

- **WINCE**

- **RPM**

SoftwarePolicyEvaluationType

An enumerated list of values that indicate the type of evaluation to be used for a software policy.

You can select one of the following xsd:string types:

ET-POPULATEEXCEPTIONS

Select this option to make a conformance check for all members in the group. If any non-conforming members are found, they are added to the Violators folder.

ET-SETUPJOBS

Select this option to make a conformance check for all members in the group and to set up, but not activate, jobs for members not conforming to the policy conditions. The job container containing the jobs is not sealed.

ET-SETUPANDEXECUTEJOBS

Select this option to make a conformance check for all members in the group, and to set up, as well as activate, jobs for members not conforming to the policy conditions. The job container containing the jobs is sealed and the jobs will execute at the specified time.

SoftwarePolicyScope

An enumerated list of the possible scope of the software policy.

You can select one of the following xsd:string scopes:

- NO-PREFERENCE
- MADE-LOCALLY-ONLY
- MADE-LOCALLY-PREFERABLY
- MADE-REMOTELY-ONLY
- MADE-REMOTELY-PREFERABLY

SourceMedium

An enumerated list of possible media on which software source may be hosted. You can select one of the following xsd:string types:

- DISKETTE
- HARDDISK
- UNDEFINED
- TAPE
- CD

StringValueTypes

An enumerated list of possible string value types. You can select one of the following xsd:string types:

- NONE
- BOOLTRUEFALSE
- BOOLYESNO
- BOOLONOFF
- BOOLENABLEDISABLE
- BOOLSUPUNSUP
- BOOLACTIVENOT
- BOOLOKERROR
- BOOLPRESENTNOT
- DSM-INT32
- INT32DEC
- INT32KB
- INT32MB
- INT32GB
- INT32TB
- INT32K
- INT32M

- INT32G
- INT32MILLI
- INT32MICRO
- INT32NANO
- INT32HEX
- INT32PLATFORM
- INT32INSTALLED
- INT32TIMESTAMP
- INT32UNITTYPE
- INT32SECTORTYPE
- INT32APPDEFPLATFORM
- INT32EVENTAREA
- DSM-INT64
- INT64DEC
- INT64KB
- INT64MB
- INT64GB
- INT64TB
- INT64K
- INT64M
- INT64G
- INT64MILLI
- INT64MICRO
- INT64NANO
- INT64PERCENT
- INT64HEX
- INT64DATE
- INT64TIMESPAN
- DSM-FLOAT
- FLOATFRACPART
- FLOATVARDATE
- STRING
- STRINGVERSION

- BINUNKNOWN
- BINBITMAP
- BINICON
- BINRESOURCE
- MEMO
- DATETIME
- DSM-DATE
- TIME

SystemGroup

An enumerated list of possible CA ITCM system groups. You can select one of the following xsd:string types:

- CO-ALL-COMPUTERS
- CO-ALL-COMPUTERSUSERSGROUP
- CO-ALL-USERACCOUNTS
- CO-ALL-USERPROFILES
- CO-ALL-USERPROFILEGROUPS
- CO-ALL-DOMAINS
- CO-ALL-QUERIES
- CO-ALL-EVENTLOGS
- CO-ALL-SERVERS
- CO-ALL-SERVERGROUPS
- CO-ALL-DOMAINGROUPS
- SD-SOFTWARE-LIBRARY
- SD-SOFTWARE-LIBRARY-CATALOG
- SD-JOBS
- SD-SOFTWARE-POLICY
- SD-DELIVERY-TARGETS

UnitProperty

An enumerated list of possible properties of units. The list encompasses properties from user accounts, user profiles, and computers. You can select one of the following `xsd:string` types:

- UNIT-USERACCOUNT-NAME
- UNIT-USERACCOUNT-USAGE-LIST
- UNIT-USERACCOUNT-URI
- UNIT-USERACCOUNT-LABEL
- UNIT-USERACCOUNT-DOMAIN-LABEL
- UNIT-USERACCOUNT-CREATION-USER
- UNIT-USERACCOUNT-CREATION-DATE
- UNIT-USERACCOUNT-LAST-UPDATED-USER
- UNIT-USERACCOUNT-LAST-UPDATED-DATE
- UNIT-USERACCOUNT-VERSION-NUMBER
- UNIT-COMPUTER-LABEL
- UNIT-COMPUTER-PRIMARY-NETWORK-ADDRESS
- UNIT-COMPUTER-HOST-NAME
- UNIT-COMPUTER-SERIAL-NUMBER
- UNIT-COMPUTER-DISC-SERIAL-NUMBER
- UNIT-COMPUTER-PRIMARY-MAC-ADDRESS
- UNIT-COMPUTER-ASSET-TAG
- UNIT-COMPUTER-CLASSID
- UNIT-COMPUTER-USAGE-LIST
- UNIT-COMPUTER-URI
- UNIT-COMPUTER-SYSTEM-ID
- UNIT-COMPUTER-VENDOR-NAME
- UNIT-COMPUTER-DOMAIN-LABEL
- UNIT-COMPUTER-CREATION-USER
- UNIT-COMPUTER-CREATION-DATE
- UNIT-COMPUTER-LAST-UPDATED-USER
- UNIT-COMPUTER-LAST-UPDATED-DATE
- UNIT-COMPUTER-VERSION-NUMBER
- UNIT-USERPROFILE-LABEL

- UNIT-USERPROFILE-REFERENCE-COUNT
- UNIT-USERPROFILE-USAGE-LIST
- UNIT-USERPROFILE-CREATION-USER
- UNIT-USERPROFILE-CREATION-DATE
- UNIT-USERPROFILE-LAST-UPDATED-USER
- UNIT-USERPROFILE-LAST-UPDATED-DATE
- UNIT-USERPROFILE-VERSION-NUMBER

UnitSoftwareJobState

- An enumerated list of states in which a unit's software job may currently be. You can select one of the following xsd:string types:
- UNDEFINED
- WAITING
- DELIVERY-ORDERED
- DELIVERING
- DELIVERY-OK
- DELIVERY-ERROR
- ALREADY-DELIVERED
- EXECUTION-ORDERED
- EXECUTING
- EXECUTION-OK
- EXECUTION-ERROR
- UNDELIVERY-ORDERED
- UNDELIVERING
- UNDELIVERY-OK
- UNDELIVERY-ERROR
- ALREADY-INSTALLED
- MANIPULATION-NOT-ALLOWED
- DELIVERY-SUSPEND-ORDERED
- DELIVERY-SUSPEND-INPROGRESS
- DELIVERY-SUSPENDED
- DELIVERY-RESUME-ORDERED
- DELIVERY-RESUME-INPROGRESS
- UNDELIVERY-SUSPEND-ORDERED
- UNDELIVERY-SUSPEND-INPROGRESS
- UNDELIVERY-SUSPENDED
- UNDELIVERY-RESUME-ORDERED
- UNDELIVERY-RESUME-INPROGRESS
- EXECUTION-ORDER-STAGED

UnitType

An enumerated list of possible CA ITCM unit types. You can select one of the following xsd:string types:

- COMPUTER
- USERACCOUNT
- USERPROFILE

UserAccountProperty

An enumerated list of possible user account properties. You can select one of the following xsd:string types:

- USERACCOUNT-NAME
- USERACCOUNT-USAGE-LIST
- USERACCOUNT-URI
- USERACCOUNT-LABEL
- USERACCOUNT-DOMAIN-LABEL
- USERACCOUNT-CREATION-USER
- USERACCOUNT-CREATION-DATE
- USERACCOUNT-LAST-UPDATED-USER
- USERACCOUNT-LAST-UPDATED-DATE
- USERACCOUNT-VERSION-NUMBER

UserProfileProperty

- An enumerated list of possible user profile properties. You can select one of the following xsd:string types:
- CU-LABEL
- CU-REFERENCE-COUNT
- CU-USAGE-LIST
- CU-CREATION-USER
- CU-CREATION-DATE
- CU-LAST-UPDATED-USER
- CU-LAST-UPDATED-DATE
- CU-VERSION-NUMBER
- CU-COMPUTER-PRIMARY-NETWORK-ADDRESS
- CU-COMPUTER-HOST-NAME
- CU-COMPUTER-SERIAL-NUMBER
- CU-COMPUTER-DISC-SERIAL-NUMBER
- CU-COMPUTER-PRIMARY-MAC-ADDRESS
- CU-COMPUTER-ASSET-TAG
- CU-COMPUTER-CLASSID
- CU-COMPUTER-USAGE-LIST
- CU-COMPUTER-URI
- CU-COMPUTER-SYSTEM-ID
- CU-COMPUTER-VENDOR-NAME
- CU-COMPUTER-DOMAIN-LABEL
- CU-COMPUTER-CREATION-USER
- CU-COMPUTER-CREATION-DATE
- CU-COMPUTER-LAST-UPDATED-USER
- CU-COMPUTER-LAST-UPDATED-DATE
- CU-COMPUTER-VERSION-NUMBER

SoftwareJobLinkageType

NOLINKAGE

No job linkage exists.

TRANSACTION

The job linkage option is set to batch job execution.

SYNCHRONIZED

The job linkage option is set to synchronized job execution.

WeekDay

The enumerated list of the possible weekdays that you can specify while scheduling a job is as follows:

- WS-SUNDAY
- WS-MONDAY
- WS-TUESDAY
- WS-WEDNESDAY
- WS-THURSDAY
- WS-FRIDAY
- WS-SATURDAY

PERIODTYPE

The enumerated list of the possible period type you can specify while scheduling a job is as follows:

- WS-NONE
- WS-DAY
- WS-WEEK
- WS-MONTH
- WS-YEAR

CO_TRIGGER_UNIT_TYPE

The enumerated list of the possible unit types is as follows:

- TRIGG-COMPUTER
- TRIGG-USERPROFILE
- TRIGG-DOCKINGDEVICE

UnitObjectType

The enumerated list of the possible unit object types are as follows:

- WS-UNIT-DISCOVERED-COMPUTER
- WS-UNIT-DISCOVERED-USER
- WS-UNIT-ENGINE
- WS-SERVER
- WS-UNIT-DISCOVERED-EXTERNAL
- WS-UNIT-COMPUTER-USER

AMExtraCmdParam

The enumerated list of the possible parameters for executing asset management job check are as follows:

- WS_RESCAN_INVENTORY
- WS_RESCAN_SOFTWARE
- WS_COLLECT

CFComponentCode

The enumerated list of the possible Component codes are as follows:

- WS-SD-AGENT
- WS-AM-AGENT

WS_OSIM_DEPLOY

Specifies an enumerated list of values. The selected value indicates if the manager must wait for the deployment of the OS images.

You can use one of the following:

WS_OSIM_DO_NOT_WAIT

Specifies that the domain manager must not wait for the OS image to be deployed.

WS_OSIM_WAIT

Specifies that the domain manager must wait for the OS image to be deployed.

WS_OSIM_CONF_TYPE

Specifies the type of installation request.

You can use one of the following:

WS_OSIM_CT_PLANNED

Specifies a planned installation request.

WS_OSIM_CT_SCHEDULED

Specifies a scheduled installation request.

WS_OSIM_CT_CURRENT

Specifies a current installation request.

WS_OSIM_IR_STATUS

Specifies the status of the installation request of the OSIM computer.

The following installation requests are possible:

WS_IR_PLANNED

Specifies a planned installation request of the target computer.

WS_IR_SCHEDULED

Specifies a scheduled installation request of the target computer.

WS_OSIM_CONFIG_STATE

Specifies the configuration state of an OSIM Computer

The following states are possible:

WS_OSIM_UNMANAGED

Specifies that the OSIM computer is in an unmanaged state.

WS_OSIM_CURRENT

Specifies that the OSIM computer is in a current state.

WS_OSIM_STOPPED

Specifies that the OSIM computer is in a stopped state.

WS_OSIM_CANCEL_PEND

Specifies that the OSIM computer is in a canceled state.

WS_OSIM_ERROR

Specifies that the OSIM computer is in a failed state.

WS_OSIM_PLANNED

Specifies that the OSIM computer is in a planned state.

WS_OSIM_ACTIVATED

Specifies that the OSIM computer is in an activated state.

WS_OSIM_ANALYZING

Specifies that the OSIM computer is in an analyzing state.

WS_OSIM_PENDING

Specifies that the OSIM computer is in a pending state.

WS_OSIM_INSTALLING

Specifies that the OSIM computer is in an installing state.

Property_Type

Specifies the type of property.

The following are the property types:

COMPUTER_MACADDRESS

Specifies the type of property as MAC address.

COMPUTER_NETWORKADDRESS

Specifies the type of property as network address.

COMPUTER_HOSTNAME

Specifies the type of property as host name.

COMPUTER_SERIAL_NUMBER

Specifies the type of property as computer serial number.

FILTERCONDITION

Specifies the valid filter conditions for different properties.

You can use the following filters:

FILTER_WILDCARD_EQ

Defines the wildcard equal filter.

Note: Use this filter with xsd:string elements.

FILTER_WILDCARD_NE

Defines the wildcard not equal xxx.

Note: Use this filter with xsd:string elements.

FILTER_REGEX

Defines the regular expression filter.

Note: Use this filter with xsd:string elements.

FILTER_EQ

Defines the equal filter.

Note: Use this filter with xsd:long, dsm:DateTime, and enumerations.

FILTER_NE

Defines the not equal filter.

Note: Use this filter with xsd:long, dsm:DateTime, and enumerations.

FILTER_GT

Defines the greater than filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

FILTER_GE

Defines the greater than or equal filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

FILTER_LT

Defines lesser than filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

FILTER_LE

Defines the lesser than or equal filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

OSIMAGEPROPERTY

Specifies the OS image properties on which the sort and filter conditions can be applied.

The following are the OS image properties:

IMAGE_NAME

Specifies the name of the OS image.

TYPE_INDICATOR

Specifies the OS type.

OPERATING_SYSTEM

Specifies the operating system of the OS image.

REGISTERED_DATETIME

Specifies the registration date and time of the OS image with the domain manager.

SD_PACKAGE_NAME

Specifies the name of the software delivery package corresponding to the OS image.

SD_PACKAGE_VERSION

Specifies the version of software delivery package corresponding to the OS image.

SD_OS_CODE

Specifies the OS code of the software delivery package corresponding to the OS image.

LOCALE

Specifies the locale of the OS image.

IMAGE_FLAGS

Specifies the FIPS image flags of the OS image.

OSIMTARGETPROPERTY

Specifies the OSIM computer properties on which the sort and filter conditions can be applied.

The following are the OSIM computer properties:

TARGET_NAME

Specifies the name of the OSIM computer.

TARGET_MACADDRESS

Specifies the MAC Address of the OSIM computer.

OSIM_CONFIG_STATE

Specifies the configuration state of the OSIM computer.

STATE_CHANGED_DATETIME

Specifies the changed or activation date and time of the OSIM computer.

OS_IMAGENAME

Specifies the name of the OS image associated with the OSIM computer.

ComputerProperty2

The enumerated list of computer properties that you can use for specifying search filters is as follows:

COMPUTER_UUID

Filters based on computer UUID value.

COMPUTER_LABEL

Filters based on Computer name/label value.

COMPUTER_DOMAIN_UUID

Filters based on Computer Domain UUID value.

COMPUTER_PRIMARY_NETWORK_ADDRESS

Filters based on the Computer Primary network address value.

COMPUTER_HOST_UUID

Filters based on Computer Host UUID value.

COMPUTER_ASSET_SOURCE_UUID

Filters based on Computer Asset Source UUID value.

COMPUTER_HOST_NAME

Filters based on Computer Hostname value.

COMPUTER_SERIAL_NUMBER

Filters based on Computer Serial Number value.

COMPUTER_DISC_SERIAL_NUMBER

Filters based on Computer Disk serial number value.

COMPUTER_PRIMARY_MAC_ADDRESS

Filters based on Computer MAC Address value.

COMPUTER_ASSET_TAG

Filters based on Computer Asset tag value.

COMPUTER_URI

Filters based on Computer URI value.

COMPUTER_SYSTEM_ID

Filters based on Computer System ID value.

COMPUTER_VENDOR_NAME

Filters based on Computer Vendor Name value.

COMPUTER_DOMAIN_LABEL

Filters based on Computer Domain Label value.

COMPUTER_CREATION_USER

Filters based on Computer Creation user value.

COMPUTER_LAST_UPDATED_USER

Filters based on Computer last updated user value.

COMPUTER-AGENT-NAME

Filters based on Computer Agent Name value.

COMPUTER-AGENT-DESCRIPTIONS

Filters based on Computer Agent Descriptions value.

COMPUTER-USER-DATA1

Filters based on Computer User defined data1.

COMPUTER-USER-DATA2

Filters based on Computer User defined data2.

COMPUTER-USER-DATA3

Filters based on Computer User defined data3.

COMPUTER-USER-DATA4

Filters based on Computer User defined data4.

COMPUTER-AGENT-SS

Filters based on Agent Scalability server name

COMPUTER-AGENT-IPADDRESS

Filters based on Agent IP Address

COMPUTER_CLASSID

Filters based on Computer Operation System value.

COMPUTER_USAGE_LIST

Filters based on Computer usage list value.

COMPUTER_CREATION_DATE

Filters based on Computer Creation date time.

COMPUTER_LAST_UPDATED_DATE

Filters based on Computer last updated date time.

COMPUTER_VERSION_NUMBER

Filters based on Computer version number.

COMPUTER-AGENT-TRUSTLEVEL

Filters based on Computer Agent trust level.

COMPUTER-AGENT-CREATION-DATETIME

Filters based on Agent Creation date time.

COMPUTER-AGENT-LASTRUN-DATETIME

Filters based on Agent last run date time.

COMPUTER_AGENT_AM_STATUS

Filters based on AM Agent Status.

COMPUTER_AGENT_SD_STATUS

Filters based on SD Agent Status.

COMPUTER_AGENT_RC_STATUS

Filters based on RC Agent Status.

COMPUTER_AGENT_PT_STATUS

Filters based on Protection Agent Status.

COMPUTER_AGENT_ASP_STATUS

Filters based on Anti Spyware Agent Status.

COMPUTER_AGENT_AV_STATUS

Filters based on Antivirus Agent Status.

COMPUTER_AGENT_HIPS_STATUS

Filters based on HIPS Agent Status.

COMPUTER_AGENT_RESTRICTIONS

Filters based on Agent Restrictions.

ComputerFilterCondition

Specifies the valid filter conditions for different computer properties.

The following are the available filters:

COMPUTER_FILTER_WILDCARD_EQ

Defines the wildcard equal filter.

Note: Use this filter with xsd:string elements.

COMPUTER_FILTER_WILDCARD_NE

Defines the wildcard not equal filter.

Note: Use this filter with xsd:string elements.

COMPUTER_FILTER_EQ

Defines the equal filter.

Note: Use this filter with xsd:long, dsm:DateTime, and enumerations.

COMPUTER_FILTER_NE

Defines the not equal filter.

Note: Use this filter with xsd:long, dsm:DateTime, and enumerations.

COMPUTER_FILTER_GT

Defines the greater than filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

COMPUTER_FILTER_GE

Defines the greater than or equal filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

COMPUTER_FILTER_LT

Defines the lesser than filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

COMPUTER_FILTER_LE

Defines the lesser than or equal filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

AMAgentStatus

Indicates the AMAgent status on the target computer.

The following are the possible values:

WS_AM_AGENTLESS

Indicates that the computer is collected through the external asset functionality.

WS_AM_ALERT

Indicates that the computer has violated a policy with severity as Alert.

WS_AM_INFORMATION

Indicates that the computer has violated a policy with severity as Information.

WS_AM_WARNING

Indicates that the computer has violated a policy with severity as Warning.

WS_AM_LEGACY

Indicates that the computer has a legacy asset management agent.

WS_AM_NOTINSTALLED

Indicates that the computer does not have the AM Agent plug-in installed.

WS_AM_OPERATIONAL

Indicates that the computer is operational.

WS_AM_PREREGISTERED

Indicates that the computer is preregistered.

SDAgentStatus

Indicates the SDAgent status.

The following are the possible values:

WS_SD_LOCKED_BY_MIGRATION

Indicates that the computer is locked for migration.

WS_SD_LOCKED_BY_MOVE

Indicates that the computer is locked for move operation.

WS_SD_LOCKED_BY_RAC

Indicates that the computer is locked by RAC.

WS_SD_LOCKED_BY_RAC_AND_MOVE

Indicates that the computer is locked by RAC and Move.

WS_SD_LOCKED_BY_RAC_AND_ROAM

Indicates that the computer is locked by RAC and ROAM.

WS_SD_LOCKED_BY_ROAM

Indicates that the computer is locked by ROAM.

WS_SD_NOTINSTALLED

Indicates that the computer does not have the SD Agent plug-in installed.

WS_SD_PREREGISTERED

Indicates that the computer is preregistered.

WS_SD_OPERATIONAL

Indicates that the computer is operational.

RCAgentStatus

Indicates the RCAgent status.

The following are the possible values:

WS_RC_ACTIVE_SESSION

Indicates that the computer has an active session with a viewer.

WS_RC_ACTIVE_LOCAL_SESSION

Indicates that the computer can manage its own security settings and has an active session with a viewer.

WS_RC_LISTENING

Indicates that the computer can accept a request for a remote control session.

WS_RC_LISTENING_LOCAL_SESSION

Indicates that the computer can manage its own security settings and accept a request for a remote control session.

WS_RC_LOCKED

Indicates that the Remote Control application on the computer has been locked. Active sessions have been terminated and no new sessions can be started with the computer.

WS_RC_LOCKED_LOCAL_SESSION

Indicates that the computer can manage its own security settings but the Remote Control application on the computer has been locked. Active sessions have been terminated and no new sessions can be started with the computer.

WS_RC_MIGRATED

Indicates that the computer has migrated Remote Control agents.

WS_RC_NOT_INSTALLED

Indicates that the Remote Control agent has not been installed on the computer.

WS_RC_NOT_LISTENING

Indicates that the computer cannot accept a request for a remote control session, but the agent can be configured.

WS_RC_NOT_LISTENING_LOCAL_SESSION

Indicates that that the computer manages its own security settings and cannot accept a request for a remote control session, but the agent can be configured.

WS_RC_PREREGISTERED

Indicates that the computer is preregistered.

WS_RC_STOPPED

Indicates that that the computer has Remote Control agent installed but is not running.

WS_RC_STOPPED_LOCAL_SESSION

Indicates that that the computer has Remote Control agent installed and manages its own security settings, but remote control is not running.

WS_RC_UNKNOWN

Indicates that the computer has the RC Status as Unknown.

ProtectionAgentStatus

Indicates the ProtectionAgent status.

The following are the possible values:

WS_PT_NOTINSTALLED

Indicates that the state of all three protection applications (CA Anti-Virus, CA Anti-Spyware, and CA HIPS) is "Not Installed" or "Unknown".

WS_PT_PREREGISTERED

Indicates that the computer is preregistered.

WS_PT_NOTPROTECTED

Indicates that protection data has been retrieved and the state of at least one of the protection applications is different from "Operational" or "Not Installed."

WS_PROTECTED

Indicates that at least one protection application is installed and the individual states of all the installed applications are "Operational."

ASPAgentStatus

Indicates the Anti-Spyware agent status.

The following are the possible values:

WS_ASP_UNKNOWN

Indicates that no protection data about ASPAgent is retrieved yet or agent object is preregistered.

WS_ASP_OPERATIONAL

Indicates that ASPAgent is operational.

WS_ASP_NOTOPERATIONAL

Indicates that ASPAgent is installed but not operational.

WS_ASP_NOTINSTALLED

Indicates that ASPAgent is not installed.

AVAgentStatus

Indicates the Anti-Virus agent status.

The following are the possible values:

WS_AV_UNKNOWN

Indicates that no protection data about AVAgent is retrieved yet or agent object is preregistered.

WS_AV_OPERATIONAL

Indicates that AVAgent is operational.

WS_AV_NOTOPERATIONAL

Indicates that AVAgent is installed but not operational.

WS_AV_NOTINSTALLED

Indicates that AVAgent is not installed.

HIPSStatus

Indicates the HIPS status.

The following are the possible values:

WS_HIPS_UNKNOWN

Indicates that no protection data about HIPS agent is retrieved yet or agent object is preregistered.

WS_HIPS_OPERATIONAL

Indicates that HIPS agent is operational.

WS_HIPS_NOTOPERATIONAL

Indicates that HIPS agent is installed but not operational.

WS_HIPS_NOTINSTALLED

Indicates that HIPS agent is not installed.

AgentRestrictions

Specifies the agent restrictions.

The following are the possible values:

JOB-RESTRICTION-NONE

Specifies that there are no agent restrictions.

JOB-RESTRICTION-LOCAL

Identifies the jobs blocked locally.

JOB-RESTRICTION-CENTRAL

Identifies the jobs blocked centrally.

JOB-RESTRICTION-ALL

Identifies the jobs blocked locally and centrally.

ObjectPreference

Specifies the object type of the group you want to retrieve.

The following are the possible values:

LOCALLY

Indicates that the group is a local group.

REPLICATE

Indicates that the group is a replicated group.

NO-PREFERENCE

Indicates that there is no preference and both the groups must be retrieved.

QueryTargetType

Indicates the target type of the query.

The following are the possible values:

QUERY-TARGET-TYPE-CUSTOM

Specifies that the target type of the query is custom.

QUERY-TARGET-TYPE-COMPUTERS

Specifies that the target of the query is computers.

QUERY-TARGET-TYPE-USERACCOUNTS

Specifies that the target of the query is user accounts.

QUERY-TARGET-TYPE-USERPROFILES

Specifies that the target of the query is user profiles.

QUERY-TARGET-TYPE-OSIMCOMPUTERS

Specifies that the target of the query is OSIM computers.

QueryFilterCondition

Specifies the filter conditions for query properties.

The following are the available filters:

QUERY-FILTER-WILDCARD-EQ

Applies wildcard characters in the filter condition for string values and filters the queries that match the wildcard pattern.

QUERY-FILTER-WILDCARD-NE

Applies wildcard characters in the filter condition for string values and filters the queries that do not match the wildcard pattern.

QUERY-FILTER-EQ

Specifies a filter condition for enumerations and filters the queries that match the specified enumeration value.

QUERY-FILTER-NE

Specifies a filter condition for enumeration and filters the queries that do not match the specified enumeration value.

HWMODEL-FILTER-CONDITION

Specifies the filter conditions for hardware models.

The following are the available filters:

HWMODEL-WILDCARD-EQ

Applies wildcard characters in the filter condition for string values and filters the queries that match the wildcard pattern.

HWMODEL-WILDCARD-NE

Applies wildcard characters in the filter condition for string values and filters the queries that do not match the wildcard pattern.

COMPONENTS

Specifies the components that are supported.

- ASSET-MANAGER
- SOFTWARE-DELIVERY
- REMOTE-CONTROL
- ASSET-COLLECTOR
- SERVICE-DESK
- OSIM
- DMDEPLOY
- AUTOMATION-SERVICE

COMPONENT-STATUS

Specifies the list of status

- INSTALLED
- NOTINSTALLED
- NOTCOMPATIBLE

ConfigurationParams

Specifies the configuration parameters that are supported:

SD-MAXJOBTIMEOUT

Corresponds to the config param `"/itrm/usd/manager/MaxActivationTime"`.

SD-MINJOBTIMEOUT

Corresponds to the config param `"/itrm/usd/manager/MinActivationTime"`.

SD-JOBCLEANUPTIME

Corresponds to the config param `"/itrm/usd/manager/JobCleanupTime"`.

SD-JOBHISTORYCLEANUPTIME

Corresponds to the config param `"/itrm/usd/manager/JobHistoryCleanupTime"`.

SD-ALREADYINST-OK

Corresponds to the config param `"/itrm/usd/manager/AlreadyInstalledStatus"`.

SD-FAIL-IMPLICIT-SS-DELIVERIES

Corresponds to the config param `"/itrm/usd/manager/ FailImplicitDeliveries"`.

SD-CHECK-TEMPLATE-POLICIES

Corresponds to the config param `"/itrm/usd/manager/ CheckTemplatePolicies"`.

SD-RAC-AUTOMATION

Corresponds to the config param `"/itrm/usd/manager/ RACAutomation"`.

SD-JOBCLEANUPINTERVAL

Corresponds to the config param `"/itrm/usd/manager/ JobCleanupInterval"`.

SD-FORCED-RESTRICTIONS

Corresponds to the config param `"/itrm/usd/manager/ ForcedRestrictions"`.

WS_OSIM_CONFIG_STATE

Specifies the configuration state of an OSIM Computer

The following states are possible:

WS_OSIM_UNMANAGED

Specifies that the OSIM computer is in an unmanaged state.

WS_OSIM_CURRENT

Specifies that the OSIM computer is in a current state.

WS_OSIM_STOPPED

Specifies that the OSIM computer is in a stopped state.

WS_OSIM_CANCEL_PEND

Specifies that the OSIM computer is in a canceled state.

WS_OSIM_ERROR

Specifies that the OSIM computer is in a failed state.

WS_OSIM_PLANNED

Specifies that the OSIM computer is in a planned state.

WS_OSIM_ACTIVATED

Specifies that the OSIM computer is in an activated state.

WS_OSIM_ANALYZING

Specifies that the OSIM computer is in an analyzing state.

WS_OSIM_PENDING

Specifies that the OSIM computer is in a pending state.

WS_OSIM_INSTALLING

Specifies that the OSIM computer is in an installing state.

Enumerations for Automation Services

The following sections describe all the enumerated types (enums) that the Automation Services uses for automated OS migration functionality. These structures are a mapping of the underlying layer functionality.

AssessmentTemplateType

Specifies whether the assessment template is based on parameters or on a query.

Following are the possible values:

- PARAMETERBASED
- QUERYBASED

AssessmentTemplateParameterName

Specifies the name of the parameter used for creating the assessment template. For example, if you want to create an assessment template based on CPU speed, specify the parameter name as CPUSPEED.

Following are the possible values:

- CPUSPEED
- HDDSPACE
- RAMSIZE
- HWMODEL
- CDROM
- QUERYUUID

AssessmentTemplateParameterDataType

Specifies the data type of the parameter used for creating the assessment template.

Following are the possible values:

- INTEGER
- LONG
- BOOLEAN
- STRING

ASSESSMENTTEMPLATEPROPERTY

Specifies the properties of the assessment templates that you can use to filter and sort the list of assessment templates.

Following are the possible values:

- ASSESSMENTTEMPLATENAME
- ASSESSMENTTEMPLATELASTMODIFIEDDATE
- ASSESSMENTTEMPLATELASTMODIFIEDUSER
- ASSESSMENTTEMPLATECREATIONDATE
- ASSESSMENTTEMPLATECREATIONUSER

ASSESSMENTTASKPROPERTY

Specifies the properties of assessment tasks:

- ASSESSMENTTASKNAME
- ASSESSMENTTASKEXECUTIONFINISHDATE
- ASSESSMENTTASKEXECUTIONSTARTDATE
- ASSESSMENTTASKCREATIONDATE
- ASSESSMENTTASKCREATIONUSER
- ASSESSMENTTASKEXECUTIONUSER
- ASSESSMENTTASKEXECUTIONGROUP

AssessmentStatus

Specifies the status of assessment job.

Following are the possible values:

- STARTED
- EVALUATING
- SUCCESS
- FAILED
- NOT_STARTED

ResultGroup

Specifies the result group type.

Following are the possible values:

- WIN7_READY
- WIN7_NOT_READY
- WIN7_INSTALLED

OsimTemplateType

Specifies the type of the OSIM template.

Following are the possible values:

- GLOBAL
- GROUP
- HARDWAREMODEL

FilterCondition

Specifies the valid filter conditions for different properties.

You can use the following filters:

FILTER_WILDCARD_EQ

Defines the wildcard equal filter.

Note: Use this filter with xsd:string elements.

FILTER_WILDCARD_NE

Defines the wildcard not equal xxx.

Note: Use this filter with xsd:string elements.

FILTER_EQ

Defines the equal filter.

Note: Use this filter with xsd:long, dsm:DateTime, and enumerations.

FILTER_NE

Defines the not equal filter.

Note: Use this filter with xsd:long, dsm:DateTime, and enumerations.

FILTER_GT

Defines the greater than filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

FILTER_GE

Defines the greater than or equal filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

FILTER_LT

Defines lesser than filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

FILTER_LE

Defines the lesser than or equal filter.

Note: Use this filter with xsd:long and dsm:DateTime elements.

OSIMTEMPLATEPROPERTY

Specifies the properties of the OSIM templates that you can use to filter and sort the list of OSIM templates.

- OSIMTEMPLATENAME
- OSIMTEMPLATELASTMODIFIEDDATE
- OSIMTEMPLATELASTMODIFIEDUSER
- OSIMTEMPLATECREATIONDATE
- OSIMTEMPLATECREATIONUSER

SOFTWAREDEPLOYMENTTEMPLATEPROPERTY

Specifies the properties of the software deployment templates that you can use to filter and sort the list of software deployment templates.

- SOFTWAREDEPLOYMENTTEMPLATENAME
- SOFTWAREDEPLOYMENTTEMPLATELASTMODIFIEDDATE

AUTOMATIONPROCESSPROPERTY

Specifies the properties of the automation process:

- AUTOMATIONPROCESSNAME
- AUTOMATIONPROCESSSTATUS
- AUTOMATIONPROCESSMODE
- AUTOMATIONPROCESSCREATIONDATE
- AUTOMATIONPROCESSEXECUTIONFINISHDATE
- AUTOMATIONPROCESSEXECUTIONSTARTDATE
- AUTOMATIONPROCESSCREATIONUSER
- AUTOMATIONPROCESSEXECUTIONUSER

AutomationProcessMode

Specifies the automation process modes:

- WIPE_AND_RELOAD
- REPLACEMENT

AutomationProcessStatus

Specifies the status of the automation process:

- NOT_INITIALIZED
- INITIALIZATION_IN_PROGRESS
- INITIALIZATION_SUCCESS
- INITIALIZATION_FAILED
- RESUME_PROGRESS
- RESUME_SUCCESS
- RESUME_FAILED
- PREPARATION_SUCCESS
- PREPARATION_FAILED
- JOB_CREATION_IN_PROGRESS
- JOB_CREATION_FAILED
- VALIDATION_IN_PROGRESS
- VALIDATION_SUCCESS
- VALIDATION_FAILED
- DEPLOYMENT_IN_PROGRESS

- DEPLOYMENT_SUCCESS
- DEPLOYMENT_FAILED

AutomationTargetStatus

Specifies the status of the computers in the automation process:

- ACTIVITY_NOT_STARTED
- ACTIVITY_STARTED
- VALIDATION_SUCCESS
- VALIDATION_FAILED
- VALIDATION_WARNING
- PRE_OS_INSTALL_PHASE_STARTED
- PRE_OS_INSTALL_PHASE_FAILED
- PRE_OS_INSTALL_PHASE_SUCCESS
- OS_INSTALL_PHASE_STARTED
- OS_INSTALL_PHASE_FAILED
- OS_INSTALL_PHASE_SUCCESS
- POST_OS_INSTALL_PHASE_STARTED
- POST_OS_INSTALL_PHASE_FAILED
- POST_OS_INSTALL_PHASE_SUCCESS
- MIGRATION_SUCCESS
- MIGRATION_FAILED
- MIGRATION_DISABLED

AutomationResultProperty

Specifies the properties available in the automation result:

- AUTOMATIONTARGETNAME
- AUTOMATIONTARGETSTATUS
- AUTOMATIONSOURCENAME

LogLevel

Specifies the log level:

- INFORMATION
- WARNING
- ERROR

ActionType

Specifies the action to be taken on the automation process:

- MIGRATION
- VALIDATION

Methods

The following sections describe the methods available in CA ITCM Web Services.

Note: A method name that ends with an integer is a newer version of the method. If possible, use the latest version. For example, CopyElementsToSoftwarePackageVolume2 is a newer version of CopyElementsToSoftwarePackageVolume.

This section contains the following topics:

[Authentication and Web Services Management](#) (see page 135)

[Health Monitoring](#) (see page 156)

[Deployment](#) (see page 158)

[Queries](#) (see page 165)

[Security](#) (see page 178)

[Software](#) (see page 210)

[OSIM](#) (see page 448)

[Software Groups](#) (see page 491)

[Units and Groups](#) (see page 498)

[Enterprise Manager Distribution Functionalities](#) (see page 575)

[Enterprise Manager Domain and Domain Group Functionalities](#) (see page 597)

[Asset Script Job Methods](#) (see page 610)

Authentication and Web Services Management

The following methods authenticate or manage the timeout a session.

This section contains the following topics:

- [GetInstalledComponents](#) (see page 135)
- [GetInstalledComponents2](#) (see page 136)
- [GetInstalledComponents3](#) (see page 137)
- [WSChallenge](#) (see page 138)
- [WSChallenge2](#) (see page 140)
- [WSAuthenticate](#) (see page 142)
- [Login](#) (see page 143)
- [Login2](#) (see page 145)
- [Login3](#) (see page 150)
- [Login4](#) (see page 152)
- [Logout](#) (see page 153)
- [ResetTimeout](#) (see page 153)
- [SetTimeoutValue](#) (see page 154)
- [IsServiceDeskIntegrationEnabled](#) (see page 154)
- [GetServiceDeskTemplates](#) (see page 155)
- [GetManagerProperties](#) (see page 155)
- [GetManagerConfigurationParameterInt](#) (see page 156)

GetInstalledComponents

The GetInstalledComponents method retrieves the installed Components.

Parameters

xsd:string sessionId (M)

The session identifier.

Return Values

Components GetInstalledComponentsReturn.

The Component Structure contains the following elements.

- isAMInstalled;
- isSDInstalled;
- isRCInstalled;
- isBABLDInstalled;
- isSDESKInstalled;

All these variables are of type Boolean. If the variables have a true value, the component is installed.

Remarks

None

GetInstalledComponents2

The GetInstalledComponents2 method retrieves the installed Components including the Asset Collector.

Parameters

xsd:string sessionId (M)

The session identifier.

Return Values

Components2 GetInstalledComponentsReturn.

The Component Structure contains the following elements.

```
isAMInstalled;  
isSDInstalled;  
isRCInstalled;  
isBABLDInstalled;  
isSDESKInstalled;  
isACInstalled;
```

All these variables are of type Boolean. If the variables have a true value, the component is installed.

Remarks

None

GetInstalledComponents3

The GetInstalledComponents3 method returns the status of components installed.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfComponents components

Specifies the array of components for which the status must be retrieved.

Return Values

dsm:ArrayOfComponentStatus getStatus

Includes an array of components and their statuses.

WSChallenge

The WSChallenge method sends NTLMSSP_NEGOTIATE message to the manager and receives the NTLMSSP_CHALLENGE message in response. If the SessionID passed is NULL, this method creates a new SessionID when you invoke this method. The NTLMSSP_CHALLENGE message and the SessionID are sent to Web Service client as a response.

Note: Unified logon uses NTLM protocol for authentication.

The WSChallenge method uses the following messages:

NTLMSSP_NEGOTIATE

Includes an authentication request that is initiated by the browser on the request of the client. This message contains host name and windows domain name of the client.

NTLMSSP_CHALLENGE

Includes the response from the domain controller to the NTLMSSP_NEGOTIATE message. This message contains a nonce (random data), which is used for creating the response at browser.

Parameters

Char * sessionID

Specifies the session id obtained from the WSChallenge method. The Web Service client must authenticate individually to the CA ITCM components: common object (CO), asset management (AM) and software delivery (SD) to access their respective functionality. The client must authenticate to CO first and then to AM or SD. Therefore, this parameter can be NULL only when the parameter CompRequiresAuthentication is set to CO_REQUIRES_AUTHENTICATION.

Char * manager

Specifies the host name of the CA ITCM manager to which the method must connect to.

enum CompRequiresAuthentication compReqAuthen

Specifies the CA ITCM component for which you require authentication.

struct AuthenticationData * negotiateData,

Specifies the authentication data to be sent to domain controller.

Return Values

WSChallengeResponse

Refers to the structure that is sent back to the client as a response.

Remarks

None

More information:

[AuthenticationData](#) (see page 631)

[WSChallengeResponse](#) (see page 914)

[CompRequiresAuthentication](#) (see page 68)

[WSAuthenticateResponse](#) (see page 914)

More Information:

[AuthenticationData](#) (see page 631)

[CompRequiresAuthentication](#) (see page 68)

WSChallenge2

The WSChallenge2 method sends NTLMSSP_NEGOTIATE message to the manager and receives the NTLMSSP_CHALLENGE message in response. If the SessionId passed is NULL, this method creates a new SessionId when you invoke this method. The NTLMSSP_CHALLENGE message and the SessionId are sent to Web Service client as a response.

Note: Unified logon uses NTLM protocol for authentication.

The WSChallenge method uses the following messages:

NTLMSSP_NEGOTIATE

Includes an authentication request that is initiated by the browser on the request of the client. This message contains host name and windows domain name of the client.

NTLMSSP_CHALLENGE

Includes the response from the domain controller to the NTLMSSP_NEGOTIATE message. This message contains a nonce (random data), which is used for creating the response at browser.

Parameters

xsd:string sessionId

Specifies the session id obtained from the WSChallenge method. The Web Service client must authenticate individually to the CA ITCM components: common object (CO), asset management (AM) and software delivery (SD) to access their respective functionality. The client must authenticate to CO first and then to AM or SD. Therefore, this parameter can be NULL only when the parameter CompRequiresAuthentication is set to CO_REQUIRES_AUTHENTICATION.

xsd:string manager

Specifies the host name of the CA ITCM manager to which the method must connect to.

xsd:string locale

Currently the supported locale values are "enu", "fra", "deu" and "jpn". This value will be used for returning locale specific error strings from Webservices. If locale is not supplied, the installed machine locale will be used

dsm:CompRequiresAuthentication CompRequiresAuthentication

Specifies the CA ITCM component for which you require authentication.

dsm: AuthenticationData negotiateData,

Specifies the authentication data to be sent to domain controller.

Return Values

xsd:string authenticationData

Specifies the authentication data returned by Domain controller

xsd:long size

Specifies the size of the authentication data

xsd:string sessionId

Specifies a unique session ID to use as an input parameter for all further calls to the web service.

Remarks

None

WSAuthenticate

The WSAuthenticate method sends an NTLMSSP_AUTH message to the manager and receives the authentication status in response. This authentication status is then sent to the Web Service client as a response.

The WSAuthenticate method uses the following messages:

NTLMSSP_AUTH

Includes the response (password hashes) from the browser to the NTLMSSP_CHALLENGE message. This message contains host name, Windows domain name, and two responses—LanManager and NT Response. The nonce passed by the NTLMSSP_CHALLENGE message is used to create these responses (password hashes).

Parameters

Char * sessionID

Specifies the session id obtained from the WSCheckout method.

Char * manager

Specifies the host name of the CA ITCM manager to which the method must connect to.

enum CompRequiresAuthentication compReqAuthen

Specifies the CA ITCM component for which you require authentication.

struct AuthenticationData * authenticationData,

Specifies the authentication data to be sent to domain controller.

Return Values

WSAuthenticateResponse

Refers to the structure that contains authentication status. This structure is sent to the Web Service client as a response.

Remarks

None

Login

The Login method authenticates and obtains authorization from the DSM manager.

Parameters

xsd:string userName

The name of a valid user account that has permission to authenticate with the DSM manager. If unified login is disabled, this is a mandatory parameter.

xsd:string password

The password that corresponds to the given user name. If unified login is disabled, this is a mandatory parameter.

xsd:string manager (M)

The host name of the DSM manager to which the method should attempt to connect.

Return Values

xsd:string LoginReturn

A unique session ID to use as an input parameter for all further calls to the web service.

Remarks

This method obtains a connection to the manager and creates a web service session, returning the session ID for further web service calls.

The credentials on Windows and Linux are as follows:

Windows Domain

url: http://<hostname>/UDSM_R11_WebService/mod_gsoap.dll

user : winnt://<domain name>/<username>

password : <password>

manager :<dsmmanagername>

Linux

url : http://<hostname>/UDSM_R11_WebService

user : unixl://<complete domain name>/<username>

password : <password>

manager :<dsmmanagername>

hostname

Specifies the name of the the computer on which Web Services is installed.

domain name

Specifies the NT domain name (either local machine name or domain name)

Complete domain name

Specifies the fully qualified machine name of the Linux machine

It is recommended that you use SSL to connect to Web Services.

When connecting to Active Directory:

url: http:// <hostname>/UDSM_R11_WebService (on Linux)

url: http://<hostname>/UDSM_R11_WebService/mod_gsoap.dll (on Windows)

user : ldap://<directory name>/<username>

<username> can be of UPN or DN format

In UPN format : <username>: user@directory name

IN DN Format : <username>: cn= user,ou=users,ou=XXXX,dc=YYY dc=com

When connecting to an LDAP enabled directory (not Active Directory):

url: http:// <hostname>/UDSM_R11_WebService (on Linux)

url: http://<hostname>/UDSM_R11_WebService/mod_gsoap.dll (on Windows)

user : ldap://<directory name>/<user dn>

<user dn> is the distinguished name format, for example

cn= user,ou=users,ou= XXXX,dc=YYY dc=com

When connecting to NDS (Windows only):

url: http://<hostname>/UDSM_R11_WebService/mod_gsoap.dll

user: nds://<tree name>/<user dn>

<user dn> is the distinguished name format; for example,

cn=user.ou=XXXX.o=aar

Login2

The Login2 method authenticates and obtains authorization from the CA ITCM manager. This method expects the UserName and Password to be a base 64 encoded UTF-8 string.

Parameters

xsd:string userName (")

The name of a valid user account that has permission to authenticate with the CA ITCM manager. The UserName needs to be a base64 encoded UTF8 string.

xsd:string password (")

The password that corresponds to the given user name.

The Password needs to be a base64 encoded UTF8 string.

xsd:string manager (")

The host name of the CA ITCM manager to which the method should attempt to connect.

Return Values

xsd:string LoginReturn

A unique session ID to use as an input parameter for all further calls to the web service.

Remarks

This method obtains a connection to the manager and creates a web service session, returning the session ID for further web service calls. The credentials on Windows and Linux is as follows:

Windows

url: http://<machine address>/UDSM_R11_WebService/mod_gsoap.dll

User : winnt://<machine address>/<userName>

Password : <password>

Manager :<dsmmanagename>

Linux

URL : http:// <Completemachine address >/UDSM_R11_WebService

User : unixl:// <Completemachine address >/<userName>

Password : <password>

Manager :<dsmmanagename>

For linux the complete machine address should be given including the domain name.

Example

On Windows

URL = http://acme.xyz.com/UDSM_R11_WebService/mod_gsoap.dll

userId = winnt://acme/testUser

password = "-----"

serverName =acme

On Linux

URL = http://acme.xyz.com/UDSM_R11_WebService

userId = unixl:///acme.xyz.com/root

password = "-----"

Base64 Encoding Algorithm

Base64 encoding uses only 6 bits (corresponding to $2^6 = 64$ characters), hence the name Base64. The 64 characters are 10 digits, 26 lowercase characters, 26 uppercase characters, the addition sign (+) and /

Base64 encoding is a two step process. A stream of bytes is divided into groups, each consisting of three consecutive bytes. Each group (three sequential bytes, each consisting of eight bits) is converted to four numbers of six bits. In the second step, numbers are converted to ASCII characters using the Base64 encoding table.

For example,

Step 1: The string "\$t@" (36, 116, 64), the corresponding bit stream is 001001000111010001000000, which in turn corresponds to the 6-bit values 9, 7, 17 and 0.

36 = 00100100

116 = 01110100

64 = 01000000

001001 = 9

000111 = 7

010001 = 17

000000 = 0

Step 2: These numbers are converted to ASCII characters using the Base64 encoding table. The 6-bit values of our example translate to the ASCII sequence "JHRA".

9 => J

7 => H

17 => R

0 => A

Note: If the size of the original data in bytes is not multiple of three, there may be one or two 8-bit bytes remaining. The solution is to append enough bytes with a value of '0' to create a 3-byte group.

These artificial trailing '0's are not encoded using the encoding table. They are represented by a 65th character. The Base64 padding character is '='.

The Base 64 Alphabet

0 A

1 B

2 C

3 D

4 E

5 F

6 G

7 H

8 I

9 J

10 K

11 L

12 M

13 N

14 O

15 P

16 Q

17 R

18 S

19 T

20 U

21 V

22 W

23 X

24 Y

25 Z

26 a

27 b

28 c

29 d

30 e

31 f

32 g

33 h

34 i

35 j

36 k

37 l

38 m

39 n

40 o

41 p

42 q

43 r

44 s

45 t

46 u

47 v

48 w

49 x

50 y

51 z

52 0

53 1

54 2

55 3

56 4

57 5

58 6

59 7

60 8

61 9

62 +

63 /

Login3

The Login3 method authenticates and obtains authorization from the DSM manager. A new parameter locale is introduced, which will accept the client locale.

Parameters

xsd:string userName

The name of a valid user account that has permission to authenticate with the DSM manager.

xsd:string passwordThe password that corresponds to the given user name. If unified login is disabled, this is a mandatory parameter.

xsd:string manager

The host name of the DSM manager to which the method should attempt to connect.

xsd:string locale

Currently the supported locale values are "enu", "fra", "deu" and "jpn". This value will be used for returning locale specific error strings from Webservices. If locale is not supplied, the installed machine locale will be used.

Return Values**xsd:string LoginReturn**

A unique session ID to use as an input parameter for all further calls to the web service.

Remarks

This method obtains a connection to the manager and creates a web service session, returning the session ID for further web service calls.

Example

On Windows

URL = http://acme.xyz.com/UDSM_R11_WebService/mod_gsoap.dll or

URL = http://<machinename>/DSM_WebService/mod_gsoap_utf8.dll

userId = winnt://acme/testUser

password = "-----"

managerName =acme

locale = ENU/FRA/DEU/JPN

On Linux

URL = http://acme.xyz.com/UDSM_R11_WebService or
http://<machine-name>/DSM_WebService

userId = winnt://acme/testUser

password = "-----"

managerName =acme

Login4

The Login4 method authenticates and obtains authorization from the CA ITCM manager. This method expects the UserName and Password to be a base 64 encoded UTF-8 string.

Parameters

xsd:string userName

The name of a valid user account that has permission to authenticate with the CA ITCM manager. The UserName needs to be a base64 encoded UTF8 string.

xsd:string password

The password that corresponds to the given user name.

The Password needs to be a base64 encoded UTF8 string.

xsd:string manager

The host name of the CA ITCM manager to which the method should attempt to connect.

xsd:string locale

Currently the supported locale values are "enu", "fra", "deu" and "jpn". This value will be used for returning locale specific error strings from Webservices. If locale is not supplied, the installed machine locale will be used

Return Values

xsd:string LoginReturn

A unique session ID to use as an input parameter for all further calls to the web service.

Remarks

This method obtains a connection to the manager and creates a web service session, returning the session ID for further web service calls.

On Windows

URL = http://acme.xyz.com/UDSM_R11_WebService/mod_gsoap.dll or

URL = http://<machinename>/DSM_WebService/mod_gsoap_utf8.dll

userId = winnt://acme/testUser

password = "-----"

managerName =acme

locale = ENU/FRA/DEU/JPN

On Linux

URL = `http://acme.xyz.com/UDSM_R11_WebService` or

URL = `http://<machine-name>/DSM_WebService`

userId = `winnt://acme/testUser`

password = `"-----"`

managerName = `acme`

Logout

The Logout method closes the web service connection to the CA ITCM manager, closes the web service session, and performs cleanup operations.

Parameters

xsd:string sessionId

The session ID identifying the web service session to close and clean up. Mandatory.

Return Values

This is a void method that does not expose a return value.

Remarks

None

ResetTimeout

The ResetTimeout method resets the web service timeout to the CA ITCM default value as defined in the CA ITCM configuration store.

Parameters

xsd:string sessionId

The session identifier. Mandatory.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetTimeoutValue

The SetTimeoutValue method sets the timeout for a web service session.

Parameters

xsd:string sessionId

The session identifier. Mandatory.

SetTimeoutProperties setTimeoutProperties

The properties that indicate the new timeout value to be set. The value is in seconds and the range is 60 to 7200; that is, 1 min. to 2 hrs. Mandatory.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetTimeoutProperties](#) (see page 766)

IsServiceDeskIntegrationEnabled

The IsServiceDeskIntegrationEnabled method verifies whether CA ITCM and CA Service Desk integration is enabled.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Return Values

xsd:boolean isIntegrationEnabled

Returns a boolean value depending on whether CA Service Desk integration is enabled.

GetServiceDeskTemplates

The GetServiceDeskTemplates method returns a list of CA Service Desk templates in CA Service Desk Manager. The list is returned based on the index and the number of templates you specify.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

xsd:long noOfTemplates

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

xsd:long totalNoOfTemplates

Returns the total number of templates in the list.

GetManagerProperties

The GetManagerProperties method returns the manager properties like version number, language, and so on.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Return Values

dsm: ManagerProperties props

Contains manager properties.

GetManagerConfigurationParameterInt

The GetManagerConfigurationParameterInt method returns the value of the configuration parameters.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfConfigurationParams params (m)

Specifies an array of configuration parameters for which the values must be retrieved.

Return Values

dsm:ArrayOfConfigurationParamValues configValues

Contains an array of configuration parameters along with their values.

Health Monitoring

This section contains the following topics:

[DeleteAlerts](#) (see page 156)

[UpdateAlerts](#) (see page 157)

[DeleteAllAlerts](#) (see page 158)

DeleteAlerts

This method deletes HM Alerts.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

dsm:ArrayOfstring deleteAlerts

Array of alert UUIDs that must be deleted.

Return Values

None.

UpdateAlerts

This method changes the state of the HM Alert (New/FollowUp/Cleared). It also allows the user to add a User Message (or Description).

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

dsm:ArrayOfstring alertIds

Array of alert UUIDs that must be updated

dsm:UpdateAlertsProperties alertProperties

The properties of the alert(s) that must be updated.

Return Values

None.

DeleteAllAlerts

This method deletes all the alerts present on the HM Alert page.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

dsm:ALERTSTATE alertState

The state (New/Follow-Up/Cleared) of the HM alerts that need to be deleted.

Return Values

None.

Deployment

This section contains the following topics:

[ScanPackageOnTargetComputers](#) (see page 158)

[AbortDeploymentJob](#) (see page 159)

[SuspendDeploymentJob](#) (see page 160)

[ResumeDeploymentJob](#) (see page 160)

[DiscardDeploymentJob](#) (see page 161)

[RetryDeploymentJob](#) (see page 161)

[GetDeploymentJobStatus](#) (see page 162)

[GetDeploymentPackagesOnScalabilityServer](#) (see page 163)

[DeployPackage](#) (see page 163)

[StagePackage](#) (see page 164)

ScanPackageOnTargetComputers

This method performs a scan of the target computers for a package to check whether deployment of it is possible on them.

Parameters

xsd:string sessionId (m)

The session identifier, obtained from the login to the web service.

dsm:ScanParameters scanParameters (m)

Contains information regarding the target computers on which the scan is required to take place. Also provides the options to specify whether the package to be deployed would come from a scalability server or manager, and whether the scan to be performed on staged packages or deployed ones on the target machines.

dsm:ScanPackageInfo scanPackageInfo (m)

Contains information regarding the package that needs to be scanned for.

xsd:string jobName (m)

The job name that's assigned to the Scan operation and the subsequent deploy.

Return Values**xsd:long**

Job Id assigned for the scan operation.

Remarks

This method mirrors the functionality of "scan" cmd line option available in DMSweep. One of the inputs specifying the target computers is mandatory in ScanParameters. GetDeploymentJobStatus() needs to be called after this method to get the status of the scan.

More information:

[ScanParameters](#) (see page 719)

[ScanPackageInfo](#) (see page 719)

AbortDeploymentJob

This method is used to halt a running deployment job as referenced by the deployment job identifier. The system will abort the installation at a safe point in the deployment for all specified targets. For targets that deployment has not yet start these will simply not occur, where package transfer has started this will be stopped and temporary files deleted, where the package installer has started the installation it will go to completion

Parameters**xsd:string sessionId (m)**

Session Id, obtained from the login to the web service.

xsd:long jobId (m)

Job identified for the deployment job.

Return Values

None

Remarks

DeploymentJobId will be coming from either scan or deploy.

SuspendDeploymentJob

This method is used to suspend a running deployment job as referenced by the deployment job identifier. The system will suspend the installation at a safe point in the deployment for all specified targets. For targets where package transfer has started this will be stopped at the current point, for targets where the package installer has started the installation it will go to completion.

Parameters

xsd:string sessionId (m)

Session Id, obtained from the login to the web service.

xsd:long jobId (m)

Job identified for the deployment job.

Return Values

None

Remarks

- Suspending a suspended job does not result in error.
- Suspend operation on an aborted or discarded job will return an error.

ResumeDeploymentJob

This method is used to resume a suspended deployment job as referenced by the job identifier.

Parameters

xsd:string sessionId (m)

Session Id, obtained from the login to the web service.

xsd:long jobId (m)

Job identified for the deployment job.

Return Values

None

Remarks

Resuming a running job does not result in error.

Targets where deployment has not yet started will be scheduled; targets where package transfer has been suspended will be resumed.

DiscardDeploymentJob

This method is used to purge a deployment job referenced by the job identifier from the DMDeploy manager.

Parameters

xsd:string sessionId (m)

Session Id, obtained from the login to the web service.

xsd:long jobId (m)

Job identified for the deployment job.

Return Values

None

RetryDeploymentJob

This method is used to redeploy to a range of machines from within a deployment job as referenced by the job identifier.

Parameters

xsd:string sessionId (M)

The session identifier, obtained from the login to the web service.

xsd:long jobId (M)

Job identifier for the deployment job.

dsm:ArrayOfMachineCredentials arrayOfTargetMachines(m)

Machine list on which the retry is attempted. Specify valid credentials if the deployment failed due to wrong credentials, else just pass in the machine name or IP with user name and password as null.

Return Values

None

More information:

[MachineCredentials](#) (see page 706)

GetDeploymentJobStatus

This method is used to retrieve the status of a current deployment from the DMDeploy manager referenced by the job identifier. This returns the status of the scan or deployment at that point of time.

Parameters

xsd:string sessionId

The session identifier, obtained from the login to the web service.

xsd:long jobId

Jobid of the deployment or scan operation.

dsm:ComputerStatusInfoRequired

Specifies the deployment status information that needs to be retrieved.

Return Values

xsd:boolean bNoMoreComputers

If the value is true, it indicates that there is no more status information to be retrieved or the required operation is done. If false, it specifies that the scan or deploy operation is still in progress and further status can be retrieved by calling the method again.

dsm:ArrayOfComputerStatusInfo arrayOfcomputerStatusInfo

Returns an array of ComputerStatusInfo structures that contain a snapshot of the status of the scan or deployment

Remarks

This method needs to be called repeatedly (generally with a time lapse) until bNoMoreComputers flag is true to track and retrieve the whole status of the deployment operation.

More information:

[ComputerStatusInfo](#) (see page 67)

[ComputerStatusInfoRequired](#) (see page 673)

[DeploymentStatusCode](#) (see page 68)

GetDeploymentPackagesOnScalabilityServer

This method lists the deployment packages found on a scalability server.

The output of this method is equivalent to the “sspack” cmd line option used with dmsweep.

Parameters

xsd:string sessionId

The session identifier, obtained from the login to the web service

xsd:string scalabilityServerName

The scalability server that needs to be queried for the list of packages.

dsm:SSPackageInfoRequired ssPackageInfoRequired

Specifies the package information required

Return Values

**dsm:GetDeploymentPackagesOnScalabilityServerResponse
getDeploymentPackagesOnSSResponse**

Returns an array of SSPackageInfo structures which contain the information regarding the various packages on a scalability server.

More information:

[SSPackageInfoRequired](#) (see page 867)

[SSPackageInfo](#) (see page 867)

DeployPackage

Used to deploy a package on the target computers. The jobId from an earlier Scan operation needs to be passed in as input to this method. This jobId basically identifies the list of computers from the scan on which the deployment needs to take place.

Parameters

xsd:string sessionId

The session identifier, obtained from the login to the web service.

xsd:long jobId

JobId returned from the scan method

dsm: DeploymentInfo deploymentInfo

Contains information regarding the target machines credentials and scalability server if required. If the user identified by `targetUserName` and `targetPassword` doesn't have access to all of the machines targeted for deployment, then the field `arrayOfTargetMachineCredentials` can be used to supply machine specific credentials. If deployment is required to take place from a scalability server, then scalability server details needs be appropriately supplied.

dsm: DeploymentPackageInfo deploymentPackageInfo

Contains information regarding the package that needs to be deployed. Appropriate parameter values for the package needs to be set before calling `Deploy`.

Return Values

None

Remarks

`GetDeploymentJobStatus()` needs to be called after this method to get the status of the deployment operation.

More information:

[DeploymentInfo](#) (see page 699)

[DeploymentPackageInfo](#) (see page 700)

[MachineCredentials](#) (see page 706)

[ParameterValue](#) (see page 708)

StagePackage

Used to stage a package on the target computers. The `jobId` from an earlier `Scan` operation needs to be passed in as input to this method. This `jobId` basically identifies the list of computers from the scan on which the deployment needs to take place.

Parameters

xsd:string sessionId

The session identifier, obtained from the login to the web service.

xsd:long jobId

JobId returned from the scan method

dsm: StagingCredentials stagingCredentials

Contains information regarding the target machine credentials. Here all the target machines are scalability servers. If the user identified by targetUserName and targetPassword doesn't have access to all of the machines targeted for staging, then the field arrayOfTargetMachineCredentials can be used to supply machine specific credentials.

dsm: DeploymentPackageInfo deploymentPackageInfo

Contains information regarding the package that needs to be deployed.

Return Values

None

Remarks

While passing in the DeploymentPackageInfo, parameter values can be omitted. Parameter values are not considered for package staging.

More information:

[DeploymentPackageInfo](#) (see page 700)

[MachineCredentials](#) (see page 706)

[StagingCredentials](#) (see page 868)

Queries

This section describes the methods that handle queries.

This section contains the following topics:

[CloseQueryList](#) (see page 166)

[CloseQueryResultList](#) (see page 166)

[CreateQuery](#) (see page 167)

[CreateQuery2](#) (see page 167)

[DeleteQueries](#) (see page 168)

[GetQueries](#) (see page 168)

[GetQuery](#) (see page 169)

[GetQuery2](#) (see page 170)

[GetQueryResults](#) (see page 170)

[GetOSTypesXML](#) (see page 172)

[OpenQueryList](#) (see page 173)

[OpenQueryResultList](#) (see page 173)

[SetQuery](#) (see page 174)

[SetQuery2](#) (see page 175)

[GetQueryFolderId](#) (see page 175)

[GetQueries2](#) (see page 176)

[GetQueryFolders](#) (see page 177)

CloseQueryList

The CloseQueryList method closes a query list that was opened by OpenQueryList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to the list to close. The handle was generated by a previous call to OpenQueryList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseQueryResultList

The CloseQueryResultList method closes a query's result list that was opened by OpenQueryResultList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to the list to close. The handle was generated by a previous call to OpenQueryResultList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateQuery

The CreateQuery method creates a new CA ITCM query.

Parameters

xsd:string sessionId (m)

The session identifier.

CreateQueryProperties createQueryProperties (m)

The properties required to create a query.

Return Values

xsd:string CreateQueryReturn

If a query was successfully created, the UUID of the new query is returned.

Remarks

None

More information:

[CreateQueryProperties](#) (see page 640)

CreateQuery2

The CreateQuery2 method creates a new DSM query.

Parameters

xsd:string sessionId (m)

The session identifier.

CreateQueryProperties2 createQueryProperties2 (m)

The properties required to create a query.

Return Values

xsd:string CreateQueryReturn

If a query was successfully created, the UUID of the new query is returned.

Remarks

None

DeleteQueries

The DeleteQueries method deletes one or more CA ITCM queries identified by their UUIDs.

Parameters

xsd:string sessionId (^m)

The session identifier.

ArrayOfstring queryIds (^m)

An array of UUIDs that identify the queries to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More Information:

[CreateQueryProperties](#) (see page 640)

GetQueries

The GetQueries method retrieves the queries from a query list.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list returned by a previous call to OpenQueryList.

xsd:long noOfQueries (^m)

The total number of queries to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenQueryList.

Return Values**ArrayOfQueryProperties GetQueriesReturn**

An array of QueryProperties - one for each query requested from the list.

Remarks

You can make multiple calls to GetQueries to iterate down the list.

GetQuery

The GetQuery method retrieves a CA ITCM query definition identified by its UUID.

Parameters**xsd:string sessionId (^m)**

The session identifier.

xsd:string queryId (^m)

The UUID of the query to retrieve.

QueryPropertiesRequired queryPropertiesRequired (^m)

This parameter specifies the query properties to retrieve.

Return Values**QueryProperties GetQueryReturn**

The query properties retrieved.

Remarks

None

More information:

[QueryProperties](#) (see page 716)

[QueryPropertiesRequired](#) (see page 718)

GetQuery2

The GetQuery2 method retrieves a CA ITCM query definition identified by its UUID. This method returns the sql statement of the query and link query name(if any).

Parameters

xsd:string sessionId

The session identifier.

xsd:string queryId

The UUID of the query to retrieve.

dsm:QueryPropertiesRequired3 queryPropertiesRequired

This parameter specifies the query properties to retrieve.

Return Values

dsm:QueryProperties3 GetQueryReturn

The query properties retrieved.

More Information:

[QueryResultProperties](#) (see page 708)

[QueryResultPropertiesRequired](#) (see page 709)

GetQueryResults

The GetQueryResults method retrieves the result objects from a query result list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list returned by a previous call to OpenQueryResultList.

xsd:long noOfResults (^m)

The total number of results to get from the list. This is less than or equal to the total number in the list returned by a previous call to `OpenQueryResultList`.

Return Values**ArrayOfQueryResultProperties GetQueryResultsReturn**

An array of `QueryResultProperties` - one for each result requested from the list.

Remarks

You can make multiple calls to `GetQueryResults` to iterate down the list.

GetOSTypesXML

This method returns the operating system class ID's and names in XML format.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

dsm:ArrayOfint classIds

If NULL, it will return the entire hierarchy of class ID's and names. If valid classIds are supplied, the tree hierarchy starting from the supplied ones will be returned.

For example, if you want all Windows 7 related class ID's, specify value 5156. The response would be in the following format:

```
<OSTypes>
<OS name="Windows 7" classID="5156">
  <OS name="Windows Embedded Standard 7 X64 Edition" classID="5255"/>
    <OS name="Windows Embedded Standard 7 Edition"
classID="5254"/>
  <OS name="Windows 7 Ultimate x64 Edition"
classID="5167"/>
    <OS name="Windows 7 Ultimate"
classID="5166"/>
      <OS name="Windows 7
Home Basic x64 Edition" classID="5165"/>
        <OS name="Windows 7 Home Basic" classID="5164"/>
          <OS name="Windows 7 Starter" classID="5163"/>
            <OS name="Windows 7 Professional x64
Edition" classID="5162"/>
              <OS name="Windows 7
Professional" classID="5161"/>
                <OS name="Windows 7 Home Premium x64 Edition" classID="5160"/>
                  <OS name="Windows 7 Home Premium" classID="5159"/>
                    <OS name="Windows 7 Enterprise x64 Edition"
classID="5158"/>
                      <OS name="Windows 7
Enterprise" classID="5157"/>
</OS>
</OSTypes>
```

Return Values

xsd:string osTypesXML

Operating system class ID's and names in XML format.

More Information:

[QueryPropertiesRequired](#) (see page 718)

OpenQueryList

The OpenQueryList opens a list of all CA ITCM queries.

Parameters

xsd:string sessionId (")

The session identifier.

QueryPropertiesRequired queryPropertiesRequired (")

The properties that will be required when getting a query from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of queries that can be passed to GetQueries

xsd:long totalNoOfQueries

The total number of queries in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetQueries method.

OpenQueryResultList

The OpenQueryResultList executes a CA ITCM query and opens a list of the results from the query.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string queryId (")

The UUID identifying the query whose results are to be listed.

QueryResultPropertiesRequired queryResultPropertiesRequired (")

The properties that will be required when getting a query result from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of queries that can be passed to GetQueryResults.

xsd:long totalNoOfQueryResults

The total number of query results in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetQueryResults method.

More information:

[QueryResultPropertiesRequired](#) (see page 709)

SetQuery

The SetQuery method updates the properties of a query definition.

Parameters

xsd:string sessionId (")

The session identifier.

SetQueryProperties setQueryProperties (")

The properties that are updated for the specified query.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetQueryProperties](#) (see page 734)

SetQuery2

The SetQuery method updates the properties of a query definition. We can update the link query using this method.

Parameters

xsd:string sessionId

The session identifier.

xsd:string queryId

UUID of the query to be updated

dsm:SetQueryProperties3 setQueryProperties

The properties that are updated for the specified query.

Return Values

None.

GetQueryFolderId

The GetQueryFolderId method returns the ID of the specified query folder.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string folderName

Specifies the name of the query folder, the ID of which you want to retrieve.

Xsd:string parentFolderName

Specifies the parent folder name.

Return Values

Xsd:long folderId

Contains the ID of the specified query folder.

GetQueries2

The GetQueries2 method retrieves the queries in a query folder. This method supports the sort and filter operations also.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string folderId

Specifies the ID of the folder from which you want to retrieve the queries. Specify 0 for the All Queries folder.

dsm:ArrayOfQueryFilter filterArray

Specifies an array of filters on query properties.

dsm:QueryProperty sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

xsd:long numRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

xsd:long totalNumQueries

Returns the total number of queries available for the specified filter condition.

dsm:ArrayOfQueries queryArray

Contains an array of queries in the specified folder, filtered and sorted as specified.

GetQueryFolders

The "GetQueryFolders" API returns the query folders.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string folderId

Specifies the ID of the parent folder from which you want to retrieve the query folders. Specify 0 to retrieve from the Queries root folder.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

xsd:long numRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

xsd:long totalNumFolders

Contains the total number of folders available.

dsm:ArrayOfQueryFolderProperties queryFolderArray

Contains an array of query folder properties inside the specified folder.

Security

This section contains the following topics:

- [CreateSecurityProfile](#) (see page 179)
- [DeleteSecurityProfiles](#) (see page 180)
- [OpenSecurityProfileList](#) (see page 180)
- [GetSecurityProfiles](#) (see page 181)
- [CloseSecurityProfileList](#) (see page 182)
- [GetSecurityProfile](#) (see page 182)
- [SetSecurityProfile](#) (see page 183)
- [OpenSecurityClassList](#) (see page 183)
- [GetSecurityClasses](#) (see page 184)
- [CloseSecurityClassList](#) (see page 185)
- [GetSecurityAreaConfiguration](#) (see page 185)
- [SetSecurityAreaConfiguration](#) (see page 186)
- [GetSecurityObjectOwnership](#) (see page 186)
- [SetSecurityObjectOwnership](#) (see page 187)
- [GetSecurityObjectPermissionsString](#) (see page 188)
- [GetSecurityObjectsSecurityAreaLinkages](#) (see page 189)
- [SetSecurityObjectsSecurityAreaLinkages](#) (see page 189)
- [SetSecurityClassPermissions](#) (see page 190)
- [GetSecurityClassPermissions](#) (see page 191)
- [GetSecurityClassPermissionsForAllSecurityProfiles](#) (see page 191)
- [GetAllSecurityClassPermissionsForSecurityProfile](#) (see page 192)
- [GetSecurityObjectPermissions](#) (see page 192)
- [GetSecurityObjectPermissionsForAllSecurityProfiles](#) (see page 193)
- [SetSecurityObjectsPermissions](#) (see page 194)
- [GetSecurityGroupsPermissions](#) (see page 195)
- [GetSecurityGroupPermissionsForAllSecurityProfiles](#) (see page 196)
- [SetSecurityGroupsPermissions](#) (see page 197)
- [LinkSecurityAreasToSecurityProfile](#) (see page 198)
- [UnLinkSecurityAreasFromSecurityProfile](#) (see page 198)
- [LinkSecurityAreasToSecurityProfileByName](#) (see page 199)
- [UnLinkSecurityAreasFromSecurityProfileByName](#) (see page 200)
- [LinkSecurityProfilesToSecurityAreaByName](#) (see page 200)
- [UnlinkSecurityProfilesFromSecurityAreaByName](#) (see page 201)
- [LinkSecurityProfilesToSecurityArea](#) (see page 201)
- [UnlinkSecurityProfilesFromSecurityArea](#) (see page 202)
- [CreateSecurityArea](#) (see page 202)
- [DeleteSecurityAreas](#) (see page 203)
- [OpenSecurityAreaList](#) (see page 203)
- [GetSecurityAreas](#) (see page 204)
- [CloseSecurityAreaList](#) (see page 204)
- [GetSecurityArea](#) (see page 205)
- [GetSecurityAreaByName](#) (see page 205)
- [SetSecurityArea](#) (see page 206)
- [DeleteSecurityAreasByName](#) (see page 206)
- [SetSecurityAreaByName](#) (see page 207)
- [GetSecurityAreaConfigurationByName](#) (see page 207)
- [SetSecurityAreaConfigurationByName](#) (see page 208)
- [DeleteSecurityProfilesByName](#) (see page 208)

[SetSecurityProfileByName](#) (see page 209)

[GetSecurityProfileByName](#) (see page 209)

CreateSecurityProfile

This method is used to create security profile.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

dsm: CreateSecurityProfileProperties createSecurityProfileProperties (m)

The properties required to create a security profile.

Return Values

xsd:string securityProfileUUID

If a security profile was successfully created, the UUID of the new security profile is returned.

Remarks

- If both SecurityProfilePermission and SecurityProfileSpecialPermissionFlags are not supplied then the default permission mask "No Access" will be applied.
- If users wants to create Special Access permissions they have to provide 'SPECIALACCESS' to 'securityProfilePermission'.
- securityProfilePermissionSupplied' should be set to 'true'.
- Users must set structure variables in structure 'SecurityProfileSpecialPermissionFlags' to either true or false. Also, 'SecurityProfileSpecialPermissionFlagsSupplied' should be set to true.

More information:

[CreateSecurityProfileProperties](#) (see page 642)

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityProfilePermission](#) (see page 90)

DeleteSecurityProfiles

This method deletes security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:ArrayOfString securityProfileUUIDs (m)

An array of security profile UUIDs identifying the security profiles to delete.

Return Values

None

OpenSecurityProfileList

This method opens a list of security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm: SecurityProfilePropertiesRequired securityProfilePropertiesRequired (m)

The properties that will be required while retrieving a security profile from the list.

Return Values

xsd:long openListHandle

A numeric handle to a list of security profiles that can be passed to GetSecurityProfiles.

xsd:long totalNoOfSecurityProfiles

Total number of security profiles in the security profile list.

More information:

[SecurityProfilePropertiesRequired](#) (see page 727)

GetSecurityProfiles

This method retrieves the security profiles from a security profile list.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:long listHandle (M)

A numeric handle to a security profile list as returned by a previous call to OpenSecurityProfileList.

xsd:long noOfSecurityProfiles (M)

The total number of security profiles to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSecurityProfileList.

Return Values

dsm:ArrayOfSecurityProfileProperties GetSecurityProfilesReturn

An array of SecurityProfileProperties – one for each security profile requested from the list.

Remarks

You can make multiple calls to GetSecurityProfiles to iterate down the security profile list.

More information:

[SecurityProfileProperties](#) (see page 726)

[SecurityProfileType](#) (see page 91)

CloseSecurityProfileList

This method closes a security profile list that was opened by `OpenSecurityProfileList` and releases the list handle.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:long listHandle (M)

A numeric handle to a security profile list as returned by a previous call to `OpenSecurityProfileList`.

Return Values

None

GetSecurityProfile

This method retrieves a security profile identified by its UUID.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileId (m)

The UUID of the security profile to retrieve.

**dsm: SecurityProfilePropertiesRequired
securityProfilePropertiesRequired (m)**

This parameter specifies which security profile properties to retrieve.

Return Values

dsm:SecurityProfileProperties securityProfileProperties

The security profile properties retrieved.

More information:

[SecurityProfileProperties](#) (see page 726)

[SecurityProfilePropertiesRequired](#) (see page 727)

[SecurityProfileType](#) (see page 91)

SetSecurityProfile

This method updates the properties of a security profile.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileUUID (m)

UUID of the security profile whose properties need to be modified.

dsm:SetSecurityProfileProperties securityProfile (m)

The properties that need to be modified for the specified security profile.

Return Values

None

More information:

[SetSecurityProfileProperties](#) (see page 727)

OpenSecurityClassList

This method opens a list of security object classes.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id, obtained from the login to the web service.

dsm: SecurityClassPropertiesRequired securityClassPropertiesRequired (m)

The properties that will be required while retrieving a security classes from the list.

Return Values

xsd:long openListHandle

A numeric handle to a list of security object class that can be passed to GetSecurityClasses.

xsd:long totalNoOfSecurityClasses

The total number of security classes in the security object class list.

More information:

[SecurityClassPropertiesRequired](#) (see page 723)

GetSecurityClasses

This method retrieves properties of the security classes from security class list.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id, obtained from the login to the webservice.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenSecurityClassList.

xsd:long noOfSecurityclasses (m)

The total number of security classes to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSecurityClassList.

dsm:ArrayOfstring securityProfileUUIDs (m)

An array of security profile UUIDs. Security classes properties will be retrieved from the list with respect to this array of security profile UUIDs.

Return Values

dsm:ArrayOfSecurityClassProperties arrOfSecurityClassesProperties

An array of SecurityClassProperties – one for each security class.

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassPermissions](#) (see page 722)

[SecurityClassProperties](#) (see page 723)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

More Information:

[SecurityClassType](#) (see page 86)

[ArrayOfSecurityClassPermissions](#) (see page 989)

CloseSecurityClassList

This method closes a security profile list that was opened by OpenSecurityClassList and releases the list handle.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id, obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a security profile list as returned by a previous call to OpenSecurityObjectClassList.

Return Values

None

GetSecurityAreaConfiguration

This method retrieves configurations related to security area. It returns whether global security area support is enabled or not. It also returns the list of security areas and there respective flags indicating whether these security areas are enabled or not.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id, obtained from the login to the web service.

Return Values

xsd:bool securityAreaSupport

It returns whether global security area support is enabled or not.

dsm:ArrayOfSecurityAreas GetsecurityAreasReturn

An array of SecurityArea – one for each security area. It returns the list of security areas and there respective flags indicating whether these security areas are enabled or not.

More information:

[SecurityArea](#) (see page 750)

SetSecurityAreaConfiguration

This method modifies configurations related to security area. It modifies global security area support. It also modifies respective flags of the list of security areas indicating whether these security areas are enabled or not.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id, obtained from the login to the web service.

dsm:SecurityAreaConfiguration securityAreaConfiguration (m)

The properties that need to be modified for security area configuration.

Return Values

None

More information:

[SecurityAreaConfiguration](#) (see page 726)

[SecurityArea](#) (see page 750)

GetSecurityObjectOwnership

This method return owners for a list of security objects of a security object class.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id, obtained from the login to the web service.

dsm:SecurityClassType securityClassType (m)

It contains security class type of the given security object. For example, 'Computer' object will be represented by 'SEC_CLASS_COMPUTER'.

dsm:ArrayOfString securityObjectUUIDs

An array of security objects UUIDs.

Return Values

dsm:ArrayOfSecurityObjectOwner securityObjectOwner

An array of SecurityObjectOwner – one for each security object.

More information:

[SecurityClassType](#) (see page 86)

[SecurityObjectOwner](#) (see page 725)

SetSecurityObjectOwnership

Set the (unique) owner for a list of secured objects of one security object class to the current user.

Parameters**xsd:string sessionId (M)**

The session identifier. Session Id, obtained from the login to the web service.

dsm:SecurityClassType securityClassType (m)

It contains security class type of the given security object. For example, 'Computer' object will be represented by 'SEC_CLASS_COMPUTER'.

dsm:ArrayOfString securityObjectUUIDs (m)

An array of security objects UUIDs. Ownership of these security objects is to be modified.

Return Values

None

More information:

[SecurityClassType](#) (see page 86)

GetSecurityObjectPermissionsString

This method returns a string that reflects the permissions for the current user on the specified security object.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

dsm:SecurityClassType securityClassType (m)

It contains security class type of the given security object. For example, 'Computer' object will be represented by 'SEC_CLASS_COMPUTER'.

xsd:string securityObjectUUID (m)

UUID of the security object for which security permissions is to be retrieved.

Return Values

xsd:string securityObjectPermissionsString

The current permission assigned to the given object for the current user.

More information:

[SecurityClassType](#) (see page 86)

GetSecurityObjectsSecurityAreaLinkages

This method retrieves Security Areas which are linked to given multiple objects of an object class.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id, obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

It contains security class type of the given security objects. For example, 'Computer' object will be represented by 'SEC_CLASS_COMPUTER'.

dsm:ArrayOfstring securityObjectsUUIDs

An array of security objects UUIDs. Security areas linked to these objects will be returned.

Return Values

dsm:ArrayOfSecurityObjectsSecurityAreaLinkages

GetSecurityObjectsSecurityAreaLinkagesReturn

An array of SecurityObjectsSecurityAreaProperties – one for each security object.

SetSecurityObjectsSecurityAreaLinkages

This method updates area linkages for all Security Profiles for multiple objects of a given security class type. This method either links or unlinks given security objects from given security areas.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id, obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

It contains security class type of the given security objects. For example, 'Computer' object will be represented by 'SEC_CLASS_COMPUTER'.

dsm:ArrayOfSetSecurityObjectSecurityAreaLinkages

securityObjectsSecurityAreaLinkages (M)

An array of SecurityObjectSecurityAreaLinkages – one for each security object.

Return Values

None

More information:

[SecurityClassType](#) (see page 86)

[SecurityObjectSecurityAreaLinkages](#) (see page 725)

[SecurityObjectLinkageProperties](#) (see page 724)

[SecurityAreaLinkage](#) (see page 86)

SetSecurityClassPermissions

This method modifies security permissions of the given set of security classes with respect to the supplied security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id, obtained from the login to the web service.

dsm:ArrayOfSecurityClassType securityClassTypeArray

An array of SecurityClassType.

dsm:ArrayOfSetSecurityClassPermissions

setSecurityClassPermissionsArray

An array of SecurityClassPermissions – one for each security profile.

Return Values

None

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassPermissions](#) (see page 722)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

GetSecurityClassPermissions

This method retrieves security permissions of the given number of security classes with respect to the array of security profiles.

Parameters

xsd:string sessionId (M)

The session identifier.Session Id, obtained from the login to the web service.

dsm:ArrayOfSecurityClassType securityClassTypeArray

An array of 'SecurityClassType'.

dsm:ArrayOfstring securityProfileUUIDs

An array of security profiles UUIDs.

Return Values

dsm:ArrayOfGetSecurityClassPermissions

arrayOfGetSecurityClassPermissions

An array of SecurityClassPermissionProperties – one for each security class.

GetSecurityClassPermissionsForAllSecurityProfiles

This method retrieves security class permissions for all security profiles.

Parameters

xsd:string sessionId (M)

The session identifier.Session Id, obtained from the login to the web service.

dsm:SecurityClassType securityClassType

The security class type. Its security permissions will be retrieved with respect to all security profiles.

Return Values

dsm: ArrayOfSecurityClassPermissions arrayOfSecurityClassPermissions

An array of [SecurityClassPermissions](#) (see page 722) – one for each security profile.

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassPermissions](#) (see page 722)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

GetAllSecurityClassPermissionsForSecurityProfile

This method retrieves all security classes permissions for the supplied security profile.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id, obtained from the login to the web service.

xsd:string securityProfileUUID

The security profile UUID.

Return Values

**dsm:ArrayOfAllSecurityClassPermissionsForSecurityProfile
arrayOfGetSecurityClassPermissions**

An array of [SecurityClassPermissionsForSecurityProfile](#) (see page 723) – one for each security class.

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassType](#) (see page 86)

[SecurityClassPermissionsForSecurityProfile](#) (see page 723)

[SecurityProfilePermission](#) (see page 90)

GetSecurityObjectPermissions

This method retrieves the security permission of security objects with respect to supplied security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

Security objects class type.

dsm:ArrayOfstring securityObjectUUIDs (M)

An array security objects UUIDs.

dsm:ArrayOfstring securityProfileUUIDs (M)

An array of security profiles UUIDs.

Return Values

dsm:ArrayOfGetSecurityObjectsPermissions

ArrayOfGetSecurityObjectsPermissions

An array of GetSecurityObjectsPermissions – one for each security object.

GetSecurityObjectPermissionsForAllSecurityProfiles

This method retrieves security permissions of the security object with respect to all security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

Security objects class type.

xsd:string securityObjectUUID

UUID of the security object.

Return Values

dsm:ArrayOfSecurityObjectPermissionsForAllSecurityProfiles

ArrayOfSecurityObjectPermissions

An array of [SecurityObjectPermissionsForAllSecurityProfiles](#) (see page 726) – one for each security profile.

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

[SecurityObjectPermissionsForAllSecurityProfiles](#) (see page 726)

[SecurityObjectProfilePermission](#) (see page 724)

SetSecurityObjectsPermissions

This method modifies security permission of the security objects with respect to the number of security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

Security objects class type.

dsm:ArrayOfstring securityObjectsUUIDs

An array of security objects UUIDs.

**dsm:ArrayOfSetSecurityObjectsPermissions
securityObjectPermissionsArray (m)**

An array of SetSecurityObjectPermissions – one for each security profiles and their associated security permissions.

Return Values

None

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

[SetSecurityObjectPermissions](#) (see page 728)

[SecurityObjectPermissions](#) (see page 725)

GetSecurityGroupsPermissions

This method retrieves security permissions of the security group with respect to security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

Security objects class type.

xsd:ArrayOfstring securityGroupsUUIDs (M)

An array of security group UUIDs.

xsd:ArrayOfstring securityProfilesUUIDs (M)

An array of security profile UUIDs.

Return Values

**dsm:ArrayOfSecurityGroupsPermissions
arrayOfGetSecurityGroupPermissions**

An array of [GetSecurityGroupPermissions](#) (see page 711) – one for each security group.

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

[SecurityObjectPermissions](#) (see page 725)

[GetSecurityGroupPermissions](#) (see page 711)

[SecurityGroupProfilePermission](#) (see page 724)

[Inheritance Level](#) (see page 76)

GetSecurityGroupPermissionsForAllSecurityProfiles

This method retrieves security permissions of the security group with respect to allSecurity Profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

Security objects class type.

xsd:string securityGroupUUID (M)

UUID of the security group.

Return Values

**dsm:ArrayOfSecurityGroupPermissionsForAllSecurityProfiles
arrayOfSecurityGroupPermissions**

An array of SecurityGroupPermissionsForAllSecurityProfiles –one for each security profile.

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

[SecurityObjectPermissions](#) (see page 725)

[SecurityGroupPermissionsForAllSecurityProfiles](#) (see page 724)

SetSecurityGroupsPermissions

This method modifies security permissions of the security group's with respect to number of security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:SecurityClassType securityClassType (M)

Security objects class type.

dsm:ArrayOfstring securityGroupsUUIDs (M)

An array of security group UUIDs.

dsm:ArrayOfSetSecurityGroupPermissions securityGroupPermissionsArray

An array of SetSecurityGroupPermissions – one for each security profile.

Return Values

None

More information:

[SecurityProfileSpecialPermissionFlags](#) (see page 728)

[SecurityClassType](#) (see page 86)

[SecurityProfilePermission](#) (see page 90)

[SecurityObjectPermissions](#) (see page 725)

[SetSecurityGroupPermissions](#) (see page 728)

LinkSecurityAreasToSecurityProfile

This method links security areas to security profile.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileUUID

UUID of the security profile.

dsm:ArrayOfstring securityAreasUUIDs

An array of security areas to be linked to security profile.

Return Values

None

More information:

[SecurityAreaConfiguration](#) (see page 726)

[SecurityArea](#) (see page 750)

[SecurityAreaLinkage](#) (see page 86)

UnLinkSecurityAreasFromSecurityProfile

This method unlinks security areas from a security profile.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileUUID

UUID of the security profile.

dsm:ArrayOfstring securityAreasUUIDs

An array of security areas to be linked to security profile.

Return Values

None

More information:

[CreateSecurityProfileProperties](#) (see page 642)

[CreateSecurityAreaProperties](#) (see page 641)

[ArrayOfSecurityAreaByName](#) (see page 987)

LinkSecurityAreasToSecurityProfileByName

This method links security areas to a security profile identified by its name.

Parameters**xsd:string sessionId (M)**

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileName

Name of the security profile.

dsm:ArrayOfstring securityAreaNames

An array of security areas to be linked to security profile.

Return Values

None

More information:

[ArrayOfSecurityAreaByName](#) (see page 987)

More Information:

[ArrayOfSecurityAreaByName](#) (see page 987)

UnLinkSecurityAreasFromSecurityProfileByName

This method unlinks security areas from security profile identified by its name.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileName

Name of the security profile.

dsm:ArrayOfstring securityAreaNames

An array of security areas to be unlinked from security profile.

Return Values

None

LinkSecurityProfilesToSecurityAreaByName

This method links security profiles to a security area identified by its name.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaName (M)

The Name of the security area.

xsd:ArrayOfstring securityProfilesNames (M)

An array of security profiles to be linked to Security Area.

Return Values

None

UnlinkSecurityProfilesFromSecurityAreaByName

This method unlinks the security area from security profiles identified by its name.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaName (M)

The name of the security area.

dsm:ArrayOfstring securityProfilesNames (M)

An array of security profiles to be unlinked from the security area.

Return Values

None

More information:

[SecurityArea](#) (see page 750)

[SecurityAreaByName](#) (see page 725)

More Information:

[SecurityArea](#) (see page 750)

LinkSecurityProfilesToSecurityArea

This method links security area to an array of security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaUUID (M)

The UUID of the security Area.

xsd:ArrayOfstring securityProfilesUUIDs (M)

An array of security profiles to be linked to a security area.

Return Values

None

More Information:

[SecurityArea](#) (see page 750)

UnlinkSecurityProfilesFromSecurityArea

This method unlinks the security Area from an array of security profiles.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaUUID (M)

The UUID of the security Area.

dsm:ArrayOfstring securityProfilesUUIDs (M)

An array of security profiles to be unlinked from a security area.

Return Values

None

CreateSecurityArea

This method creates a security area.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm: CreateSecurityAreaProperties createSecurityAreaProperties (M)

The properties required to create a security area.

Return Values

xsd:string securityAreaUUID

The uuid of the created area.

More information:

[SecurityArea](#) (see page 750)

[CreateSecurityAreaProperties](#) (see page 641)

DeleteSecurityAreas

This method deletes security areas.

Parameters

xsd:string sessionId (M)

The session identifier. Session Id obtained from the login to the web service.

dsm:ArrayOfstring arealds (M)

UUIDs of the security areas which are to be deleted.

Return Values

None

More information:

[SecurityArea](#) (see page 750)

OpenSecurityAreaList

This method opens a list of security areas.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

dsm: SecurityAreaPropertiesRequired securityAreaPropertiesRequired (m)

The properties required while retrieving a security area from the list.

Return Values

xsd:long openListHandle

A numeric handle to a list of security Areas that can be passed to GetSecurityAreas.

xsd:long totalNoOfSecurityAreas

Total number of security areas in the security area list.

More information:

[SecurityAreaPropertiesRequired](#) (see page 722)

GetSecurityAreas

This method retrieves the list of security areas from the security area list.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a security area list as returned by a previous call to [OpenSecurityAreaList](#) (see page 203).

xsd:long noOfSecurityAreas (m)

The total number of security areas to be obtained from the list. This should be less than or equal to the total number in the list as returned by a previous call to [OpenSecurityAreaList](#) (see page 203).

Return Values

dsm:ArrayOfSecurityAreaProperties GetSecurityAreasReturn

An array of SecurityAreaProperties – one for each security area requested from the list.

More information:

[SecurityAreaProperties](#) (see page 722)

CloseSecurityAreaList

This CloseSecurityAreaList method closes a security area list that was opened by [OpenSecurityAreaList](#) (see page 203) and releases the list handle.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a security area list as returned by a previous call to [OpenSecurityAreaList](#) (see page 203).

Return Values

None

GetSecurityArea

This method retrieves a security area identified by its UUID.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaUUID (m)

UUID of the security area to retrieve.

dsm:SecurityAreaPropertiesRequired securityPropertiesRequired

Specifies the security area properties to retrieve.

Return Values

dsm:SecurityAreaProperties GetSecurityAreaReturn

Security area properties retrieved.

More information:

[SecurityAreaPropertiesRequired](#) (see page 722)

[SecurityAreaProperties](#) (see page 722)

GetSecurityAreaByName

This method retrieves a security area identified by its name.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaName (m)

Name of the security area to retrieve.

dsm:SecurityAreaPropertiesRequired securityAreaPropertiesRequired

Specifies the security area properties to retrieve.

Return Values

dsm:SecurityAreaProperties GetSecurityAreaReturn

Security area properties retrieved.

More information:

[SecurityAreaPropertiesRequired](#) (see page 722)

[SecurityAreaProperties](#) (see page 722)

SetSecurityArea

This method updates the properties of a security area identified by its UUID.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaUUID (m)

UUID of the security area to be updated

dsm:SetSecurityAreaProperties setSecurityAreaProperties

Specifies the security area properties that are to be updated.

Return Values

None

More information:

[SetSecurityAreaProperties](#) (see page 751)

DeleteSecurityAreasByName

This method deletes security areas identified by their names.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

dsm:ArrayOfstring areaNames (m)

Array of security areas to be deleted.

Return Values

None

More information:

[ArrayOfSecurityAreaByName](#) (see page 987)

SetSecurityAreaByName

This method updates the properties of a security area identified by its name.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityAreaName (m)

Name of the security area to be updated

dsm:SetSecurityAreaProperties setSecurityAreaProperties

Specifies the security area properties that are to be updated.

Return Values

None

More information:

[SetSecurityAreaProperties](#) (see page 751)

GetSecurityAreaConfigurationByName

This method retrieves whether security area support is enabled or not and the list of SecurityAreaNames with their corresponding SecurityAreaEnabled flag set or not.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

Return Values

bool securityAreaSupport

Specifies whether security area global support is enabled or disabled.

dsm:ArrayOfSecurityAreasByName *GetsecurityAreasReturn

An array of SecurityAreaByName— one for each security area.

More information:

[SecurityAreaByName](#) (see page 725)

SetSecurityAreaConfigurationByName

This method sets the SecurityAreaSupport flag to the user supplied value and changes the state of SecurityAreas to either enable or disabled.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

dsm: SecurityAreaConfigurationByName securityAreaConfiguration (m)

SecurityAreaSupport Value and list of SecurityAreaNames, SecurityAreaEnabled values to be set.

Return Values

None

More information:

[SecurityAreaByName](#) (see page 725)

[SecurityAreaConfigurationByName](#) (see page 726)

DeleteSecurityProfilesByName

This method is used for deleting security profiles.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

dsm:ArrayOfString securityprofileNames (m)

Specifies the names of the security profiles which are to be deleted.

Return Values

None

More information:

[SecurityProfileProperties](#) (see page 726)

SetSecurityProfileByName

This method updates the properties of a security profile identified by its name.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileName (m)

The name of the security profile.

dsm:SetSecurityProfileProperties setSecurityProfileProperties (m)

The properties that should be updated for the specified security profile.

Return Values

None

More information:

[SecurityProfileType](#) (see page 91)

[SetSecurityProfileProperties](#) (see page 727)

GetSecurityProfileByName

This method retrieves a security profile identified by its Name.

Parameters

xsd:string sessionId (m)

The session identifier. Session Id obtained from the login to the web service.

xsd:string securityProfileName (m)

The name of the security profile to retrieve.

**dsm: SecurityProfilePropertiesRequired
securityProfilePropertiesRequired (m)**

The parameters specify which security profile properties to retrieve.

Return Values

dsm:SecurityProfileProperties securityProfileProperties

The security profile properties retrieved.

More information:

[SecurityProfileProperties](#) (see page 726)

[SecurityProfilePropertiesRequired](#) (see page 727)

[SecurityProfileType](#) (see page 91)

Software

The following are Software methods.

This section contains the following topics:

[Software Job Containers](#) (see page 211)

[Software Jobs](#) (see page 241)

[Software Package Groups](#) (see page 288)

[Software Package Procedure Dependencies](#) (see page 310)

[Software Package Procedures](#) (see page 315)

[Software Package Procedure Groups](#) (see page 351)

[Software Package Volumes](#) (see page 368)

[Software Packages](#) (see page 378)

[Software Policies](#) (see page 398)

[Priority, Reinstall, and Policy Enhancement](#) (see page 428)

Software Job Containers

This section describes methods that handle software job containers.

Note: CA ITCM Web Services does not support distribution job containers related functionalities at Enterprise Manager Level.

This section contains the following topics:

[ActivateSoftwareJobContainer](#) (see page 212)
[CloseSoftwareJobContainerList](#) (see page 212)
[CloseSoftwareJobContainerList2](#) (see page 213)
[CloseParentSoftwarePackageJobContainerList](#) (see page 213)
[CloseSoftwareJobContainerListRecursive](#) (see page 214)
[CloseSoftwareJobContainerSoftwareJobList](#) (see page 214)
[CreateSoftwareJobContainer](#) (see page 215)
[CreateSoftwareJobContainer3](#) (see page 216)
[CreateSoftwareJobContainerProperties2](#) (see page 216)
[DeleteSoftwareJobContainers](#) (see page 217)
[FindSoftwareJobContainer](#) (see page 218)
[GetParentSoftwareJobContainers](#) (see page 219)
[GetParentSoftwareJobContainers3](#) (see page 220)
[GetSoftwareJobContainer](#) (see page 221)
[GetSoftwareJobContainer3](#) (see page 222)
[GetSoftwareJobContainers](#) (see page 222)
[GetSoftwareJobContainers3](#) (see page 223)
[GetSoftwareJobContainerSoftwareJobs](#) (see page 225)
[GetSoftwareJobContainerSoftwareJobs3](#) (see page 226)
[GetSoftwareJobContainersRecursive](#) (see page 227)
[GetSoftwareJobContainersRecursive3](#) (see page 228)
[HaltDeliverySoftwareJobContainer](#) (see page 229)
[OpenParentSoftwareJobContainerList](#) (see page 229)
[OpenParentSoftwareJobContainerList3](#) (see page 230)
[OpenSoftwareJobContainerList](#) (see page 231)
[OpenSoftwareJobContainerList3](#) (see page 232)
[OpenSoftwareJobContainerListRecursive](#) (see page 233)
[OpenSoftwareJobContainerListRecursive3](#) (see page 234)
[OpenSoftwareJobContainerSoftwareJobList](#) (see page 235)
[ResumeDeliverySoftwareJobContainer](#) (see page 236)
[SealAndBuildSoftwareJobContainer](#) (see page 236)
[SealSoftwareJobContainers](#) (see page 237)
[SetSoftwareJobContainer](#) (see page 237)
[SetSoftwareJobContainer3](#) (see page 238)
[UnsealSoftwareJobContainers](#) (see page 239)
[CreateSoftwareJobContainer4](#) (see page 239)
[GetServiceDeskRequestURLForContainer](#) (see page 240)

More Information:

[SoftwareJobContainerScope](#) (see page 92)

[SoftwareJobContainerProperties](#) (see page 786)

[SoftwareJobContainerState](#) (see page 92)

ActivateSoftwareJobContainer

The ActivateSoftwareJobContainer method activates a software job container identified by its UUID.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwareJobContainerId (^m)

The UUID identifying the software job container to activate.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwareJobContainerList

The CloseSoftwareJobContainerList method closes a software job container list that was opened by OpenSoftwareJobContainerList and releases the list handle.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwareJobContainerList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwareJobContainerList2

The CloseSoftwareJobContainerList2 method closes a software job container list that was opened by OpenSoftwareJobContainerList2 and releases the list handle

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwareJobContainerList2.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseParentSoftwarePackageJobContainerList

The CloseParentSoftwarePackageJobContainerList method closes a parent software job container list that was opened by OpenParentSoftwareJobContainerList and releases the list handle.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:long listHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenParentSoftwareJobContainerList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwareJobContainerListRecursive

The CloseSoftwareJobContainerListRecursive method closes a software job container list that was opened by OpenSoftwareJobContainerListRecursive and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwareJobContainerListRecursive.

Return Values

xsd:int CloseSoftwareJobContainerListRecursiveReturn

Remarks

None

CloseSoftwareJobContainerSoftwareJobList

The CloseSoftwareJobContainerSoftwareJobList method closes a software job list that was opened by OpenSoftwareJobContainerSoftwareJobList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwareJobContainerSoftwareJobList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateSoftwareJobContainer

The CreateSoftwareJobContainer method creates a new software job container.

Parameters

xsd:string sessionId (")

The session identifier.

softwarejobContainer (")

The properties required to create a software job container.

Return Values

xsd:string CreateSoftwareJobContainerReturn

If a software job container was successfully created, the UUID of the new container is returned.

Remarks

None

More information:

[CreateSoftwareJobContainerProperties](#) (see page 643)

CreateSoftwareJobContainer3

The CreateSoftwareJobContainer3 method creates a new software job container.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence CreateSoftwareJobContainerProperties3. This sequence includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (")

Specifies the session identifier obtained from the login method of the web service.

softwarejobContainer3 (")

Specifies the properties required to create a software job container.

Return Values

xsd:string softwareJobContainerId

Returns the UUID of the new container if a software job container is successfully created.

Remarks

None

More Information:

[SoftwareJobContainerPropertyMask](#) (see page 820)More Information:
[SoftwareJobContainerPropertyMask](#) (see page 820)

CreateSoftwareJobContainerProperties2

- xsd:string name (m)
- xsd:string comment
- xsd:boolean commentSupplied
- xsd:long priority

Default: 5

Limits: 1 to 10

- xsd:long prioritySupplied
- xsd: string parentSoftwareJobContainerId
Default: SD_Jobs
- xsd:boolean parentSoftwareJobContainerIdSupplied
- SoftwareJobContainerPropertyMask
softwareJobContainerPropertyMask
Default: linkageBatch and Cascade.
- xsd:boolean softwareJobContainerPropertyMaskSupplied

DeleteSoftwareJobContainers

The DeleteSoftwareJobContainers method deletes one or more software job containers.

Parameters

xsd:string sessionId (")

The session identifier.

ArrayOfstring softwareJobContainerIds (")

An array of software job container UUIDs identifying the software job containers to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

If a user attempts to delete a container that is sealed and activated, the SD API server initiates the deletion process and SD Task Manager performs the deletion. Use the Error Code 'SDA000816' and Error description 'Job cancellation initiated' as success. If the container is not sealed, the SD API server deletes it immediately.

More Information:

[SoftwareJobContainerScope](#) (see page 92)

FindSoftwareJobContainer

The FindSoftwareJobContainer method finds a job container given its name, type, and the UUID of its parent container.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string parentSoftwareJobContainerId

The UUID identifying the parent container of the job container to search for.

SoftwareJobContainerScope parentSoftwareJobContainerType (^m)

The type of the job container to search for.

xsd:string softwareJobContainerName (^m)

The name of the software job container to search for.

Return Values

xsd:string FindSoftwareJobContainerReturn

Remarks

The name must be specified exactly. This method does not allow wildcards. The software job container name is case insensitive.

If the parent UUID is not supplied, this method finds the SoftwareJobContainer under the superfolder, for example, All Software Jobs.

GetParentSoftwareJobContainers

The GetParentSoftwareJobContainers method retrieves the parent software job containers from a software job containers's parent group list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenParentSoftwareJobContainerList.

xsd:long noOfParentSoftwareJobContainers (")

The total number of software job containers to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenParentSoftwareJobContainerList.

Return Values

ArrayOfSoftwareJobContainerProperties

GetParentSoftwareJobContainersReturn

An array of SoftwareJobContainerProperties - one for each parent container requested from the list.

Remarks

You can make multiple calls to GetParentSoftwareJobContainers to iterate down the list.

More information:

[SoftwareJobContainerProperties](#) (see page 786)

GetParentSoftwareJobContainers3

The GetParentSoftwareJobContainers3 method retrieves the parent software job containers from a software job containers's parent group list.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved array of parent software job containers properties in the sequence SoftwareJobContainerProperties3. This sequence includes the userMessage parameter that specifies the custom administrator message information linked with the job container in SD.

Parameters

xsd:string sessionId (^m)

Specifies the session identifier obtained from the login method of the web service.

xsd:long listHandlesoftwareJobContainerId (^m)

Specifies a numeric handle to a list as returned by a previous call to OpenParentSoftwareJobContainerList3.

xsd:long noOfParentSoftwareJobContainers (^m)

Specifies the total number of software job containers to get from the list. This number must be less than or equal to the total number in the list as returned by a previous call to OpenParentSoftwareJobContainerList3.

Return Values

ArrayOfSoftwareJobContainerProperties3

arrayOfSoftwareJobContainerProperties3

Returns an array of SoftwareJobContainerProperties3—one for each parent container requested from the list.

Remarks

None

More information:

[SoftwareJobContainerProperties3](#) (see page 816)

GetSoftwareJobContainer

The GetSoftwareJobContainer method retrieves a software job container identified by its UUID.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwareJobContainerId (^m)

The UUID of the software job container to retrieve.

**SoftwareJobContainerPropertiesRequired
softwareJobContainerPropertiesRequired (^m)**

This parameter specifies which software job container properties to retrieve.

Return Values

SoftwareJobContainerProperties GetSoftwareJobContainerReturn

The software job container properties retrieved.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired](#) (see page 818)

GetSoftwareJobContainer3

The GetSoftwareJobContainer3 method retrieves a software job container identified by its UUID.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved software job container properties in the sequence SoftwareJobContainerProperties3. This sequence includes the userMessage parameter that specifies the custom administration message information linked with the job container in SD.

Parameters

xsd:string sessionId ("")

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwareJobContainerId ("")

Retrieves the UUID of the software job container.

SoftwareJobContainerPropertiesRequired3 softwareJobContainerPropertiesRequired3 ("")

Specifies which software job container properties to retrieve.

Return Values

SoftwareJobContainerProperties3 softwareJobContainerProperties3

Retrieves the software job container properties.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired3](#) (see page 819)

GetSoftwareJobContainers

The `GetSoftwareJobContainers` method retrieves the job containers from a software job container list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to `OpenSoftwareJobContainerList`.

xsd:long noOfSoftwareJobContainers ("")

The total number of software job containers to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwareJobContainerList`.

Return Values

ArrayOfSoftwareJobContainerProperties

GetSoftwareJobContainersReturn

An array of `SoftwareJobContainerProperties` - one for each container requested from the list.

Remarks

You can make multiple calls to `GetSoftwareJobContainers` to iterate down the list.

More information:

[SoftwareJobContainerProperties](#) (see page 786)

GetSoftwareJobContainers3

The `GetSoftwareJobContainers3` method retrieves the job containers from a software job container list.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved array of software job container properties in the sequence `SoftwareJobContainerProperties3`. The sequence includes the parameter `userMessage` that specifies the custom administrator message information linked with the job container in SD.

Parameters

xsd:string sessionId (")

Specifies the session identifier obtained from the login method of the web service.

xsd:long listHandle softwareJobContainerId (")

Specifies a numeric handle to a list as returned by a previous call to `OpenSoftwareJobContainerList3`.

xsd:long noOfSoftwareJobContainers (")

Specifies the total number of software job containers to get from the list. The number should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwareJobContainerList3`.

Return Values

ArrayOfSoftwareJobContainerProperties3

arrayOfSoftwareJobContainerProperties3

Returns an array of `SoftwareJobContainerProperties3`—one for each container requested from the list.

Remarks

None

More information:

[SoftwareJobContainerProperties3](#) (see page 816)

GetSoftwareJobContainerSoftwareJobs

The GetSoftwareJobContainerSoftwareJobs method retrieves the software jobs from a software job container's job list.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list as returned by a previous call to OpenSoftwareJobContainerSoftwareJobList.

xsd:long noOfSoftwareJobs (^m)

The total number of jobs to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobContainerSoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties

GetSoftwareJobContainerSoftwareJobsReturn

An array of SoftwareJobProperties - one for each job requested from the list.

Remarks

You can make multiple calls to GetSoftwareJobContainerSoftwareJobs to iterate down the list.

More information:

[SoftwareJobProperties](#) (see page 768)

GetSoftwareJobContainerSoftwareJobs3

The GetSoftwareJobContainerSoftwareJobs3 method retrieves the software jobs from a job list of a software job container.

Additionally, this method retrieves the ShutdownAfterLast property that includes information on whether a target computer is to be shut down after completion of the last SD job. The method returns a retrieved list of software job properties (within the software job container) in the sequence SoftwareJobProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast.

This method also retrieves the userMessage property that includes the custom administrator message linked with the software job in SD. The sequence SoftwareJobProperties3 includes the option userMessage.

Parameters

xsd:string sessionId (")

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (")

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwareJobContainerSoftwareJobList.

xsd:long noOfSoftwareJobs (")

Specifies the total number of software jobs to get from the list. This number must be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobContainerSoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties3 arrayOfSoftwareJobProperties3

Returns an array of SoftwareJobProperties3—one for each job requested from the list.

Remarks

None

More information:

[SoftwareJobProperties3](#) (see page 774)

GetSoftwareJobContainersRecursive

The GetSoftwareJobContainersRecursive method retrieves the job containers from a software job container list.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list as returned by a previous call to OpenSoftwareJobContainersRecursive.

xsd:long noOfSoftwareJobContainers (^m)

The number of software job containers.

Return Values

ArrayOfSoftwareJobContainerProperties
GetSoftwareJobContainersRecursiveReturn

An array of SoftwareJobContainerProperties - one for each container requested from the list

Remarks

You can make multiple calls to GetSoftwareJobContainersRecursive to iterate down the list.

More information:

[SoftwareJobContainerProperties](#) (see page 786)

GetSoftwareJobContainersRecursive3

The GetSoftwareJobContainersRecursive3 method retrieves the job containers from a software job container list.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved array of software job container properties in the sequence SoftwareJobContainerProperties3. This sequence includes the parameter userMessage that specifies the custom administrator message information linked with the job container in SD.

Parameters

xsd:string sessionId ("")

Specifies the session identifier obtained from the login method of the web service.

xsd:long listHandle ("")

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwareJobContainersRecursive3.

xsd:long noOfSoftwareJobContainers ("")

Specifies the number of software job containers.

Return Values

ArrayOfSoftwareJobContainerProperties3

arrayOfSoftwareJobContainerProperties3

Returns an array of SoftwareJobContainerProperties3—one for each container requested from the list.

Remarks

None

More information:

[SoftwareJobContainerProperties3](#) (see page 816)

HaltDeliverySoftwareJobContainer

The HaltDeliverySoftwareJobContainer method halts the delivery of files to the scalability server for all the jobs inside the software job container identified by its UUID.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwareJobContainerId (^m)

The UUID identifying the software job container whose delivery should be halted.

Return Values

This is a void method that does not expose a return value.

Remarks

None

OpenParentSoftwareJobContainerList

The OpenParentSoftwareJobContainerList method opens a list of the software job containers that are parent containers of the given software job container.

The method returns the total number of software package procedure groups in the list, and a handle to the list that can be used for further calls to GetParentSoftwarePackageProcedureGroups

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwareJobContainerId (^m)

The UUID of the software job container whose parent containers should be listed.

**SoftwareJobContainerPropertiesRequired
softwareJobContainerPropertiesRequired (^m)**

The properties that will be required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software job containers that can be passed to GetParentSoftwareJobContainers

xsd:long totalNoOfParentSoftwareJobContainers

The total number of parent software job containers in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired](#) (see page 818)

OpenParentSoftwareJobContainerList3

The OpenParentSoftwareJobContainerList3 method opens a list of the software job containers that are parent containers of the given software job container. The method returns the total number of software package procedure groups in the list, and a handle to the list that you can use for further calls to GetParentSoftwareJobContainers3.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence SoftwareJobContainerPropertiesRequired3. This sequence uses a parameter to set the requirement for the userMessage field in the retrieved software job container properties.

Parameters

xsd:string sessionId ("")

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwareJobContainerId ("")

Specifies the software job container identifier. If no value is provided, the root container is opened

SoftwareJobContainerPropertiesRequired3 softwareJobContainerPropertiesRequired3 ("")

Specifies the properties that are required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle

Returns a numeric handle to a list of software job containers that you can pass to `GetParentSoftwareJobContainers3`.

xsd:long totalNoOfParentSoftwareJobContainers

Returns the total number of parent software job containers in the list for which the handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired3](#) (see page 819)

OpenSoftwareJobContainerList

The `OpenSoftwareJobContainerList` method opens a list of software job containers.

The method returns the total number of software job containers in the list, and a handle to the list that can be used for further calls to `GetParentSoftwarePackageGroups`

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwareJobContainerId (")

The software job container identifier. If not supplied, the root container is opened.

**SoftwareJobContainerPropertiesRequired
softwareJobContainerPropertiesRequired (")**

The properties that will be required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software job containers that can be passed to GetSoftwareJobContainers.

xsd:long totalNoOfSoftwareJobContainers

The total number of software job containers in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired](#) (see page 818)

OpenSoftwareJobContainerList3

The OpenSoftwareJobContainerList3 method opens a list of software job containers. The method returns the total number of software job containers in the list, and a handle to the list that you can use for further calls to GetParentSoftwarePackageGroups.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence SoftwareJobContainerPropertiesRequired3. The sequence uses a parameter to set the requirement for the userMessage field in the retrieved software job container properties.

Parameters

xsd:string sessionId ("")

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwareJobContainerId ("")

Specifies the UUID of the software job container to retrieve.

SoftwareJobContainerPropertiesRequired3 softwareJobContainerPropertiesRequired3 ("")

Specifies which software job container properties to retrieve.

Return Values

xsd:long OpenListHandle

Returns a numeric handle to a list of software job containers that you can pass to GetSoftwareJobContainers.

xsd:long totalNoOfSoftwareJobContainers

Returns the total number of software job containers in the list for which the handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired3](#) (see page 819)

OpenSoftwareJobContainerListRecursive

The OpenSoftwareJobContainerListRecursive method opens a list of software job containers.

The method returns the total number of software job containers in the list, and a handle to the list that can be used for further calls to GetParentSoftwarePackageGroups.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwareJobContainerId

The software job container identifier. If not supplied, the root container is opened.

**SoftwareJobContainerPropertiesRequired
softwareJobContainerPropertiesRequired (")**

The properties that will be required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle totalNoOfSoftwareJobContainers

The total number of software job containers in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired](#) (see page 818)

OpenSoftwareJobContainerListRecursive3

The OpenSoftwareJobContainerListRecursive3 method opens a list of software job containers. The method returns the total number of software job containers in the list, and a handle to the list that you can use for further calls to GetSoftwareJobContainersRecursive3.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence SoftwareJobContainerPropertiesRequired3 that includes a parameter to set the requirement for the userMessage field in the retrieved software job container properties.

Parameters

xsd:string sessionId ("")

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwareJobContainerId

Specifies the software job container identifier. If no value is provided, the root container is opened.

**SoftwareJobContainerPropertiesRequired3
softwareJobContainerPropertiesRequired3 ("")**

Specifies the properties that are required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle

Returns a numeric handle to a list of software job containers that you can pass to GetSoftwareJobContainersRecursive3.

xsd:long totalNoOfSoftwareJobContainers

Returns the total number of software job containers in the list for which the handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired3](#) (see page 819)

OpenSoftwareJobContainerSoftwareJobList

The OpenSoftwareJobContainerSoftwareJobList method opens a list of the software jobs that are contained by the given software job container.

The method returns the total number of software jobs in the list, and a handle to the list that can be used for further calls to GetSoftwareJobContainerSoftwareJobs

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwareJobContainerId ("")

The UUID of the software job container whose member jobs should be listed.

SoftwareJobPropertiesRequired softwareJobPropertiesRequired ("")

The properties that will be required when getting a job from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of jobs that can be passed to GetSoftwareJobContainerSoftwareJobs

xsd:long totalNoOfSoftwareJobs

The total number of jobs in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobPropertiesRequired](#) (see page 777)

ResumeDeliverySoftwareJobContainer

The ResumeDeliverySoftwareJobContainer method resumes the delivery of files to the scalability server for all the jobs inside a software job container, identified by its UUID, that is currently in a halted state.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwareJobContainerId ("")

The UUID identifying the software job container whose delivery should be resumed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SealAndBuildSoftwareJobContainer

The SealAndBuildSoftwareJobContainer method seals and builds a single software job container.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwareJobContainerId ("")

The UUID identifying the software job container to seal and build.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SealSoftwareJobContainers

The SealSoftwareJobContainers method seals one or more software job containers.

Parameters

xsd:string sessionId (^m)

The session identifier.

ArrayOfstring softwareJobContainerIds (^m)

An array of UUIDs identifying the software job containers to be sealed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetSoftwareJobContainer

The SetSoftwareJobContainer method updates the properties of a software job container.

Parameters

xsd:string sessionId (^m)

The session identifier.

SetSoftwareJobContainerProperties softwareJobContainer (^m)

The properties that should be updated for the specified software job container.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareJobContainerProperties](#) (see page 739)

SetSoftwareJobContainer3

The SetSoftwareJobContainer3 method updates the properties of a software job container.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence CreateSoftwareJobContainerProperties3. This sequence includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (m)

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwareJobContainerId (m)

Specifies the UUID of the container for which properties are to be modified.

SetSoftwareJobContainerProperties3 softwareJobContainer3 (m)

Specifies the properties required to create a software job container.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareJobContainerProperties3](#) (see page 740)

UnsealSoftwareJobContainers

The UnsealSoftwareJobContainers method unseals one or more software job containers. After unsealing, you can modify the container.

Parameters

xsd:string sessionId (")

The session identifier.

ArrayOfstring softwareJobContainerIds (")

An array of UUIDs identifying the software job containers to be unsealed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateSoftwareJobContainer4

The CreateSoftwareJobContainer4 method creates a software job container.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:CreateSoftwareJobContainerProperties4 properties (m)

Specifies properties required to create a software job container.

Return Values

xsd:string softwareJobContainerId

Returns the UUID of the new container, if a software job container was successfully created.

GetServiceDeskRequestURLForContainer

The GetServiceDeskRequestURLForContainer method returns the CA Service Desk request URL for a job container.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string jobContainerId (m)

Specifies the UUID of the software job container.

Return Values

xsd:string requestURL

Contains the CA Service Desk request URL for container.

Software Jobs

This section describes software jobs methods.

This section contains the following topics:

[CancelSoftwareJobs](#) (see page 242)
[CloseSoftwareJobList](#) (see page 242)
[CloseSoftwareJobUnitSoftwareJobList](#) (see page 243)
[CreateActivateSoftwareJob](#) (see page 243)
[CreateActivateSoftwareJob3](#) (see page 245)
[CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList](#) (see page 246)
[CreateActivateSoftwareJobInSoftwarePolicy](#) (see page 247)
[CreateActivateSoftwareJobInSoftwarePolicy3](#) (see page 248)
[CreateActivateSoftwareJobInSoftwarePolicyByName](#) (see page 249)
[CreateActivateSoftwareJobInSoftwarePolicyByName3](#) (see page 250)
[CreateConfigureSoftwareJob](#) (see page 251)
[CreateConfigureSoftwareJob3](#) (see page 253)
[CreateConfigureSoftwareJobInSoftwarePolicy](#) (see page 254)
[CreateConfigureSoftwareJobInSoftwarePolicy3](#) (see page 255)
[CreateConfigureSoftwareJobInSoftwarePolicyByName](#) (see page 256)
[CreateConfigureSoftwareJobInSoftwarePolicyByName3](#) (see page 258)
[CreateInstallSoftwareJob](#) (see page 259)
[CreateInstallSoftwareJob3](#) (see page 260)
[CreateInstallSoftwareJobInSoftwarePolicy](#) (see page 261)
[CreateInstallSoftwareJobInSoftwarePolicy3](#) (see page 262)
[CreateRenewSoftwareJob](#) (see page 263)
[CreateRenewSoftwareJob3](#) (see page 264)
[CreateRecoverSoftwareJob](#) (see page 265)
[CreateRecoverSoftwareJob3](#) (see page 266)
[CreateUninstallSoftwareJob](#) (see page 267)
[CreateUninstallSoftwareJob3](#) (see page 268)
[CreateUninstallSoftwareJobInSoftwarePolicy](#) (see page 269)
[CreateUninstallSoftwareJobInSoftwarePolicy3](#) (see page 271)
[CreateUnInstallSoftwareJobInSoftwarePolicyByName](#) (see page 272)
[CreateUnInstallSoftwareJobInSoftwarePolicyByName3](#) (see page 273)
[DeleteSoftwareJobsInComputer](#) (see page 274)
[DeleteSoftwareJobsInComputerByName](#) (see page 275)
[FindSoftwareJob](#) (see page 275)
[GetSoftwareJob](#) (see page 276)
[GetSoftwareJob3](#) (see page 277)
[GetSoftwareJobs](#) (see page 278)
[GetSoftwareJobs3](#) (see page 279)
[GetSoftwareJobUnitSoftwareJobs](#) (see page 280)
[GetSoftwareJobUnitSoftwareJobs2](#) (see page 280)
[OpenSoftwareJobList](#) (see page 281)
[OpenSoftwareJobUnitSoftwareJobList](#) (see page 282)
[ResumeSoftwareJobs](#) (see page 283)
[SetSoftwareJob](#) (see page 283)
[SetSoftwareJob2](#) (see page 284)
[SetSoftwareJob3](#) (see page 285)
[SuspendSoftwareJobs](#) (see page 286)

[GetSoftwareJobOutputFile](#) (see page 286)

[GetSoftwareJobOutputFile2](#) (see page 287)

CancelSoftwareJobs

The CancelSoftwareJobs method cancels the Software Jobs.

Parameters

xsd:string sessionId (")

The session identifier.

ArrayOfstring softwareJobIds (")

To cancel a Particular Job the Corresponding SoftwareJobIds needs to be passed in the ArrayOfString. At least one softwarejobids should be available.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwareJobList

The CloseSoftwareJobList method closes a software jobs list that was opened by OpenSoftwareJobList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwareJobList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwareJobUnitSoftwareJobList

The CloseSoftwareJobUnitSoftwareJobListResponse method closes a unit software jobs list that was opened by OpenSoftwareJobUnitSoftwareJobListResponse and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwareJobUnitSoftwareJobListResponse.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateActivateSoftwareJob

The CreateActivateSoftwareJob method creates a new software Activate job.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageProcedureId (")

The UUID identifying the software package procedure that should be used by the job.

xsd:string softwarePackageInstallProcedureId (^m)

The UUID identifying the software package install procedure that should be used by the job.

CreateSoftwareJobOrderProperties softwareJobOrderProperties (^m)

The properties required to create and order a software job.

xsd:string softwareJobContainerId (^m)

Specifies the UUID identifying the job container associated with this new software job.

ArrayOfstring softwarePackageInstallationIds

Specifies an array of UUIDs identifying the installation jobs on which this activation job must be based. The targets from the given installation jobs are used, so do not supply the unitGroupIds and unitIds parameters.

ArrayOfstring unitGroupIds

An array of UUIDs identifying the unit groups to which the job should be applied.

ArrayOfstring unitIds

An array of UUIDs identifying the units to which the job should be applied.

Return Values

xsd:string CreateActivateSoftwareJobReturn

If a software Activate job was successfully created, the UUID of the new job is returned.

Remarks

If softwarePackageInstallationIds is not supplied, you must supply unitId or UnitGroupid.

More information:

[CreateSoftwareJobOrderProperties](#) (see page 685)

CreateActivateSoftwareJob3

The CreateActivateSoftwareJob3 method creates a new software job of type Activate.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

xsd:string softwarePackageInstallProcedureId ("")

Specifies the UUID identifying the software package install procedure that the job must use.

CreateSoftwareJobOrderProperties3 softwareJobOrderProperties3 ("")

Specifies the properties required to create and order a software job.

xsd:string softwareJobContainerId ("")

Specifies the UUID identifying the job container associated with this new software job.

ArrayOfstring softwarePackageInstallationIds

Specifies an array of UUIDs identifying the installation jobs on which this activation job must be based. The targets from the given installation jobs are used, so do not supply the unitGroupIds and unitIds parameters.

ArrayOfstring unitGroupIds

Specifies an array of UUIDs identifying the unit groups to which the job must be applied.

ArrayOfstring unitIds

Specifies an array of UUIDs identifying the units to which the job must be applied.

Return Values

xsd:string CreateActivateSoftwareJobReturn

Returns the UUID of the new job if a software Activate job is successfully created.

Remarks

None

More information:

[CreateSoftwareJobOrderProperties3](#) (see page 687)

CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList

The CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList method closes a software package procedure list that was opened by OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[CreateSoftwarePackageProcedureProperties](#) (see page 644)

CreateActivateSoftwareJobInSoftwarePolicy

The CreateActivateSoftwareJobInSoftwarePolicy method creates a new Activate software job within a given software policy.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageProcedureId ("")

The UUID identifying the software package procedure that should be used by the job.

xsd:string softwarePackageInstallProcedureId ("")

The UUID identifying the software package install procedure that should be used by the job.

CreateSoftwareJobOrderPolicyProperties

softwareJobOrderPolicyProperties ("")

The properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId ("")

The UUID identifying the software policy in which the software job should be created.

Return Values

xsd:string createActivateSoftwareJobInSoftwarePolicy

If an Activate software job was successfully created, the UUID of the new job is returned.

Remarks

None

More information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateActivateSoftwareJobInSoftwarePolicy3

The CreateActivateSoftwareJobInSoftwarePolicy3 method creates a new software job of type Activate within a given software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderPolicyProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderPolicyProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

xsd:string softwarePackageInstallProcedureId ("")

Specifies the UUID identifying the software package install procedure that the job must use.

**CreateSoftwareJobOrderPolicyProperties3
softwareJobOrderPolicyProperties3 ("")**

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId ("")

Specifies the UUID identifying the software policy in which you must create the software job.

Return Values

xsd:string createActivateSoftwareJobInSoftwarePolicy

Returns the UUID of the new job if an Activate software job is successfully created.

Remarks

None

More information:

[CreateSoftwareJobOrderPolicyProperties3](#) (see page 683)

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateActivateSoftwareJobInSoftwarePolicyByName

This method creates an activate software job in the named software policy.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

SoftwarePackage softwarePackage (m)

Specifies the software package

xsd:string softwarePackageActivateProcedureName (m)

Specifies the name of the software package procedure of type activate.

xsd:string softwarePackageInstallProcedureName (m)

Specifies the install software package procedure name.

CreateSoftwareJobOrderPolicyProperties**createSoftwareJobOrderPolicyProperties (m)**

Software job order policy properties are specified.

xsd:string softwarePolicyName (m)

Name of the Software policy.

Return Values

xsd:string configureSoftwareJobId

The UUID of the configure software job.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties3](#) (see page 683)

CreateActivateSoftwareJobInSoftwarePolicyByName3

The CreateActivateSoftwareJobInSoftwarePolicyByName3 method creates a software job of type Activate within a given software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderPolicyProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderPolicyProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters**xsd:string sessionId (m)**

Specifies the session identifier obtained from the login method of the web service.

SoftwarePackage softwarePackage (m)

Specifies the software package.

xsd:string softwarePackageProcedureName(mandatory)

Specifies the name of the software package procedure of type Activate.

xsd:string softwarePackageInstallProcedureName (m)

Specifies the install software package procedure name.

**CreateSoftwareJobOrderPolicyProperties3
softwareJobOrderPolicyProperties3 (m)**

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyName (m)

Specifies the name of the software policy in which the software job must be created.

Return Values

xsd:string softwareJobId

Returns the UUID of the new job if an Activate software job is successfully created.

Remarks

None

More Information:

[CreateSoftwareJobOrderProperties](#) (see page 685)

CreateConfigureSoftwareJob

The CreateConfigureSoftwareJob method creates a new software job of type 'Configure'.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageProcedureId (")

The UUID identifying the software package procedure that should be used by the job.

xsd:string softwarePackageInstallProcedureId (")

The UUID identifying the software package install procedure that should be used by the job.

CreateSoftwareJobOrderProperties softwareJobOrderProperties (")

The properties required to create and order a software job.

xsd:string softwareJobContainerId (")

The UUID identifying the job container associated with this new software job.

ArrayOfstring softwarePackageInstallationIds

An array of UUIDs identifying the installation jobs on which this activation job should be based. The targets from the given installation jobs are used so do not supply the unitGroupIds and unitIds parameters.

ArrayOfstring unitGroupIds

An array of UUIDs identifying the unit groups to which the job should be applied. Do not specify this parameter, if an array of softwarePackageInstallationIds has been given.

ArrayOfstring unitIds

An array of UUIDs identifying the units to which the job should be applied. Do not specify this parameter, if an array of softwarePackageInstallationIds has been given.

Return Values

xsd:string CreateConfigureSoftwareJobReturn

If a software Configure job was successfully created, the UUID of the new job is returned.

Remarks

If softwarePackageInstallationIds is Null, you must specify unitId or UnitGroupid.

More Information:

[CreateSoftwareJobOrderProperties3](#) (see page 687)

CreateConfigureSoftwareJob3

The CreateConfigureSoftwareJob3 method creates a new software job of type Configure.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

xsd:string softwarePackageInstallProcedureId ("")

Specifies the UUID identifying the software package install procedure that the job must use.

CreateSoftwareJobOrderProperties3 softwareJobOrderProperties3 ("")

Specifies the properties required to create and order a software job.

xsd:string softwareJobContainerId ("")

Specifies the UUID identifying the job container associated with this new software job.

ArrayOfstring softwarePackageInstallationIds

Specifies an array of UUIDs identifying the installation jobs on which this activation job must be based. The targets from the given installation jobs are used, so do not supply the unitGroupIds and unitIds parameters.

ArrayOfstring unitGroupIds

Specifies an array of UUIDs identifying the unit groups to which the job must be applied.

ArrayOfstring untids

Specifies an array of UUIDs identifying the units to which the job must be applied.

Return Values**xsd:string CreateConfigureSoftwareJobReturn**

Returns the UUID of the new job if a software Configure job is successfully created.

Remarks

None

More information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateConfigureSoftwareJobInSoftwarePolicy

The CreateConfigureSoftwareJobInSoftwarePolicy method creates a new software job of type 'Configure' within a given software policy.

Parameters**xsd:string sessionId ("")**

The session identifier.

xsd:string softwarePackageProcedureId ("")

The UUID identifying the software package procedure that should be used by the job.

xsd:string softwarePackageInstallProcedureId ("")

The UUID identifying the software package install procedure that should be used by the job.

**CreateSoftwareJobOrderPolicyProperties
softwareJobOrderPolicyProperties ("")**

The properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId ("")

The UUID identifying the software policy in which the software job should be created.

Return Values

xsd:string createConfigureSoftwareJobInSoftwarePolicy

If a 'Configure' software job was successfully created, the UUID of the new job is returned.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties3](#) (see page 683)

CreateConfigureSoftwareJobInSoftwarePolicy3

The CreateConfigureSoftwareJobInSoftwarePolicy3 method creates a new software job of type Configure within a given software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderPolicyProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderPolicyProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

xsd:string softwarePackageInstallProcedureId (m)

Specifies the UUID identifying the software package install procedure that the job must use.

**CreateSoftwareJobOrderPolicyProperties3
softwareJobOrderPolicyProperties3 (m)**

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId (m)

Specifies the UUID identifying the software policy in which you must create the software job.

Return Values

xsd:string createConfigureSoftwareJobInSoftwarePolicy

Returns the UUID of the new job if a Configure software job is successfully created.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateConfigureSoftwareJobInSoftwarePolicyByName

This method creates a configure software job for a specified software policy.

Parameters

xsd:string sessionId (m)

Session identifier got from the login method of the web service.

SoftwarePackage softwarePackage (m)

Specifies the software package properties.

xsd:string softwarePackageConfigureProcedureName (m)

Specifies the configure package procedure name.

xsd:string softwarePackageInstallProcedureName (m)

Specifies the install software package procedure name.

CreateSoftwareJobOrderPolicyProperties

createSoftwareJobOrderPolicyProperties (m)

Software job order policy properties are specified.

xsd:string softwarePolicyName (m)

Name of the software policy.

Return Values

xsd:string configureSoftwareJobId

The UUID of the job created.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties3](#) (see page 683)

CreateConfigureSoftwareJobInSoftwarePolicyByName3

The CreateConfigureSoftwareJobInSoftwarePolicyByName3 method creates a software job of type Configure for a specified software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderPolicyProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderPolicyProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters**xsd:string sessionId (m)**

Specifies the session identifier obtained from the login method of the web service.

SoftwarePackage softwarePackage (m)

Specifies the software package properties.

xsd:string softwarePackageProcedureName(mandatory)

Specifies the configure software package procedure name.

xsd:string softwarePackageInstallProcedureName (m)

Specifies the install software package procedure name.

CreateSoftwareJobOrderPolicyProperties3**softwareJobOrderPolicyProperties3 (m)**

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyName (m)

Specifies the name of the software policy in which the software job must be created.

Return Values

xsd:string softwareJobId

Returns the UUID of the new job if a Configure software job is successfully created.

Remarks

None

More Information:

[CreateSoftwareJobOrderProperties](#) (see page 685)

CreateInstallSoftwareJob

The CreateInstallSoftwareJob method creates a new software job of type 'Install'.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageProcedureId ("")

The UUID identifying the software package procedure that should be used by the job.

CreateSoftwareJobOrderProperties softwareJobOrder ("")

The properties required to create and order a software job.

xsd:string softwarejobContainerId ("")

The UUID identifying the job container associated with this new software job.

ArrayOfstring unitIds

An array of UUIDs identifying the units to which the job should be applied.

ArrayOfstring unitGroupIds

An array of UUIDs identifying the unit groups to which the job should be applied.

Return Values

xsd:string CreateInstallSoftwareJobReturn

If a software Install job was successfully created, the UUID of the new job is returned.

Remarks

You must specify UnitIds or UnitGroupids or both.

More Information:

[CreateInstallSoftwareJobOrderPolicyProperties3](#) (see page 634)

CreateInstallSoftwareJob3

The CreateInstallSoftwareJob3 method creates a new software job of type Install.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateInstallSoftwareJobOrderProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateInstallSoftwareJobOrderProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

CreateInstallSoftwareJobOrderProperties3
createInstallSoftwareJobOrderProperties3

Specifies the properties required to create and order a software job.

xsd:string softwareJobContainerId (")

Specifies the UUID identifying the job container associated with this new software job.

ArrayOfstring unitIds

Specifies an array of UUIDs identifying the units to which the job must be applied.

ArrayOfstring unitGroupIds

Specifies an array of UUIDs identifying the unit groups to which the job must be applied.

Return Values**xsd:string softwareJobId**

Returns the UUID of the new job if a software Install job is successfully created.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateInstallSoftwareJobInSoftwarePolicy

The CreateInstallSoftwareJobInSoftwarePolicy method creates a new software job of type 'Install' within a given software policy.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePackageProcedureId (")

The UUID identifying the software package procedure that should be used by the job.

**CreateSoftwareJobOrderPolicyProperties
softwareJobOrderPolicyProperties (^m)**

The properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId (^m)

The UUID identifying the software policy in which the software job should be created.

Return Values**xsd:string createInstallSoftwareJobInSoftwarePolicy**

If an 'Install' software job was successfully created, the UUID of the new job is returned.

Remarks

None

More Information:

[CreateInstallSoftwareJobOrderPolicyProperties3](#) (see page 634)

CreateInstallSoftwareJobInSoftwarePolicy3

The CreateInstallSoftwareJobInSoftwarePolicy3 method creates a new software job of type Install within a given software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateInstallSoftwareJobOrderPolicyProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateInstallSoftwareJobOrderPolicyProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters**xsd:string sessionId (^m)**

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId (^m)

Specifies the UUID identifying the software package procedure that the job must use.

CreateInstallSoftwareJobOrderPolicyProperties3
createInstallSoftwareJobOrderPolicyProperties3 (")

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId (")

Specifies the UUID identifying the software policy in which you must create the software job.

Return Values**xsd:string softwareJobId**

Returns the UUID of the new job if a software Install job is successfully created.

Remarks

None

CreateRenewSoftwareJob

The CreateRenewSoftwareJob method creates a new software 'Renew' job to re-evaluate and activate a job.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwareJobId (")

The UUID identifying the software job to be recovered.

xsd:string softwareJobContainerId (")

The UUID identifying the software job container to be recovered.

SoftwareJobOrderProperties softwareJobOrderProperties (")

The properties required to create and order a software job.

Return Values**xsd:string CreateRenewSoftwareJobReturn**

If a software Renew job was successfully created, the UUID of the new job is returned.

Remarks

None

More information:

[SoftwareJobOrderProperties](#) (see page 823)

CreateRenewSoftwareJob3

The CreateRenewSoftwareJob3 method creates a new software job of type Renew to re-evaluate and activate a job.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence SoftwareJobOrderProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence SoftwareJobOrderProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (")

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobId (")

Specifies the UUID identifying the software job that you want to recover.

xsd:string softwareJobContainerId (")

Specifies the UUID identifying the software job container that you want to recover.

SoftwareJobOrderProperties3 softwareJobOrderProperties3 (")

Specifies the properties required to create and order a software job.

Return Values

xsd:string CreateRenewSoftwareJobReturn

Returns the UUID of the new job if a software Renew job is successfully created.

Remarks

None

More information:

[SoftwareJobOrderProperties3](#) (see page 825)

CreateRecoverSoftwareJob

The CreateRecoverSoftwareJob method creates a new software 'Recover' job to clean up and recover a failed job.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwareJobId ("")

The UUID identifying the software job to be recovered.

xsd:string softwareJobContainerId ("")

The UUID identifying the software job container to be recovered.

xsd:string softwarePackageProcedureId ("")

The UUID identifying the software package procedure that should be used by the job.

SoftwareJobOrderProperties softwareJobOrderProperties ("")

The properties required to create and order a software job.

Return Values

xsd:string CreateRecoverSoftwareJobReturn

If a software Recover job was successfully created, the UUID of the new job is returned.

Remarks

None

More information:

[SoftwareJobOrderProperties](#) (see page 823)

CreateRecoverSoftwareJob3

The CreateRecoverSoftwareJob3 method creates a new software job of type Recover to clean up and recover a failed job.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence SoftwareJobOrderProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence SoftwareJobOrderProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobId ("")

Specifies the UUID identifying the software job to be recovered.

xsd:string softwareJobContainerId ("")

Specifies the UUID identifying the software job container to be recovered.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

SoftwareJobOrderProperties3 softwareJobOrderProperties3 ("")

Specifies the properties required to create and order a software job.

Return Values

xsd:string CreateRecoverSoftwareJobReturn

Returns the UUID of the new job if a software Recover job is successfully created.

Remarks

None

More information:

[SoftwareJobOrderProperties3](#) (see page 825)

CreateUninstallSoftwareJob

The CreateUninstallSoftwareJob method creates a new software job of type Uninstall.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageProcedureId (^m)

The UUID identifying the software package procedure that should be used by the job.

xsd:string softwarePackageInstallProcedureId (^m)

The UUID identifying the software package install procedure that should be used by the job.

CreateSoftwareJobOrderProperties softwareJobOrderProperties (^m)

The properties required to create and order a software job.

xsd:string softwareJobContainerId (^m)

The UUID identifying the job container associated with this new software job.

ArrayOfstring softwarePackageInstallationIds

An array of UUIDs identifying the installation jobs on which this activation job should be based. The targets from the given installation jobs will be used so the unitGroupIds and unitIds parameters should not be supplied.

ArrayOfstring unitGroupIds

An array of UUIDs identifying the unit groups to which the job should be applied. Do not supply this parameter if an array of softwarePackageInstallationIds has been given.

ArrayOfstring unitIds

An array of UUIDs identifying the units to which the job should be applied. Do not specify this parameter if an array of softwarePackageInstallationIds has been given.

Return Values

xsd:string CreateUninstallSoftwareJobReturn

If a software Uninstall job was successfully created, the UUID of the new job will be returned.

Remarks

If softwarePackageInstallationIds is not supplied, you must specify unitId or UnitGroupid.

More information:

[SoftwareJobOrderProperties](#) (see page 823)

CreateUninstallSoftwareJob3

The CreateUninstallSoftwareJob3 method creates a new software job of type Uninstall.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

xsd:string softwarePackageInstallProcedureId ("")

Specifies the UUID identifying the software package install procedure that the job must use.

CreateSoftwareJobOrderProperties3 softwareJobOrderProperties3 ("")

Specifies the properties required to create and order a software job.

xsd:string softwareJobContainerId ("")

Specifies the UUID identifying the job container associated with this new software job.

ArrayOfstring softwarePackageInstallationIds

Specifies an array of UUIDs identifying the installation jobs on which this activation job must be based. The targets from the given installation jobs will be used, so the unitGroupIds and unitIds parameters must not be supplied.

ArrayOfstring unitGroupIds

Specifies an array of UUIDs identifying the unit groups to which the job must be applied.

ArrayOfstring unitIds

Specifies an array of UUIDs identifying the units to which the job must be applied.

Return Values**xsd:string CreateUninstallSoftwareJobReturn**

Returns the UUID of the new job if a software Uninstall job is successfully created.

Remarks

None

More information:

[SoftwareJobOrderProperties3](#) (see page 825)

More Information:

[SoftwarePackageType](#) (see page 99)

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateUninstallSoftwareJobInSoftwarePolicy

This method creates uninstall software job in the specified software policy.

Parameters**xsd:string sessionId (^m)**

The session identifier.

xsd:string softwarePackageProcedureId (^m)

The UUID identifying the software package procedure that should be used by the job.

xsd:string softwarePackageInstallProcedureId (m)

The UUID identifying the software package install procedure that should be used by the job.

**CreateSoftwareJobOrderPolicyProperties
softwareJobOrderPolicyProperties (m)**

The properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId (m)

The UUID identifying the software policy in which the software job should be created.

SoftwarePackage softwarePackage (m)

Specifies the software package

xsd:string softwarePackageUninstallProcedureName (m)

Specifies the uninstall procedure name.

xsd:string softwarePackageInstallProcedureName (m)

Specifies the procedure name for install software package

**CreateSoftwareJobOrderPolicyProperties
createSoftwareJobOrderPolicyProperties (m)**

Software job order policy properties are specified.

xsd:string softwarePolicyName (m)

Name of the Software policy.

Return Values

xsd:string uninstallSoftwareJobId

The UUID of the job created.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties3](#) (see page 683)

CreateUninstallSoftwareJobInSoftwarePolicy3

This method creates a software job of type Uninstall in the specified software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence `CreateSoftwareJobOrderPolicyProperties3`, which includes another sequence `SoftwareJobBootMask3`. The `SoftwareJobBootMask3` sequence contains the option `ShutdownAfterLast` that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence `CreateSoftwareJobOrderPolicyProperties3` includes two parameters: `userMessage`, which takes the message input from the user, and `userMessageSupplied`, which checks the presence of the `userMessage` field for the custom administrator message in SD.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId ("")

Specifies the UUID identifying the software package procedure that the job must use.

xsd:string softwarePackageInstallProcedureId ("")

Specifies the UUID identifying the software package install procedure that the job must use.

**CreateSoftwareJobOrderPolicyProperties3
softwareJobOrderPolicyProperties3 ("")**

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId ("")

Specifies the UUID identifying the software policy in which you must create the software job.

Return Values

xsd:string createUninstallSoftwareJobInSoftwarePolicy

Returns the UUID of the new job if a software Uninstall job is successfully created.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateUnInstallSoftwareJobInSoftwarePolicyByName

The CreateUninstallSoftwareJobInSoftwarePolicyByName method creates a new software job of type Uninstall within a given software policy.

Parameters

xsd:string sessionId (m)

The session identifier.

dsm:softwarePackage softwarePackage (m)

The software package details that should be used by the job.

xsd:string softwarePackageProcedureName (m)

The name of the software package procedure that should be used by the job.

xsd:string softwarePackageInstallProcedureName (m)

The name of the software package install procedure that should be used by the job.

CreateSoftwareJobOrderPolicyProperties softwareJobOrderPolicyProperties (m)

The properties required to create and order a software job in a software policy.

xsd:string softwarePolicyName (m)

The name the software policy in which the software job should be created.

Return Values

xsd:string createUninstallSoftwareJobInSoftwarePolicy

If an 'Uninstall' software job was successfully created, the UUID of the new job will be returned.

Remarks

None

More Information:

[CreateSoftwareJobOrderPolicyProperties3](#) (see page 683)

CreateUnInstallSoftwareJobInSoftwarePolicyByName3

The CreateUninstallSoftwareJobInSoftwarePolicyByName3 method creates a new software job of type Uninstall within a given software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwareJobOrderPolicyProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateSoftwareJobOrderPolicyProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

SoftwarePackage softwarePackage (m)

Specifies the software package details that the job must use.

xsd:string softwarePackageName (m)

Specifies the name of the software package procedure that the job must use.

xsd:string softwarePackageInstallProcedureName (m)

Specifies the name of the software package install procedure that the job must use.

**CreateSoftwareJobOrderPolicyProperties3
softwareJobOrderPolicyProperties3 (m)**

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyName (m)

Specifies the name of the software policy in which you must create the software job.

Return Values

xsd:string softwareJobId

Returns the UUID of the new job if a software Uninstall job is successfully created.

Remarks

None

DeleteSoftwareJobsInComputer

Deletes a specified job from the computer. Jobs are specified by the uuids.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:Arrayofstring softwareJobIds (m)

The uuid of jobs to be removed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteSoftwareJobsInComputerByName

Removes jobs from a specified computer created by a software .

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string computerHostName (m)

The name of the computer whose job list is to be targeted for the job removal.

SoftwareByName softwareByName (m)

The software package details of the job to be removed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

FindSoftwareJob

The FindSoftwareJob method finds a software job given its procedure ID and install procedure ID.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePackageInstallProcedureId (m)

The UUID identifying the install procedure that it used for the job to be found.

xsd:string softwarePackageProcedureId (m)

The UUID identifying the procedure that is used for the job to be found.

Return Values

xsd:string softwareJobId

If a software job was found successfully, the UUID of the software job will be returned.

xsd:string softwareJobContainerId

If a software job was found successfully and that job is part of a job container, the UUID of the job's container will be returned.

Remarks

None

GetSoftwareJob

The GetSoftwareJob method retrieves a software job identified by its UUID

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwareJobId (^m)

The UUID of the software package to retrieve.

SoftwareJobPropertiesRequired softwareJobPropertiesRequired (^m)

This parameter specifies which software job properties to retrieve.

Return Values

SoftwareJobPropertiesGetSoftwareJobReturn

The software job properties retrieved.

Remarks

None

More information:

[SoftwareJobPropertiesRequired](#) (see page 777)

GetSoftwareJob3

The GetSoftwareJob3 method retrieves a software job identified by its UUID.

Additionally, this method retrieves the ShutdownAfterLast property that includes information on whether a target computer is to be shut down after completion of the last SD job. The method returns the retrieved software job properties in the sequence SoftwareJobProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast.

This method also retrieves the userMessage property that includes the custom administrator message linked with the software job in SD. The sequence SoftwareJobProperties3 includes the option userMessage.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobId ("")

Specifies the UUID of the software job that you want to update.

SoftwareJobPropertiesRequired softwareJobPropertiesRequired ("")

Specifies the software job properties to retrieve.

Return Values

SoftwareJobProperties3 softwareJobProperties3

Returns the retrieved software job properties.

Remarks

None

More information:

[SoftwareJobPropertiesRequired](#) (see page 777)

GetSoftwareJobs

The GetSoftwareJobs method retrieves the software jobs from a software job list.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list as returned by a previous call to OpenSoftwareJobList.

xsd:long noOfSoftwareJobs (^m)

The total number of software jobs to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties GetSoftwareJobsReturn

An array of SoftwareJobProperties - one for each software job requested from the list.

Remarks

You can make multiple calls to GetSoftwareJobs to iterate down the list.

More information:

[SoftwareJobProperties](#) (see page 768)

GetSoftwareJobs3

The GetSoftwareJobs3 method retrieves the software jobs from a software job list.

Additionally, this method retrieves the ShutdownAfterLast property that includes information on whether a target computer is to be shut down after completion of the last SD job. The method returns a retrieved list of software job properties in the sequence SoftwareJobProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast.

This method also retrieves the userMessage property that includes the custom administrator message linked with the software job in SD. The sequence SoftwareJobProperties3 includes the option userMessage.

Parameters

xsd:string sessionId (")

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (")

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwareJobList.

xsd:long noOfSoftwareJobs (")

Specifies the total number of software jobs to get from the list. The number must be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties3 arrayOfSoftwareJobProperties3

Returns an array of SoftwareJobProperties3—one for each software job requested from the list.

Remarks

None

More information:

[SoftwareJobProperties3](#) (see page 774)

GetSoftwareJobUnitSoftwareJobs

The GetSoftwareJobUnitSoftwareJobs method retrieves the unit software jobs (instances of a software job applied to a specified unit) from a software job's list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to OpenSoftwareJobUnitSoftwareJobList.

xsd:long noOfSoftwareJobUnits ("")

The total number of unit software jobs to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobUnitSoftwareJobList.

Return Values

ArrayOfUnitSoftwareJobProperties

GetSoftwareJobUnitSoftwareJobsReturn

An array of UnitSoftwareJobProperties, one for each unit software job requested from the list.

Remarks

You can make multiple calls to GetSoftwareJobUnitSoftwareJobs to iterate down the list.

More information:

[UnitSoftwareJobProperties](#) (see page 880)

GetSoftwareJobUnitSoftwareJobs2

The GetSoftwareJobUnitSoftwareJobs2 method retrieves the unit software jobs (instances of a software job applied to a specified unit) from a software job's list. This method also retrieves software package format (regular or virtual).

Parameters

xsd:string sessionId ("")

The session identifier

xsd:long listHandle ("")

A numeric handle to a software packages list, as returned by a previous call to OpenSoftwareJobUnitSoftwareJobList

xsd:long noOfSoftwareJobUnits (")

The total number of packages to get from the list. This number should be less than or equal to the total number in the list, as returned by a previous call to `OpenSoftwareJobUnitSoftwareJobList`.

Return Values**ArrayOfUnitSoftwareJobProperties2****GetSoftwareJobUnitSoftwareJobsReturn**

An array of `UnitSoftwareJobProperties` - one for each unit software job requested from the list

Remarks

You can make multiple calls to `GetSoftwareJobUnitSoftwareJobs2` to iterate down the list.

More information:

[UnitSoftwareJobProperties2](#) (see page 883)

OpenSoftwareJobList

The `OpenSoftwareJobList` opens a list of all software jobs.

Parameters**xsd:string sessionId (")**

The session identifier.

SoftwareJobPropertiesRequired softwareJobPropertiesRequired (")

The properties that will be required when getting a software job from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of software jobs that can be passed to `GetSoftwareJobs`

xsd:long totalNoOfSoftwareJobs

The total number of software jobs in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the `GetSoftwareJobs` method.

More information:

[SoftwareJobPropertiesRequired](#) (see page 777)

OpenSoftwareJobUnitSoftwareJobList

The OpenSoftwareJobUnitSoftwareJobList function opens a list of unit software jobs (instances of a software job applied to a specified unit) for a given software job.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwareJobId (^m)

The UUID identifying the software job whose unit instances should be listed.

unitSoftwareJobPropertiesRequired

UnitSoftwareJobPropertiesRequired (^m)

The properties that will be required when getting a unit software job from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of unit software jobs that can be passed to GetSoftwareJobUnitSoftwareJobs

xsd:long totalNoOfUnitSoftwareJobs

The total number of unit software jobs in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetSoftwareJobUnitSoftwareJobs method.

More information:

[UnitSoftwareJobPropertiesRequired](#) (see page 886)

ResumeSoftwareJobs

The ResumeSoftwareJobs method resumes the delivery of files to the scalability server for one or more software jobs that may have been suspended.

Parameters

xsd:string sessionId ("")

The session identifier.

ArrayOfstring softwarejobsIds ("")

An array of UUIDs identifying the software jobs to be resumed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetSoftwareJob

The SetSoftwareJob method updates the properties of a software job

Parameters

xsd:string sessionId ("")

The session identifier.

SetSoftwareJobProperties setSoftwareJobProperties ("")

The properties that should be updated for the specified software job.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareJobProperties](#) (see page 742)

SetSoftwareJob2

The SetSoftwareJob2 method updates the properties of a software job. The method takes the SoftwareJobOperationMask2 sequence, which has an extra flag reinstall. See the sequence SoftwareJobOperationMask2 for more details.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwareJobId (m)

The UUID of the software job to be updated.

SetSoftwareJobProperties2 setSoftwareJobProperties2 (m)

The properties that should be updated for the specified software job.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareJobProperties2](#) (see page 743)

SetSoftwareJob3

The SetSoftwareJob3 method updates the properties of a software job.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence SetSoftwareJobProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence SetSoftwareJobProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobId (m)

Specifies the UUID of the software job that you want to update.

SetSoftwareJobProperties3 setSoftwareJobProperties3 (m)

Specifies the properties that you must update for the specified software job.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareJobProperties3](#) (see page 744)

SuspendSoftwareJobs

The SuspendSoftwareJobs method suspends the delivery of files to the scalability server for one or more software jobs that may be in progress.

Parameters

xsd:string sessionId ("")

The session identifier.

ArrayOfstring softwarejobsIds ("")

An array of UUIDs identifying the software jobs to be suspended.

Return Values

This is a void method that does not expose a return value.

Remarks

None

GetSoftwareJobOutputFile

This method returns the path of Software job output file location, if any.

Parameters

xsd:string sessionId

Defines the session Id of the web service login.

xsd:string unitSoftwareJobId

Defines the UUID of the unitSoftwareJob which is obtained from *GetUnitUnitSoftwareJobs2* or *GetUnitUnitSoftwareInstallationJobs*.

Return Values

xsd:string filePath

Defines the location of job output file.

Xsd:Boolean isCopiedFile

Defines the value whether the job output file is a temporary file or the master file in the manager SD library. When Web Services is connected to a remote manager, the job output file is copied to the webservice machine. The webservice API user can delete this copied file after its usage. For more information, see [GetSoftwareJobOutputFile Location](#) (see page 287).

GetSoftwareJobOutputFile2

This method returns the path of Software job output file location, if any.

Parameters

xsd:string sessionId

Defines the session Id of the web service login.

xsd:string softwareJobId

Defines the UUID of the softwareJob.

xsd:string unitId

Defines the UUID of the computer or user profile to which the job is deployed.

xsd:int unitType

Specifies the type of unit.

0 for computer.

1 for user profile.

Return Values

xsd:string filePath

Defines the location of job output file.

Xsd:Boolean isCopiedFile

Defines the value whether the job output file is a temporary file or the master file in the manager SD library. When Web Services is connected to a remote manager, the job output file is copied to the webservicess machine. The webservice API user must delete this copied file after its usage. For more information, see [GetSoftwareJobOutputFile Location](#) (see page 287).

GetSoftwareJobOutputFile Location

The location job output file depends on the following conditions:

■ **WS is Local to Domain Manager:**

The filepath location is local. For example, *<DSMInstallDir Dir>\SD\ASM\LIBRARY\output\<UnitSoftwareJobID>.app\applic.res*

■ **WS is Remote to Domain Manager:**

- When NOS is configured on both the client and the manager machine, on Windows, the UNC path is returned. For example, *\\<Domain_Server_name>\SDLIBRARY\$\output\<UUID of application>.app\applic.res*.
- When NOS is NOT configured on either or both of the machines, the output file is copied to the temporary directory on webservice machine.

Software Package Groups

This section describes methods for software package groups.

This section contains the following topics:

[CloseParentSoftwarePackageGroupList](#) (see page 289)
[CloseSoftwarePackageGroupList](#) (see page 289)
[CloseSoftwarePackageGroupListRecursive](#) (see page 290)
[CloseSoftwarePackageGroupSoftwarePackageList](#) (see page 290)
[CloseSoftwarePackageGroupSoftwarePackageListRecursive](#) (see page 291)
[CreateCatalogSoftwarePackageGroup](#) (see page 291)
[CreateSoftwarePackageGroup](#) (see page 292)
[DeleteSoftwarePackageGroups](#) (see page 293)
[EmptySoftwarePackageGroups](#) (see page 293)
[FindSoftwarePackageGroup](#) (see page 294)
[GetParentSoftwarePackageGroups](#) (see page 295)
[GetSoftwarePackageGroup](#) (see page 295)
[GetSoftwarePackageGroups](#) (see page 296)
[GetSoftwarePackageGroupSoftwarePackages](#) (see page 297)
[GetSoftwarePackageGroupSoftwarePackages2](#) (see page 298)
[GetSoftwarePackageGroupSoftwarePackagesRecursive](#) (see page 299)
[GetSoftwarePackageGroupSoftwarePackagesRecursive2](#) (see page 299)
[GetSoftwarePackageGroupsRecursive](#) (see page 300)
[LinkSoftwarePackageGroupsToSoftwarePackageGroup](#) (see page 301)
[LinkSoftwarePackagesToSoftwarePackageGroup](#) (see page 301)
[LinkSoftwarePackagesToSoftwarePackageGroupByName](#) (see page 302)
[OpenParentSoftwarePackageGroupList](#) (see page 303)
[OpenSoftwarePackageGroupList](#) (see page 304)
[OpenSoftwarePackageGroupListRecursive](#) (see page 305)
[OpenSoftwarePackageGroupSoftwarePackageList](#) (see page 306)
[OpenSoftwarePackageGroupSoftwarePackageListRecursive](#) (see page 306)
[SetCatalogSoftwarePackageGroup](#) (see page 308)
[SetSoftwarePackageGroup](#) (see page 308)
[UnlinkSoftwarePackageGroupsFromSoftwarePackageGroup](#) (see page 309)
[UnlinkSoftwarePackagesFromSoftwarePackageGroupByName](#) (see page 309)

CloseParentSoftwarePackageGroupList

The CloseParentSoftwarePackageGroupList method closes a software package group list that was opened by OpenParentSoftwarePackageGroupList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenParentSoftwarePackageGroupList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePackageGroupList

The CloseSoftwarePackageGroupList method closes a software package group list that was opened by OpenSoftwarePackageGroupList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageGroupList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePackageGroupListRecursive

The CloseSoftwarePackageGroupListRecursive method closes a software package group list that was opened by OpenSoftwarePackageGroupListRecursive and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageGroupListRecursive.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePackageGroupSoftwarePackageList

The CloseSoftwarePackageGroupSoftwarePackageList method closes a software package list that was opened by OpenSoftwarePackageGroupSoftwarePackageList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageGroupSoftwarePackageList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePackageGroupSoftwarePackageListRecursive

The CloseSoftwarePackageGroupSoftwarePackageListRecursive method closes a software package list that was opened by OpenSoftwarePackageGroupSoftwarePackageListRecursive and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageGroupSoftwarePackageListRecursive.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateCatalogSoftwarePackageGroup

The CreateCatalogSoftwarePackageGroup method creates a catalog software package group.

Parameters

xsd:string sessionId (")

The session identifier.

**CreateCatalogSoftwarePackageGroupProperties
catalogSoftwarePackageGroup (")**

The properties required to create a catalog software package group.

Return Values

xsd:string CreateCatalogSoftwarePackageGroupReturn

If a catalog software package group was successfully created, the UUID of the new group is returned.

Remarks

None

More information:

[CreateCatalogSoftwarePackageGroupProperties](#) (see page 674)

CreateSoftwarePackageGroup

The CreateSoftwarePackageGroup method creates a software package group.

Parameters

xsd:string sessionId (^m)

The session identifier.

CreateSoftwarePackageGroupProperties

Specifies the software package group properties.

softwarePackageGroup (^m)

The properties required to create a software package group. This cannot be set to NULL.

Return Values

xsd:string CreateSoftwarePackageGroupReturn

If a software package group was successfully created, the UUID of the new group is returned.

Remarks

None

More information:

[CreateSoftwarePackageGroupProperties](#) (see page 689)

DeleteSoftwarePackageGroups

The DeleteSoftwarePackageGroups method deletes software package groups

Parameters

xsd:string sessionId ("")

The session identifier.

ArrayOfstring softwarePackageGroupIds ("")

An array of software package UUIDs identifying the software package groups to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

EmptySoftwarePackageGroups

The EmptySoftwarePackageGroups method empties one or more software package groups identified by their UUIDs. The members will be removed from the given group.

Parameters

xsd:string sessionId ("")

The session identifier.

ArrayOfstring softwarePackageGroupIds ("")

An array of UUIDs identifying the software package groups to be emptied.

Return Values

This is a void method that does not expose a return value.

Remarks

None

FindSoftwarePackageGroup

The FindSoftwarePackageGroup method finds a software package group given its name and the UUID and type of the parent software package group.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string parentSoftwarePackageGroupId

The UUID identifying the parent group of the software package group to search for.

SoftwarePackageGroupScope parentSoftwarePackageGroupScope (")

The type of the parent group of the software package group to search for.

xsd:string softwarePackageName (")

The name of the software package group to search for.

Return Values

xsd:string FindSoftwarePackageGroupReturn

If a software package group is found its UUID is returned.

Remarks

The group name must be specified exactly. This method does not allow wildcards. The group name is case insensitive.

If the parent software package Group UUID is not supplied, this method finds the software package group under Software Package Group superfolder (for example, Software Library).

More information:

[SoftwarePackageGroupScope](#) (see page 95)

GetParentSoftwarePackageGroups

The GetParentSoftwarePackageGroups method retrieves the parent software package groups from a software package group's parent group list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenParentSoftwarePackageGroupList.

xsd:long noOfParentSoftwarePackageGroups (")

The total number of software package groups to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenParentSoftwarePackageGroupList.

Return Values

ArrayOfSoftwarePackageGroupProperties

GetParentSoftwarePackageGroupsReturn

An array of SoftwarePackageGroupProperties - one for each parent group requested from the list.

Remarks

You can make multiple calls to GetParentSoftwarePackageGroups to iterate down the list.

More information:

[SoftwarePackageGroupProperties](#) (see page 828)

GetSoftwarePackageGroup

The GetSoftwarePackageGroup method retrieves a software package group identified by its UUID.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageGroupId (")

The UUID of the software package group to retrieve.

**SoftwarePackageGroupPropertiesRequired
softwarePackageGroupPropertiesRequired (")**

This parameter specifies which software package group properties to retrieve.

Return Values

SoftwarePackageGroupProperties GetSoftwarePackageGroupReturn

The software package group properties retrieved.

Remarks

None

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

GetSoftwarePackageGroups

The GetSoftwarePackageGroups method retrieves the software package groups from a software package group list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageGroupList.

xsd:long noOfSoftwarePackageGroups (")

The total number of software package groups to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageGroupList.

Return Values

ArrayOfSoftwarePackageGroupProperties

GetSoftwarePackageGroupsReturn

An array of SoftwarePackageGroupProperties - one for each group requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageGroups to iterate down the list.

More information:

[SoftwarePackageGroupProperties](#) (see page 828)

GetSoftwarePackageGroupSoftwarePackages

The GetSoftwarePackageGroupSoftwarePackages method retrieves the software packages from a software package group's package list

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageGroupSoftwarePackageList.

xsd:long noOfSoftwarePackages ("")

The total number of packages to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageGroupSoftwarePackageList.

Return Values

ArrayOfSoftwarePackageProperties

GetSoftwarePackageGroupSoftwarePackagesReturn

An array of SoftwarePackageProperties - one for each package requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageGroupSoftwarePackages to iterate down the list.

More information:

[SoftwarePackageProperties](#) (see page 847)

GetSoftwarePackageGroupSoftwarePackages2

The GetSoftwarePackageGroupSoftwarePackages2 method retrieves the software packages from a software package group's package list. This method also retrieves software package format (regular or virtual).

Parameters

xsd:string sessionId (")

The session identifier

xsd:long listHandle (")

A numeric handle to a software jobs list, as returned by a previous call to OpenSoftwarePackageGroupSoftwarePackageList.

xsd:long noOfSoftwarePackages (")

The total number of software installation jobs to get from the list. This number should be less than or equal to the total number in the list, as returned by a previous call to OpenSoftwarePackageGroupSoftwarePackageList.

Return Values

ArrayOfSoftwarePackageProperties2

GetSoftwarePackageGroupSoftwarePackagesReturn

An array of SoftwarePackageProperties - one for each package requested from the list

Remarks

You can make multiple calls to GetSoftwarePackageGroupSoftwarePackages2 to iterate down the list.

More information:

[SoftwarePackageProperties2](#) (see page 849)

GetSoftwarePackageGroupSoftwarePackagesRecursive

The GetSoftwarePackageGroupSoftwarePackagesRecursive method retrieves the software packages from a software package group's package list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageGroupSoftwarePackageListRecursive.

xsd:long noOfSoftwarePackages (")

The total number of packages to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageGroupSoftwarePackageListRecursive.

Return Values

ArrayOfSoftwarePackageProperties

GetSoftwarePackageGroupSoftwarePackagesRecursiveReturn

Remarks

You can make multiple calls to GetSoftwarePackageGroupSoftwarePackagesRecursive to iterate down the list.

More Information:

[ArrayOfSoftwarePackageProperties2](#) (see page 985)

GetSoftwarePackageGroupSoftwarePackagesRecursive2

The GetSoftwarePackageGroupSoftwarePackagesRecursive2 method retrieves the software packages from a software package group's package list. This method also retrieves software package format (regular or virtual).

Parameters

xsd:string sessionId (")

The session identifier

xsd:long listHandle (")

A numeric handle to a software packages list, as returned by a previous call to
OpenSoftwarePackageGroupSoftwarePackageListRecursive.

xsd:long noOfSoftwarePackages (")

The total number of packages to get from the list. This number should be less than or equal to the total number in the list, as returned by a previous call to
OpenSoftwarePackageGroupSoftwarePackageListRecursive.

Return Values

ArrayOfSoftwarePackageProperties2

GetSoftwarePackageGroupSoftwarePackagesRecursiveReturn

An array of SoftwarePackageProperties - one for each package requested from the list

Remarks

You can make multiple calls to
GetSoftwarePackageGroupSoftwarePackagesRecursive2 to iterate down the list.

GetSoftwarePackageGroupsRecursive

The GetSoftwarePackageGroupsRecursive method retrieves the software package groups from a software package group list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to
OpenSoftwarePackageGroupSoftwarePackageListRecursive.

xsd:long noOfSoftwarePackageGroups (")

The total number of software package groups to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to
OpenSoftwarePackageGroupListRecursive.

Return Values

ArrayOfSoftwarePackageGroupProperties

GetSoftwarePackageGroupsRecursiveReturn

An array of SoftwarePackageGroupProperties - one for each group requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageGroupsRecursive to iterate down the list. This API behaves identically to GetSoftwarePackageGroups.

LinkSoftwarePackageGroupsToSoftwarePackageGroup

The LinkSoftwarePackageGroupsToSoftwarePackageGroup method links one or more software package groups to a given software package group, making them members of the group.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageGroupId ("")

The UUID identifying the software package group to which the software package groups should be linked.

ArrayOfstring softwarePackageGroupIds ("")

An array of UUIDs identifying the software package groups that should be linked to the software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkSoftwarePackagesToSoftwarePackageGroup

The LinkSoftwarePackagesToSoftwarePackageGroup method links one or more software packages to a given software package group, making them members of the group.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageGroupId (")

The UUID identifying the software package group to which the software packages should be linked.

itrm:ArrayOfstring softwarePackageIds (")

An array of UUIDs identifying the software packages that should be linked to the software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkSoftwarePackagesToSoftwarePackageGroupByName

The LinkSoftwarePackagesToSoftwarePackageGroupByName method links one or more software packages to a software package group, making them members of the group.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwarePackageGroupName (m)

Specifies the package group name to which the software package is added.

ArrayOfSoftwarePackage softwarePackages (m)

Defines an array list of sequence SoftwarePackage. Each member of type SoftwarePackage contains details specifying a particular software package.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SoftwarePackage](#) (see page 865)

OpenParentSoftwarePackageGroupList

The OpenParentSoftwarePackageGroupList method opens a list of the software package groups that are parent groups of the given software package group.

The method returns the total number of software package groups in the list, and a handle to the list that can be used for further calls to GetParentSoftwarePackageGroups

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageGroupId ("")

The UUID of the software package group whose parent groups should be listed.

**SoftwarePackageGroupPropertiesRequired
softwarePackageGroupPropertiesRequired ("")**

The properties that will be required when getting a software package group from the list.

Return Values

xsd:long OpenListHandle ("")

A numeric handle to a list of software package groups that can be passed to GetParentSoftwarePackageGroups.

xsd:long totalNoOfParentSoftwarePackageGroups ("")

The total number of parent software package groups in the list whose handle has been returned.

Remarks

None

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

OpenSoftwarePackageGroupList

The OpenSoftwarePackageGroupList method opens a list of software package groups that are members of a given software package group (if there are any).

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageGroupId

The UUID identifying the software package group whose member groups should be listed. If not supplied root package group is opened

**SoftwarePackageGroupPropertiesRequired
softwarePackageGroupPropertiesRequired (^m)**

The properties that will be required when getting a software package group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package groups that can be passed to GetSoftwarePackageGroups

xsd:long totalNoOfSoftwarePackageGroups

The total number of software package groups in the list whose handle has been returned.

Remarks

If the softwarePackageGroupId is not supplied, the method will list the software package groups that are members of the superfolder (for example, Software Library).

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

OpenSoftwarePackageGroupListRecursive

The OpenSoftwarePackageGroupListRecursive method opens a list of software package groups, and of any software package groups recursively contained within that group, if there are any.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageGroupId

The software package group identifier. If not supplied root package group is opened.

**SoftwarePackageGroupPropertiesRequired
softwarePackageGroupPropertiesRequired (^m)**

The properties that will be required when getting a software package group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package groups that can be passed to GetSoftwarePackageGroups

xsd:long totalNoOfSoftwarePackageGroups

The total number of software package groups in the list whose handle has been returned.

Remarks

If the softwarePackageGroupId is not supplied, the method will list the software package groups that are members of the superfolder (for example, library).

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

OpenSoftwarePackageGroupSoftwarePackageList

The OpenSoftwarePackageGroupSoftwarePackageList method opens a list of the software packages that are members of the given software package group.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageGroupId (^m)

The UUID of the software package group whose member packages should be listed.

**SoftwarePackagePropertiesRequired
softwarePackagePropertiesRequired (^m)**

The properties that will be required when getting a package from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of packages that can be passed to GetSoftwarePackageGroupSoftwarePackages

xsd:long totalNoOfSoftwarePackages

The total number of packages in the list whose handle has been returned.

Remarks

The method returns the total number of software packages in the list, and a handle to the list that you can use for further calls to GetSoftwarePackageGroupSoftwarePackages.

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

OpenSoftwarePackageGroupSoftwarePackageListRecursive

The OpenSoftwarePackageGroupSoftwarePackageListRecursive method opens a list of the software packages that are members of the given software package group, and of any software groups contained within that group, expanding all member groups in the hierarchy recursively.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageGroupId (")

The UUID of the software package group whose member packages should be recursively listed.

**SoftwarePackagePropertiesRequired
softwarePackagePropertiesRequired (")**

The properties that will be required when getting a package from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of packages that can be passed to `GetSoftwarePackageGroupSoftwarePackagesRecursive`

xsd:long totalNoOfSoftwarePackages

The total number of packages in the list whose handle has been returned.

Remarks

The method returns the total number of software packages in the list, and a handle to the list that can be used for further calls to `GetSoftwarePackageGroupSoftwarePackages`.

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

SetCatalogSoftwarePackageGroup

The SetCatalogSoftwarePackageGroup method updates the properties of a catalog software package group

Parameters

xsd:string sessionId (")

The session identifier.

**SetCatalogSoftwarePackageGroupProperties
catalogSoftwarePackageGroup (")**

The properties that should be updated for the specified catalog software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetCatalogSoftwarePackageGroupProperties](#) (see page 729)

SetSoftwarePackageGroup

The SetSoftwarePackageGroup method updates the properties of a software package group

Parameters

xsd:string sessionId (")

The session identifier.

SetSoftwarePackageGroupProperties softwarePackageGroup (")

The properties that should be updated for the specified software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePackageGroupProperties](#) (see page 751)

UnlinkSoftwarePackageGroupsFromSoftwarePackageGroup

The `UnlinkSoftwarePackageGroupsFromSoftwarePackageGroup` method unlinks one or more software package groups from a given software package group.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePackageGroupId (m)

The UUID identifying the software package group from which the software package groups should be unlinked.

ArrayOfstring softwarePackageGroupIds (m)

An array of UUIDs identifying the software package groups that should be unlinked from the software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkSoftwarePackagesFromSoftwarePackageGroupByName

The `UnlinkSoftwarePackagesFromSoftwarePackageGroupByName` method unlinks one or more software packages from a software package group.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwarePackageName (m)

Specifies the name of an existing software package group from which to unlink the software packages.

ArrayOfSoftwarePackage softwarePackages (m)

An array list of sequence SoftwarePackage. Each member of type SoftwarePackage contains details specifying the software package to be unlinked.

Return Values

This is a void method that does not expose a return value.

Remarks

It does not delete the software package items completely; it only unlinks the items from the parent group.

More information:

[SoftwarePackage](#) (see page 865)

Software Package Procedure Dependencies

This section describes the software package procedure dependencies methods.

This section contains the following topics:

[CloseSoftwarePackageProcedureDependencyList](#) (see page 311)

[CreateSoftwarePackageProcedureDependency](#) (see page 311)

[DeleteSoftwarePackageProcedureDependency](#) (see page 312)

[GetSoftwarePackageProcedureDependencies](#) (see page 312)

[OpenSoftwarePackageProcedureDependencyList](#) (see page 313)

CloseSoftwarePackageProcedureDependencyList

The CloseSoftwarePackageProcedureDependencyList method closes a software package procedure dependency list that was opened by OpenSoftwarePackageProcedureDependencyList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageProcedureDependencyList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateSoftwarePackageProcedureDependency

The CreateSoftwarePackageProcedureDependency method creates a new software package procedure dependency.

Parameters

xsd:string sessionId ("")

The session identifier.

**CreateSoftwarePackageProcedureDependencyProperties
softwarePackageProcedureDependency ("")**

The properties required to create a software package procedure dependency.

Return Values

xsd:string CreateSoftwarePackageProcedureDependencyReturn

If a software package procedure dependency was successfully created, the UUID of the new dependency is returned.

Remarks

None

More information:

[CreateSoftwarePackageProcedureDependencyProperties](#) (see page 644)

DeleteSoftwarePackageProcedureDependency

The DeleteSoftwarePackageProcedureDependency method deletes one or more software package procedure dependencies identified by their UUIDs.

Parameters

xsd:string sessionId (^m)

The session identifier.

ArrayOfstring softwarePackageProcedureDepIds (^m)

An array of UUIDs that identify the software package procedure dependencies that should be deleted.

Return Values

This is a void method that does not expose a return value.

Remarks

None

GetSoftwarePackageProcedureDependencies

The GetSoftwarePackageProcedureDependencies method retrieves the dependencies from a software package procedure dependency list.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureDependencyList.

xsd:long noOfSoftwarePackageProcedureDependencies (")

The total number of procedure dependencies to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwarePackageProcedureDependencyList`.

Return Values**ArrayOfSoftwarePackageProcedureDependencyProperties
GetSoftwarePackageProcedureDependenciesReturn**

An array of `SoftwarePackageProcedureDependencyProperties` - one for each procedure requested from the list.

Remarks

You can make multiple calls to `GetSoftwarePackageProcedureDependencies` to iterate down the list.

More information:

[SoftwarePackageProcedureDependencyProperties](#) (see page 837)

OpenSoftwarePackageProcedureDependencyList

The `OpenSoftwarePackageProcedureDependencyList` opens a list of dependencies for a given software package procedure.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePackageProcedureId (")

the UUID identifying the software package procedure whose dependencies should be listed.

**SoftwarePackageProcedureDependencyPropertiesRequired
softwarePackageProcedureDependencyPropertiesRequired (")**

The properties that will be required when getting a procedure dependency from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of software packages that can be passed to `GetSoftwarePackageProcedureDependencies`

xsd:long totalNoOfSoftwarePackageProcedureDependencies

The total number of procedure dependencies in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the `GetSoftwarePackageProcedureDependencies` method.

More information:

[SoftwarePackageProcedureDependencyPropertiesRequired](#) (see page 838)

Software Package Procedures

This section describes Software Package Procedures methods.

This section contains the following topics:

[CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList](#) (see page 316)

[CloseSoftwarePackageProcedureList](#) (see page 317)

[CreateAddedSoftwarePackageProcedure](#) (see page 317)

[CreateAddedSoftwarePackageProcedure2](#) (see page 318)

[CreateAddedSoftwarePackageProcedure3](#) (see page 319)

[CreateSoftwarePackageProcedure](#) (see page 319)

[CreateSoftwarePackageProcedure3](#) (see page 320)

[CreateSoftwarePackageProcedure2](#) (see page 321)

[DeleteSoftwarePackageProcedureGroups](#) (see page 322)

[DeleteSoftwarePackageProcedures](#) (see page 322)

[EmptySoftwarePackageProcedureGroups](#) (see page 323)

[FindSoftwarePackageProcedure](#) (see page 323)

[FindSoftwarePackageProcedureGroup](#) (see page 324)

[GetParentSoftwarePackageProcedureGroups](#) (see page 325)

[GetSoftwarePackageProcedure](#) (see page 326)

[GetSoftwarePackageProcedure2](#) (see page 326)

[GetSoftwarePackageProcedureGroup](#) (see page 327)

[GetSoftwarePackageProcedureGroups](#) (see page 328)

[GetSoftwarePackageProcedureGroupSoftwarePackageProcedures](#) (see page 328)

[GetSoftwarePackageProcedureGroupSoftwarePackageProcedures2](#) (see page 329)

[GetSoftwarePackageProcedureGroupsRecursive](#) (see page 330)

[GetSoftwarePackageProcedures](#) (see page 331)

[GetSoftwarePackageProcedures2](#) (see page 332)

[LinkSoftwarePackageProcedureGroupsToSoftwarePackageGroup](#) (see page 333)

[LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroupByName](#) (see page 333)

[LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroup](#) (see page 334)

[LinkSoftwarePackagesToSoftwarePackageGroup](#) (see page 335)

[OpenParentSoftwarePackageProcedureGroupList](#) (see page 335)

[OpenSoftwarePackageProcedureGroupList](#) (see page 336)

[OpenSoftwarePackageProcedureGroupListRecursive](#) (see page 337)

[OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList](#) (see page 338)

[OpenSoftwarePackageProcedureList](#) (see page 339)

[RepositionSoftwarePackageProcedure](#) (see page 340)

[RepositionSoftwarePackageProcedureByName](#) (see page 341)

[SetSoftwarePackageProcedure](#) (see page 342)

[SetSoftwarePackageProcedure2](#) (see page 343)

[SetSoftwarePackageProcedure3](#) (see page 344)

[SetSoftwarePackageProcedureGroup](#) (see page 344)

[OpenSoftwarePackageProcedureGroupListRecursive](#) (see page 345)

[UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroup](#) (see page 346)

[GetSoftwarePackageProcedure3](#) (see page 347)

[GetSoftwarePackageProcedure4](#) (see page 348)

[GetSoftwarePackageProcedures3](#) (see page 349)

[GetSoftwarePackageProcedureList](#) (see page 350)

CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList

The

CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList method closes a software package procedure list that was opened by OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[CreateSoftwarePackageProcedureProperties](#) (see page 644)

CloseSoftwarePackageProcedureList

The CloseSoftwarePackageProcedure method closes a software package procedure list that was opened by OpenSoftwarePackageProcedureList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageProcedureList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateAddedSoftwarePackageProcedure

The CreateAddedSoftwarePackageProcedure method creates a new software package procedure. Files additional to those in the software package to which this procedure is associated may be added to the procedure definition.

Parameters

xsd:string sessionId (")

The session identifier.

**CreateSoftwarePackageProcedureProperties
softwarePackageProcedureProperties (")**

The properties required to create a software package procedure.

xsd:string sourcePath

The path to any files that the procedure requires in addition to those files in the software package to which the procedure is associated.

Return Values

xsd:string CreateAddedSoftwarePackageProcedureReturn

If an Added software package procedure was successfully created, the UUID of the new procedure is returned.

Remarks

None

More information:

[SoftwarePackageProcedureProperties](#) (see page 843)

CreateAddedSoftwarePackageProcedure2

The `CreateAddedSoftwarePackageProcedure2` method creates a new software package procedure. Files in addition to those that are in the software package (to which this procedure is associated) may be added to the procedure definition.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence `CreateSoftwarePackageProcedureProperties2`, which includes another sequence `SoftwarePackageProcedureBootMask2`. The `SoftwarePackageProcedureBootMask2` sequence contains the option `ShutdownAfterLast` that lets you shut down the target computer after completion of the jobs.

This method also lets you use the software package procedure comment as a default user message for the jobs created using this procedure.

Parameters

xsd:string sessionId (")

Specifies the session identifier obtained from the login method of the web service.

**CreateSoftwarePackageProcedureProperties2
softwarePackageProcedureProperties (")**

Specifies the properties required to create a software package procedure.

xsd:string sourcePath

Specifies the path to the files that the procedure requires in addition to those that are in the software package to which the procedure is associated.

Return Values

xsd:string createAddedSoftwarePackageProcedureReturn

Returns the UUID of the new procedure if a software package procedure is successfully created.

Remarks

None

More information:

[SoftwarePackageProcedureProperties2](#) (see page 845)

CreateAddedSoftwarePackageProcedure3

The CreateAddedSoftwarePackageProcedure3 method creates a new software package procedure. Files in addition to those that are in the software package (to which this procedure is associated) may be added to the procedure definition.

This method is provided to support “excludeFromRAC” option while creating the procedure, the existing API’s doesn’t have the option.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

**dsm:CreateSoftwarePackageProcedureProperties3
softwarePackageProcedureProperties**

Specifies the properties required to create a software package procedure.

xsd:string sourcePath

Specifies the path to the files that the procedure requires in addition to those that are in the software package to which the procedure is associated.

Return Values

xsd:string softwarePackageProcedureUUID

UUID of the newly created software procedure

CreateSoftwarePackageProcedure

The CreateSoftwarePackageProcedure method creates a new software package procedure.

Parameters

xsd:string sessionId (")

The session identifier.

**CreateSoftwarePackageProcedureProperties
softwarePackageProcedureProperties (")**

The properties required to create a software package procedure.

xsd:boolean externalProcedure

If not supplied, the default value is false.

Return Values

xsd:string createSoftwarePackageProcedureReturn

If a software package procedure was successfully created, the UUID of the new procedure is returned.

Remarks

None

More information:

[SoftwarePackageProcedureProperties](#) (see page 843)

CreateSoftwarePackageProcedure3

The CreateSoftwarePackageProcedure3 method creates a new software package procedure. This method is provided to support “excludeFromRAC” option while creating the procedure, the existing API’s doesn’t have the option.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

**dsm:CreateSoftwarePackageProcedureProperties3
softwarePackageProcedureProperties**

Specifies the properties required to create a software package procedure.

Return Values

xsd:string softwarePackageProcedureUUID

UUID of the newly created software procedure.

CreateSoftwarePackageProcedure2

The CreateSoftwarePackageProcedure2 method creates a new software package procedure.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateSoftwarePackageProcedureProperties2, which includes another sequence SoftwarePackageProcedureBootMask2. The SoftwarePackageProcedureBootMask2sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also lets you use the software package procedure comment as a default user message for the jobs created using this procedure.

Parameters

xsd:string sessionId (")

Specifies the session identifier obtained from the login method of the web service

**CreateSoftwarePackageProcedureProperties2
softwarePackageProcedureProperties (")**

Specifies the properties required to create a software package procedure.

Return Values

xsd:string createSoftwarePackageProcedureReturn

Returns the UUID the new procedure if a software package procedure is successfully created.

Remarks

None

More information:

[SoftwarePackageProcedureProperties2](#) (see page 845)

DeleteSoftwarePackageProcedureGroups

The DeleteSoftwarePackageProcedureGroups method deletes software package procedure groups

Parameters

xsd:string sessionId (^m)

The session identifier.

ArrayOfstring softwarePackageProcedureGroupIds (^m)

An array of software package UUIDs identifying the software package procedure groups to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteSoftwarePackageProcedures

The DeleteSoftwarePackageProcedures method deletes one or more software package procedures identified by their UUIDs.

Parameters

xsd:string sessionId (^m)

The session identifier.

ArrayOfstring softwarePackageProcedureIds (^m)

An array of UUIDs that identify the software package procedures that should be deleted.

Return Values

This is a void method that does not expose a return value.

Remarks

None

EmptySoftwarePackageProcedureGroups

The EmptySoftwarePackageProcedureGroups method empties one or more software package procedure group(s) identified by their UUIDs. The members will be removed from the given group.

Parameters

xsd:string sessionId (")

The session identifier.

ArrayOfstring softwarePackageProcedureGroupIds (")

An array of UUIDs identifying the software package procedure groups to be emptied.

Return Values

This is a void method that does not expose a return value.

Remarks

None

FindSoftwarePackageProcedure

The FindSoftwarePackageProcedure method finds a software package procedure given its name and task type, and given the name and version of the software package with which it is associated.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageName (")

The name of the software package to which the requested procedure should belong.

xsd:string softwarePackageVersion (")

The version of the software package to which the requested procedure should belong.

xsd:string softwarePackageProcedureName (")

The name of the software package procedure to find.

SoftwarePackageProcedureTask softwarePackageProcedureTask

The type of the software package procedure to find.

Return Values

xsd:string FindSoftwarePackageProcedureReturn

If a software package procedure is found its UUID is returned.

Remarks

The names and versions must be specified exactly. This method does not allow wildcards. The software package and procedure names are case insensitive.

More information:

[SoftwarePackageProcedureTask](#) (see page 97)

FindSoftwarePackageProcedureGroup

The FindSoftwarePackageProcedureGroup method finds a software package procedure group given its name and the UUID and type of the parent software package group.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string parentSoftwarePackageProcedureGroupId

The UUID identifying the parent group of the software package procedure group to search for.

SoftwarePackageProcedureGroupScope parentSoftwarePackageProcedureGroupScope (")

The type of the parent group of the software package procedure group to search for.

xsd:string softwarePackageProcedureGroupName (")

The name of the software package procedure group to search for.

Return Values

xsd:string FindSoftwarePackageProcedureGroupReturn

If a software package procedure group is found its UUID is returned.

Remarks

The group name must be specified exactly. This method does not allow wildcards. The group name is case insensitive.

More information:

[SoftwarePackageProcedureGroupScope](#) (see page 96)

GetParentSoftwarePackageProcedureGroups

The GetParentSoftwarePackageProcedureGroups method retrieves the parent software package procedure groups from a software package procedure group's parent group list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenParentSoftwarePackageProcedureGroupList.

xsd:long noOfParentSoftwarePackageGroups (")

The total number of software package procedure groups to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenParentSoftwarePackageProcedureGroupList.

Return Values

ArrayOfSoftwarePackageGroupProperties

GetParentSoftwarePackageProcedureGroupsReturn

An array of SoftwarePackageProcedureGroupProperties - one for each parent group requested from the list.

Remarks

You can make multiple calls to GetParentSoftwarePackageProcedureGroups to iterate down the list.

More information:

[SoftwarePackageGroupProperties](#) (see page 828)

GetSoftwarePackageProcedure

The GetSoftwarePackageProcedure method retrieves a software package procedure identified by its UUID.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageProcedureId ("")

The UUID of the software package procedure to retrieve.

**SoftwarePackageProcedurePropertiesRequired
softwarePackageProcedurePropertiesRequired ("")**

This parameter specifies which software package procedure properties to retrieve.

Return Values

**SoftwarePackageProcedureProperties
GetSoftwarePackageProcedureReturn**

The software package procedure properties retrieved.

Remarks

None

More information:

[SoftwarePackageProcedurePropertiesRequired](#) (see page 833)

GetSoftwarePackageProcedure2

The GetSoftwarePackageProcedure2 method retrieves a software package procedure that its UUID identifies.

This method also retrieves the ShutdownAfterLast property (that includes information on whether a target computer is to be shut down after completion of the last SD job) and useDefaultJobMessage property (that includes information on whether a software package procedure comment is to be used as a default user message for the jobs).

Parameters

xsd:string sessionId ("")

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwarePackageProcedureId ("")

Retrieves the UUID of the software package procedure.

SoftwarePackageProcedurePropertiesRequired
softwarePackageProcedurePropertiesRequired (")

Retrieves the software package procedure properties.

Return Values**SoftwarePackageProcedureProperties2**
GetSoftwarePackageProcedureReturn

Returns the retrieved software package procedure properties.

Remarks

None

More information:

[SoftwarePackageProcedurePropertiesRequired](#) (see page 833)

GetSoftwarePackageProcedureGroup

The GetSoftwarePackageProcedureGroup method retrieves a software package procedure group identified by its UUID.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePackageProcedureGroupId (")

The UUID of the software package procedure group to retrieve.

SoftwarePackageProcedureGroupPropertiesRequired
softwarePackageProcedureGroupPropertiesRequired (")

This parameter specifies which software package procedure group properties to retrieve.

Return Values**SoftwarePackageProcedureGroupProperties**
GetSoftwarePackageProcedureGroupReturn

The software package procedure group properties retrieved.

Remarks

None

More information:

[SoftwarePackageProcedureGroupPropertiesRequired](#) (see page 839)

GetSoftwarePackageProcedureGroups

The GetSoftwarePackageProcedureGroups method retrieves the procedure groups from a software package procedure group list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureGroupList.

xsd:long noOfSoftwarePackageProcedureGroups (")

The total number of software package procedure groups to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageProcedureGroupList.

Return Values

ArrayOfSoftwarePackageProcedureGroupProperties

GetSoftwarePackageProcedureGroupsReturn

An array of SoftwarePackageProcedureGroupProperties - one for each group requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageProcedureGroups to iterate down the list.

More information:

[SoftwarePackageProcedureGroupProperties](#) (see page 838)

GetSoftwarePackageProcedureGroupSoftwarePackageProcedures

The GetSoftwarePackageProcedureGroupSoftwarePackageProcedures method retrieves the software package procedures from a software package procedure group's procedure list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList.

xsd:long noOfSoftwarePackageProcedures (")

The total number of procedures to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList`.

Return Values**ArrayOfSoftwarePackageProcedureProperties****GetSoftwarePackageProcedureGroupSoftwarePackageProceduresReturn**

An array of `SoftwarePackageProcedureProperties` - one for each procedure requested from the list.

Remarks

You can make multiple calls to `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures` to iterate down the list.

More information:

[SoftwarePackageProcedureProperties](#) (see page 843)

GetSoftwarePackageProcedureGroupSoftwarePackageProcedures2

The `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures2` method retrieves the software package procedures from a software package procedure group's procedure list.

This method also retrieves the `ShutdownAfterLast` property (that includes information on whether a target computer is to be shut down after completion of the last SD job) and `useDefaultJobMessage` property (that includes information on whether a software package procedure comment is to be used as a default user message for the jobs).

Parameters**xsd:string sessionId (")**

Specifies the session identifier obtained from the login method of the web service.

xsd:long listHandle (")

Specifies a numeric handle to a list as returned by a previous call to `OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList`.

xsd:long noOfSoftwarePackageProcedures (")

Specifies the total number of procedures to get from the list. The number must be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList`.

Return Values**ArrayOfSoftwarePackageProcedureProperties2****GetSoftwarePackageProcedureGroupSoftwarePackageProceduresReturn**

Returns an array of `SoftwarePackageProcedureProperties2`—one for each procedure requested from the list.

Remarks

None

More information:

[SoftwarePackageProcedureProperties2](#) (see page 845)

GetSoftwarePackageProcedureGroupsRecursive

The `GetSoftwarePackageProcedureGroupsRecursive` method retrieves the procedure groups from a software package procedure group list.

Parameters**xsd:long listHandle (")**

A numeric handle to a list as returned by a previous call to `OpenSoftwarePackageProcedureGroupListRecursive`.

xsd:long noOfSoftwarePackageProcedureGroups (")

The total number of software package procedure groups in the list whose handle has been returned.

Return Values**ArrayOfSoftwarePackageProcedureGroupProperties****GetSoftwarePackageProcedureGroupsReturn**

An array of `SoftwarePackageProcedureGroupProperties` - one for each group requested from the list.

Remarks

You can make multiple calls to `GetSoftwarePackageProcedureGroupsRecursive` to iterate down the list.

More information:

[SoftwarePackageProcedureGroupProperties](#) (see page 838)

GetSoftwarePackageProcedures

The GetSoftwarePackageProcedures method retrieves the procedures from a software package procedure list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureList.

xsd:long noOfSoftwarePackageProcedures ("")

The total number of procedures to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageProcedureList.

Return Values

**ArrayOfSoftwarePackageProcedureProperties
getSoftwarePackageProcedures**

An array of SoftwarePackageProcedureProperties - one for each procedure requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageProcedures to iterate down the list.

More information:

[SoftwarePackageProcedureProperties](#) (see page 843)

GetSoftwarePackageProcedures2

The GetSoftwarePackageProcedures2 method retrieves the procedures from a software package procedure list.

This method also retrieves the ShutdownAfterLast property (that includes information on whether a target computer is to be shut down after completion of the last SD job) and useDefaultJobMessage property (that includes information on whether a software package procedure comment is to be used as a default user message for the jobs).

Parameters

xsd:string sessionId ("")

Specifies the session identifier obtained from the login method of the web service.

xsd:long listHandle ("")

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureList.

xsd:long noOfSoftwarePackageProcedures ("")

Specifies the total number of procedures to get from the list. The number must be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageProcedureList.

Return Values

**ArrayOfSoftwarePackageProcedureProperties2
getSoftwarePackageProcedures**

Returns an array of SoftwarePackageProcedureProperties2—one for each procedure requested from the list.

Remarks

None

More information:

[SoftwarePackageProcedureProperties2](#) (see page 845)

LinkSoftwarePackageProcedureGroupsToSoftwarePackageGroup

The LinkSoftwarePackageProcedureGroupsToSoftwarePackageGroup method links one or more software package procedure group(s) to a given software package group, making them members of the group.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePackageGroupId (m)

The UUID identifying the software package group to which the software package procedure groups should be linked.

ArrayOfstring softwarePackageProcedureGroupIds (m)

An array of UUIDs identifying the software package procedure groups that should be linked to the software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroupName

The LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroupName method links one or more software package procedures to a software package procedure group, making them members of the group.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwarePackageProcedureGroupName (m)

Specifies the package group name to which the software package procedure is to be added.

ArrayOfSoftwarePackageProcedure softwarePackageProcedures (m)

Defines a set of software package procedures to be linked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SoftwarePackageProcedure](#) (see page 865)

LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroup

The LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroup method links one or more software package procedure(s) to a given software package procedure group, making them members of the group.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageProcedureGroupId (")

The UUID identifying the software package procedure group to which the software package procedures should be linked.

ArrayOfstring softwarePackageProcedureIds (")

An array of UUIDs identifying the software package procedures that should be linked to the software package procedure group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkSoftwarePackagesToSoftwarePackageGroup

The LinkSoftwarePackagesToSoftwarePackageGroup method links one or more software packages to a given software package group, making them members of the group.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageGroupId (")

The UUID identifying the software package group to which the software packages should be linked.

itrm:ArrayOfstring softwarePackageIds (")

An array of UUIDs identifying the software packages that should be linked to the software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

OpenParentSoftwarePackageProcedureGroupList

The OpenParentSoftwarePackageProcedureGroupList method opens a list of the software package procedure groups that are parent groups of the given software package procedure group.

The method returns the total number of software package procedure groups in the list, and a handle to the list that can be used for further calls to GetParentSoftwarePackageProcedureGroups

All open functions return a list handle and total count of items. The items can be retrieved by using the corresponding Get Methods

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageProcedureGroupId ("")

The UUID of the software package procedure group whose parent groups should be listed.

SoftwarePackageGroupPropertiesRequired

softwarePackageProcedureGroupPropertiesRequired ("")

The properties that will be required when getting a software package procedure group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package procedure groups that can be passed to GetParentSoftwarePackageProcedureGroups

xsd:long totalNoOfParentSoftwarePackageGroups

The total number of parent software package procedure groups in the list whose handle has been returned.

Remarks

None

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

OpenSoftwarePackageProcedureGroupList

This method opens a list of software package procedure groups. The method returns the total number of software package procedure groups in the list, and a handle to the list that can be used for further calls to GetSoftwarePackageProcedureGroups

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageGroupId ("")

The UUID identifying the softwarePackageProcedureGroupPropertiesRequired.

SoftwarePackageProcedureGroupPropertiesRequired (")

The properties that will be required when getting a software package procedure group from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of software package procedure groups that can be passed to GetSoftwarePackageProcedureGroups

xsd:long totalNoOfSoftwarePackageProcedureGroups

The total number of software package procedure groups in the list whose handle has been returned.

Remarks

None

More information:

[SoftwarePackageProcedureGroupPropertiesRequired](#) (see page 839)

OpenSoftwarePackageProcedureGroupListRecursive

The OpenSoftwarePackageProcedureGroupListRecursive method opens a list of the software package procedure groups that are members of the given software package group, if any, recursively.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePackageGroupId (")

The UUID identifying the software package group whose member procedure groups should be listed.

SoftwarePackageProcedureGroupPropertiesRequired
softwarePackageProcedureGroupPropertiesRequired (")

The properties that will be required when getting a software package procedure group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package procedure groups that can be passed to GetSoftwarePackageProcedureGroupsListRecursive

xsd:long totalNoOfSoftwarePackageProcedureGroups

The total number of software package procedure groups in the list whose handle has been returned.

Remarks

If the softwarePackageGroupId is not supplied, the method will list the software package procedure groups that are members of the superfolder (for example, library).

More information:

[SoftwarePackageProcedureGroupPropertiesRequired](#) (see page 839)

OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList

The OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList method opens a list of the software package procedures that are members of the given software package procedure group.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageProcedureGroupId (")

The UUID of the software package procedure group whose member procedures should be listed.

SoftwarePackageProcedurePropertiesRequired
softwarePackageProcedurePropertiesRequired (")

The properties that will be required when getting a procedure from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of procedures that can be passed to `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures`

xsd:long totalNoOfSoftwarePackageProcedures

The total number of procedures in the list whose handle has been returned.

Remarks

The method returns the total number of software package procedures in the list, and a handle to the list that can be used for further calls to `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures`.

More information:

[SoftwarePackageProcedurePropertiesRequired](#) (see page 833)

More information:

[SoftwarePackageProcedurePropertiesRequired](#) (see page 833)

[SoftwarePackageProcedureTypeMask](#) (see page 834)

OpenSoftwarePackageProcedureList

The `OpenSoftwarePackageProcedureList` opens a list of procedures for a given software package.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePackageld (")

The UUID of the software package whose volumes should be listed.

SoftwarePackageProcedureTypeMask softwarePackageProcedureTypes (m)

The structure can be used to set which types of procedure should be returned in the list.

SoftwarePackageProcedurePropertiesRequired softwarePackageProcedurePropertiesRequired (m)

The properties that will be required when getting a procedure from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package procedures that can be passed to GetSoftwarePackageProcedures

xsd:long totalNoOfSoftwarePackageProcedures

The total number of procedures in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetSoftwarePackageProcedures method.

RepositionSoftwarePackageProcedure

Reposition the order of procedure in the given procedure group

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwarePackageProcedureGroupId (m)

Specifies theUUID of the software procedure group in which the given procedure is to be repositioned.

xsd:string softwarePackageProcedureId (m)

Takes the UUID of procedure to be repositioned.

xsd:long repositionNumber

Identifies the number to which the procedure is to be repositioned.

Return Values

This is a void method that does not expose a return value.

Remarks

None

RepositionSoftwarePackageProcedureByName

Repositions the order of procedures in the given procedure group.

Parameters**xsd:string sessionId (m)**

Specifies the session id obtained from the login to the web service.

xsd:string softwarePackageProcedureGroupName (m)

Identifies the procedure group name in which the procedure is to be repositioned.

SoftwarePackageProcedure softwarePackageProcedure (m)

Identifies the procedure and the package in which the procedure exists.

xsd:long repositionNumber

Indicates the position number to which the procedure is to be repositioned.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SoftwarePackageProcedure](#) (see page 865)

SetSoftwarePackageProcedure

The SetSoftwarePackageProcedure method updates the properties of a software package procedure.

Parameters

xsd:string sessionId (^m)

The session identifier.

SetSoftwarePackageProcedureProperties

setSoftwarePackageProcedureProperties (^m)

The properties that should be updated for the specified software package procedure.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePackageProcedureProperties](#) (see page 752)

SetSoftwarePackageProcedure2

The SetSoftwarePackageProcedure2 method updates the properties of a software package procedure.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence SetSoftwarePackageProcedureProperties2, which includes another sequence SoftwarePackageProcedureBootMask2. The SoftwarePackageProcedureBootMask2 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also lets you use the software package procedure comment as a default user message for the jobs created using this procedure.

Parameters

xsd:string sessionId (^m)

Specifies the session identifier obtained from the login method of the web service.

**SetSoftwarePackageProcedureProperties2
setSoftwarePackageProcedureProperties (^m)**

Specifies the properties that must be updated for the specified software package procedure.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePackageProcedureProperties2](#) (see page 754)

SetSoftwarePackageProcedure3

The SetSoftwarePackageProcedure3 method updates the properties of a software package procedure. This method is provided to support “excludeFromRAC” option while updating the procedure properties, the existing API’s doesn’t have the option.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

**dsm:SetSoftwarePackageProcedureProperties3
setSoftwarePackageProcedureProperties**

Specifies the properties that must be updated for the specified software package procedure.

Return Values

None.

SetSoftwarePackageProcedureGroup

The SetSoftwarePackageProcedureGroup method updates the properties of a software package procedure group.

Parameters

xsd:string sessionId (")

The session identifier.

**SetSoftwarePackageProcedureGroupProperties
softwarePackageProcedureGroup (")**

The properties that should be updated for the specified software package procedure group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePackageProcedureGroupProperties](#) (see page 735)

OpenSoftwarePackageProcedureGroupListRecursive

The `OpenSoftwarePackageProcedureGroupListRecursive` method opens a list of the software package procedure groups that are members of the given software package group, if any, recursively.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageGroupId (^m)

The UUID identifying the software package group whose member procedure groups should be listed.

**SoftwarePackageProcedureGroupPropertiesRequired
softwarePackageProcedureGroupPropertiesRequired (^m)**

The properties that will be required when getting a software package procedure group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package procedure groups that can be passed to `GetSoftwarePackageProcedureGroupsListRecursive`

xsd:long totalNoOfSoftwarePackageProcedureGroups

The total number of software package procedure groups in the list whose handle has been returned.

Remarks

If the `softwarePackageGroupId` is not supplied, the method will list the software package procedure groups that are members of the superfolder (for example, library).

More information:

[SoftwarePackageProcedureGroupPropertiesRequired](#) (see page 839)

UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroup

The `UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroup` method unlinks one or more software package procedures from a given software package procedure group.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageProcedureGroupId (^m)

The UUID identifying the software package procedure group from which the software package procedures should be unlinked.

ArrayOfstring softwarePackageProcedureIds (^m)

An array of UUIDs identifying the software package procedures that should be unlinked from the software package procedure group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

GetSoftwarePackageProcedure3

The GetSoftwarePackageProcedure3 method retrieves a software package procedure that its UUID identifies. This method is provided to return “excludeFromRAC” option in the procedure properties, the existing API’s doesn’t have the option.

Parameters

xsd:string sessionId

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwarePackageProcedureId

UUID of the software procedure whose properties should be retrieved.

**dsm:SoftwarePackageProcedurePropertiesRequired
softwarePackageProcedurePropertiesRequired**

Specifies the properties to be retrieved

Return Values

**dsm:SoftwarePackageProcedureProperties3
GetSoftwarePackageProcedureReturn**

Procedure properties.

GetSoftwarePackageProcedure4

The GetSoftwarePackageProcedure4 method retrieves the software package procedure properties.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string softwarePackageProcedureId

Specifies the UUID of the software package procedure id, the properties of which you want to retrieve.

dsm: SoftwarePackageProcedurePropertiesRequired2

Specifies which procedure properties that are required when retrieving a software package procedure property.

**dsm: SoftwarePackageProcedurePropertiesRequired2
softwarePackageProcedure
PropertiesRequired**

Specifies the procedure properties that are required when retrieving a software package procedure property.

Return Values

**dsm: SoftwarePackageProcedureProperties4
GetSoftwarePackageProcedureReturn**

Returns the properties of a SoftwarePackage procedure.

More information:

[SoftwarePackageProcedureProperties4](#) (see page 907)

[ArrayOfSoftwarePackageProcedureProperties4](#) (see page 985)

[SoftwarePackageProcedurePropertiesRequired2](#) (see page 910)

GetSoftwarePackageProcedures3

The GetSoftwarePackageProcedures3 method retrieves the procedures from a software package procedure list. This method is provided to return “excludeFromRAC” option in the procedure properties, the existing API’s doesn’t have the option.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

xsd:long listHandle

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureList.

xsd:long noOfSoftwarePackageProcedures

Specifies the total number of procedures to get from the list. The number must be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageProcedureList.

Return Values

**dsm:ArrayOfSoftwarePackageProcedureProperties3
getSoftwarePackageProcedures**

Array of software package procedure properties.

GetSoftwarePackageProcedureList

The GetSoftwarePackageProcedureList method retrieves the list of software package procedures in software package.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string softwarePackageId

Specifies the UUID of the software package from which the procedures list is retrieved.

**dsm:SoftwarePackageProcedureTypeMask
softwarePackageProcedureTypes**

Specifies the types of procedure returned in the list.

**dsm:SoftwarePackageProcedurePropertiesRequired2
softwarePackageProcedurePropertiesRequired**

Specifies the properties that are required when retrieving a procedure from the list.

Xsd:long index

Specifies the starting index for the list.

xsd:long numRequired

Specifies the number of procedures to get from the list.

Return Values

xsd:long totalNumProcs

Returns the total number of procedures that are available.

**dsm:ArrayOfSoftwarePackageProcedureProperties4
getSoftwarePackageProcedure**

Specifies the array of software package procedure properties, one for each procedure.

getSoftwarePackageProcedures

Returns the array of SoftwarePackage procedure properties.

More information:

[ArrayOfSoftwarePackageProcedureProperties4](#) (see page 985)
[SoftwarePackageProcedurePropertiesRequired2](#) (see page 910)

Software Package Procedure Groups

This section describes the Software Package Procedure Group methods.

This section contains the following topics:

[CloseParentSoftwarePackageProcedureGroupList](#) (see page 352)

[CloseSoftwarePackageProcedureGroupList](#) (see page 352)

[CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList](#) (see page 353)

[CreateSoftwarePackageProcedureGroup](#) (see page 354)

[DeleteSoftwarePackageProcedureGroups](#) (see page 354)

[EmptySoftwarePackageProcedureGroups](#) (see page 355)

[FindSoftwarePackageProcedureGroup](#) (see page 355)

[GetParentSoftwarePackageProcedureGroups](#) (see page 356)

[GetSoftwarePackageProcedureGroup](#) (see page 357)

[GetSoftwarePackageProcedureGroups](#) (see page 357)

[GetSoftwarePackageProcedureGroupSoftwarePackageProcedures](#) (see page 358)

[GetSoftwarePackageProcedureGroupSoftwarePackageProcedures3](#) (see page 360)

[GetSoftwarePackageProcedureGroupsRecursive](#) (see page 360)

[LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroup](#) (see page 361)

[OpenParentSoftwarePackageProcedureGroupList](#) (see page 362)

[OpenSoftwarePackageProcedureGroupList](#) (see page 363)

[OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList](#) (see page 364)

[SetSoftwarePackageProcedureGroup](#) (see page 365)

[UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroup](#) (see page 365)

[UnlinkSoftwarePackagesFromSoftwarePackageGroup](#) (see page 366)

[GetSoftwarePackageProcedureGroupProcedureList](#) (see page 367)

CloseParentSoftwarePackageProcedureGroupList

The CloseParentSoftwarePackageProcedureGroupList method closes a software package procedure group list that was opened by OpenParentSoftwarePackageProcedureGroupList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenParentSoftwarePackageProcedureGroupList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePackageProcedureGroupList

The CloseSoftwarePackageProcedureGroupList method closes a software package procedure group list that was opened by OpenSoftwarePackageProcedureGroupList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageProcedureGroupList.

Return Values

This is a void method that does not expose a return value.

Return Values

None

CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList

The CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList method closes a software package procedure list that was opened by OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[CreateSoftwarePackageProcedureProperties](#) (see page 644)

CreateSoftwarePackageProcedureGroup

The CreateSoftwarePackageProcedureGroup method creates a software package procedure group.

Parameters

xsd:string sessionId (")

The session identifier.

**CreateSoftwarePackageProcedureGroupProperties
softwarePackageProcedureGroup (")**

The properties required to create a software package procedure group.

Return Values

xsd:string CreateSoftwarePackageProcedureGroupReturn

If a software package group was successfully created, the UUID of the new group is returned.

Remarks

None

More information:

[CreateSoftwarePackageProcedureGroupProperties](#) (see page 644)

DeleteSoftwarePackageProcedureGroups

The DeleteSoftwarePackageProcedureGroups method deletes software package procedure groups

Parameters

xsd:string sessionId (")

The session identifier.

ArrayOfstring softwarePackageProcedureGroupIds (")

An array of software package UUIDs identifying the software package procedure groups to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

EmptySoftwarePackageProcedureGroups

The EmptySoftwarePackageProcedureGroups method empties one or more software package procedure group(s) identified by their UUIDs. The members will be removed from the given group.

Parameters

xsd:string sessionId ("")

The session identifier.

ArrayOfstring softwarePackageProcedureGroupIds ("")

An array of UUIDs identifying the software package procedure groups to be emptied.

Return Values

This is a void method that does not expose a return value.

Remarks

None

FindSoftwarePackageProcedureGroup

The FindSoftwarePackageProcedureGroup method finds a software package procedure group given its name and the UUID and type of the parent software package group.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string parentSoftwarePackageProcedureGroupId

The UUID identifying the parent group of the software package procedure group to search for.

SoftwarePackageProcedureGroupScope

parentSoftwarePackageProcedureGroupScope ("")

The type of the parent group of the software package procedure group to search for.

xsd:string softwarePackageProcedureGroupName ("")

The name of the software package procedure group to search for.

Return Values

xsd:string FindSoftwarePackageProcedureGroupReturn

If a software package procedure group is found its UUID is returned.

Remarks

The group name must be specified exactly. This method does not allow wildcards. The group name is case insensitive.

More information:

[SoftwarePackageProcedureGroupScope](#) (see page 96)

GetParentSoftwarePackageProcedureGroups

The GetParentSoftwarePackageProcedureGroups method retrieves the parent software package procedure groups from a software package procedure group's parent group list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenParentSoftwarePackageProcedureGroupList.

xsd:long noOfParentSoftwarePackageGroups (")

The total number of software package procedure groups to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenParentSoftwarePackageProcedureGroupList.

Return Values

ArrayOfSoftwarePackageGroupProperties

GetParentSoftwarePackageProcedureGroupsReturn

An array of SoftwarePackageProcedureGroupProperties - one for each parent group requested from the list.

Remarks

You can make multiple calls to GetParentSoftwarePackageProcedureGroups to iterate down the list.

More information:

[SoftwarePackageGroupProperties](#) (see page 828)

GetSoftwarePackageProcedureGroup

The GetSoftwarePackageProcedureGroup method retrieves a software package procedure group identified by its UUID.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageProcedureGroupId ("")

The UUID of the software package procedure group to retrieve.

**SoftwarePackageProcedureGroupPropertiesRequired
softwarePackageProcedureGroupPropertiesRequired ("")**

This parameter specifies which software package procedure group properties to retrieve.

Return Values

**SoftwarePackageProcedureGroupProperties
GetSoftwarePackageProcedureGroupReturn**

The software package procedure group properties retrieved.

Remarks

None

More information:

[SoftwarePackageProcedureGroupPropertiesRequired](#) (see page 839)

GetSoftwarePackageProcedureGroups

The GetSoftwarePackageProcedureGroups method retrieves the procedure groups from a software package procedure group list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureGroupList.

xsd:long noOfSoftwarePackageProcedureGroups (^m)

The total number of software package procedure groups to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwarePackageProcedureGroupList`.

Return Values

**ArrayOfSoftwarePackageProcedureGroupProperties
GetSoftwarePackageProcedureGroupsReturn**

An array of `SoftwarePackageProcedureGroupProperties` - one for each group requested from the list.

Remarks

You can make multiple calls to `GetSoftwarePackageProcedureGroups` to iterate down the list.

More information:

[SoftwarePackageProcedureGroupProperties](#) (see page 838)

GetSoftwarePackageProcedureGroupSoftwarePackageProcedures

The `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures` method retrieves the software package procedures from a software package procedure group's procedure list.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list as returned by a previous call to `OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList`.

xsd:long noOfSoftwarePackageProcedures (^m)

The total number of procedures to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwarePackageProcedureGroupSoftwarePackageProceduresList`.

Return Values**ArrayOfSoftwarePackageProcedureProperties****GetSoftwarePackageProcedureGroupSoftwarePackageProceduresReturn**

An array of `SoftwarePackageProcedureProperties` - one for each procedure requested from the list.

Remarks

You can make multiple calls to `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures` to iterate down the list.

More information:

[SoftwarePackageProcedureProperties](#) (see page 843)

GetSoftwarePackageProcedureGroupSoftwarePackageProcedures3

The GetSoftwarePackageProcedureGroupSoftwarePackageProcedures3 method retrieves the software package procedures from a software package procedure group's procedure list. This method is provided to return “excludeFromRAC” option in the procedure properties, the existing API's doesn't have the option.

Parameters

xsd:string sessionId

Specifies the session identifier obtained from the login method of the web service.

xsd:long listHandle

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList.

xsd:long noOfSoftwarePackageProcedures

Specifies the total number of procedures to get from the list. The number must be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList.

Return Values

dsm:ArrayOfSoftwarePackageProcedureProperties3

GetSoftwarePackageProcedureGroupSoftwarePackageProceduresReturn

Returns an array of SoftwarePackageProcedureProperties3—one for each procedure requested from the list.

GetSoftwarePackageProcedureGroupsRecursive

The GetSoftwarePackageProcedureGroupsRecursive method retrieves the procedure groups from a software package procedure group list.

Parameters

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageProcedureGroupListRecursive.

xsd:long noOfSoftwarePackageProcedureGroups (")

The total number of software package procedure groups in the list whose handle has been returned.

Return Values

ArrayOfSoftwarePackageProcedureGroupProperties GetSoftwarePackageProcedureGroupsReturn

An array of SoftwarePackageProcedureGroupProperties - one for each group requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageProcedureGroupsRecursive to iterate down the list.

More information:

[SoftwarePackageProcedureGroupProperties](#) (see page 838)

LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroup

The LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroup method links one or more software package procedure(s) to a given software package procedure group, making them members of the group.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageProcedureGroupId (")

The UUID identifying the software package procedure group to which the software package procedures should be linked.

ArrayOfstring softwarePackageProcedureIds (")

An array of UUIDs identifying the software package procedures that should be linked to the software package procedure group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

OpenParentSoftwarePackageProcedureGroupList

The OpenParentSoftwarePackageProcedureGroupList method opens a list of the software package procedure groups that are parent groups of the given software package procedure group.

The method returns the total number of software package procedure groups in the list, and a handle to the list that can be used for further calls to GetParentSoftwarePackageProcedureGroups

All open functions return a list handle and total count of items. The items can be retrieved by using the corresponding Get Methods

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePackageProcedureGroupId (^m)

The UUID of the software package procedure group whose parent groups should be listed.

SoftwarePackageGroupPropertiesRequired

softwarePackageProcedureGroupPropertiesRequired (^m)

The properties that will be required when getting a software package procedure group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package procedure groups that can be passed to GetParentSoftwarePackageProcedureGroups

xsd:long totalNoOfParentSoftwarePackageGroups

The total number of parent software package procedure groups in the list whose handle has been returned.

Remarks

None

More information:

[SoftwarePackageGroupPropertiesRequired](#) (see page 778)

OpenSoftwarePackageProcedureGroupList

This method opens a list of software package procedure groups. The method returns the total number of software package procedure groups in the list, and a handle to the list that can be used for further calls to `GetSoftwarePackageProcedureGroups`

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageGroupId (")

The UUID identifying the `softwarePackageProcedureGroupPropertiesRequired`.

SoftwarePackageProcedureGroupPropertiesRequired (")

The properties that will be required when getting a software package procedure group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package procedure groups that can be passed to `GetSoftwarePackageProcedureGroups`

xsd:long totalNoOfSoftwarePackageProcedureGroups

The total number of software package procedure groups in the list whose handle has been returned.

Remarks

None

More information:

[SoftwarePackageProcedureGroupPropertiesRequired](#) (see page 839)

OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList

The `OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList` method opens a list of the software package procedures that are members of the given software package procedure group.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageProcedureGroupId ("")

The UUID of the software package procedure group whose member procedures should be listed.

**SoftwarePackageProcedurePropertiesRequired
softwarePackageProcedurePropertiesRequired ("")**

The properties that will be required when getting a procedure from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of procedures that can be passed to `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures`

xsd:long totalNoOfSoftwarePackageProcedures

The total number of procedures in the list whose handle has been returned.

Remarks

The method returns the total number of software package procedures in the list, and a handle to the list that can be used for further calls to `GetSoftwarePackageProcedureGroupSoftwarePackageProcedures`.

More information:

[SoftwarePackageProcedurePropertiesRequired](#) (see page 833)

SetSoftwarePackageProcedureGroup

The SetSoftwarePackageProcedureGroup method updates the properties of a software package procedure group.

Parameters

xsd:string sessionId ("")

The session identifier.

**SetSoftwarePackageProcedureGroupProperties
softwarePackageProcedureGroup ("")**

The properties that should be updated for the specified software package procedure group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePackageProcedureGroupProperties](#) (see page 735)

UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroup

The UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroup method unlinks one or more software package procedures from a given software package procedure group.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageProcedureGroupId ("")

The UUID identifying the software package procedure group from which the software package procedures should be unlinked.

ArrayOfstring softwarePackageProcedureIds (^m)

An array of UUIDs identifying the software package procedures that should be unlinked from the software package procedure group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkSoftwarePackagesFromSoftwarePackageGroup

The UnlinkSoftwarePackagesFromSoftwarePackageGroup method unlinks one or more software packages from a given software package group.

Parameters

xsd:string sessionId

The session identifier.

xsd:string softwarePackageGroupId

The UUID identifying the software package group from which the software packages should be unlinked.

ArrayOfstring softwarePackageIds

An array of UUIDs identifying the software packages that should be unlinked from the software package group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

GetSoftwarePackageProcedureGroupProcedureList

The GetSoftwarePackageProcedureGroupProcedureList method retrieves the list of software package procedures in a procedure group.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string softwarePackageId

Species the UUID of the software package procedure group from which the procedures list is retrieved.

dsm:SoftwarePackageProcedurePropertiesRequired2 propsRequired

Specifies the properties that are required when retrieving a procedure from the list.

xsd:long index

Specifies the starting index for the list.

xsd:long numRequired

Specifies the number of procedures you get from the list.

Return Values

xsd:long totalNumProcs

Returns the available total number of procedures.

dsm:ArrayOfSoftwarePackageProcedureProperties4

Returns an array of SoftwarePackage procedure properties.

dsm:ArrayOfSoftwarePackageProcedureProperties4**getSoftwarePackageProcedures**

Returns an array of SoftwarePackage procedure properties.

More information:

[ArrayOfSoftwarePackageProcedureProperties4](#) (see page 985)

[SoftwarePackageProcedurePropertiesRequired2](#) (see page 910)

Software Package Volumes

This section describes software package volume methods.

This section contains the following topics:

[CloseSoftwarePackageVolumeElementList](#) (see page 368)
[CloseSoftwarePackageVolumeList](#) (see page 369)
[CopyElementsToSoftwarePackageVolume](#) (see page 369)
[CopyElementsToSoftwarePackageVolume2](#) (see page 370)
[CreateSoftwarePackageVolume](#) (see page 372)
[DeleteElementsFromSoftwarePackageVolume](#) (see page 372)
[DeleteSoftwarePackageVolumes](#) (see page 373)
[GetSoftwarePackageVolume](#) (see page 373)
[GetSoftwarePackageVolumeElements](#) (see page 374)
[GetSoftwarePackageVolumes](#) (see page 375)
[OpenSoftwarePackageVolumeElementList](#) (see page 376)
[OpenSoftwarePackageVolumeList](#) (see page 377)
[SetSoftwarePackageVolume](#) (see page 377)

CloseSoftwarePackageVolumeElementList

The CloseSoftwarePackageVolumeElementList method closes a volume's element list that was opened by OpenSoftwarePackageVolumeList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageVolumeElementList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePackageVolumeList

The CloseSoftwarePackageVolumeList method closes a software package volume list that was opened by OpenSoftwarePackageVolumeList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageVolumeList

Return Values

This is a void method that does not expose a return value.

Remarks

None

CopyElementsToSoftwarePackageVolume

The CopyElementsToSoftwarePackageVolume method copies the software elements pointed to by a series of directory paths, into the given software package volume.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageVolumeId ("")

The UUID identifying the software package volume to which the elements should be copied.

xsd:string softwarePackageVolumeSubPath ("")

The path in the volume to where the elements will be copied to.

ArrayOfFileSystemElementPath fileSystemElementPath ("")

An array of paths to the file system elements that should be copied to the given software package volume.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CopyElementsToSoftwarePackageVolume2

The CopyElementsToSoftwarePackageVolume2 method copies the software elements pointed to by a series of directory paths into the given software package volume. This method has an extra parameter, sourceOnManager, that enables user to initiate copying of software elements from the manager or Web Services.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePackageVolumeId (m)

The UUID identifying the software package volume to which the elements should be copied.

xsd:string softwarePackageVolumeSubPath (m)

The path in the volume to where the elements will be copied to.

xsd:Boolean sourceOnManager

If sourceOnManager is true, the manager initiates the copying of software elements. If sourceOnManager is false, the CA ITCM Webservice initiates the copying of software elements.

If the manager initiates copying, the value of fileSystemElementPath is from the context of the manager machine. If the webservice initiates copying, the value of fileSystemElementPath is from the context of the webservice (and the machine the web service is running on).

ArrayOfFileSystemElementPath fileSystemElementPath (m)

An array of paths to the file system elements that should be copied to the given software package volume.

Return Values

This is a void method that does not expose a return value.

Remarks

If `SourceOnManager` is true, then the copying of the elements is performed from the context of the user used to run CAF (caf user) at the manager.

If `sourceOnManager` is false, then the copying of the elements is performed from the context of the web service. The user associated with the thread (web service user) performing the copying at the web service must have permission to do the copy.

The web service user or caf user must have the following permissions:

- Read permission on the source elements to be copied (the files referred to by `fileSystemElementPath[...]`).
- Write permission on `%INSTALL DIR%\SD\ASM\LIBRARY`

If CA ITCM Webservice is installed on Windows 2000 Advance Server and `fileSystemElementPath` contains UNC paths to access remote files, then `sourceOnManager` must be set to False and the user must have the above specified permission to access the remote files.

Definition of the Web Service User

Web Service on Windows

If the web service is running on Windows and the security provider used for authentication (in the web service login method) is WINNT, the web service will attempt to impersonate the user used in logon to perform the copying rather than use the user set by IIS (for the thread).

Note: If the web service is running on Windows Advanced Server 2000, then to impersonate the supplied credentials (in login) the user associated with the thread must have the following local security privileges:

- Act as a part of the operating System
- Bypass traverse checking
- Log on locally

The user associated with the current thread (before impersonation takes place) is whatever has been configured in IIS, by default the IIS anonymous user.

If the web service is running on Windows but the WINNT security provider is not used (for example LDAP could be used), or an impersonation attempt fails, then the web service user will be the user associated with the current thread as configured by IIS.

Web Service on Linux

If the web service is running on Linux, the user will be the user associated with the process, by default root.

CreateSoftwarePackageVolume

The CreateSoftwarePackageVolume method creates a new software package volume.

Parameters

xsd:string sessionId ("")

The session identifier.

**CreateSoftwarePackageVolumeProperties
createSoftwarePackageVolumeProperties ("")**

The properties required to create a software package volume.

Return Values

xsd:string CreateSoftwarePackageVolumeReturn

If a software package volume was successfully created, the UUID of the new volume is returned.

Remarks

None

More information:

[CreateSoftwarePackageVolumeProperties](#) (see page 690)

DeleteElementsFromSoftwarePackageVolume

The DeleteElementsFromSoftwarePackageVolume method deletes one or more elements from a software package volume.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageVolumeId ("")

The UUID identifying the software package volume from which the elements should be deleted.

ArrayOfstring softwarePackageVolumeSubPath (")

An array of paths to the elements within the software package volume that should be deleted.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteSoftwarePackageVolumes

The DeleteSoftwarePackageVolumes method deletes one or more software package volumes identified by their UUIDs.

Parameters**xsd:string sessionId (")**

The session identifier.

ArrayOfstring softwarePackageVolumeIds (")

An array of UUIDs that identify the software package volumes that should be deleted.

Return Values

This is a void method that does not expose a return value.

Remarks

None

GetSoftwarePackageVolume

The GetSoftwarePackageVolume method retrieves a software package volume identified by its UUID.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePackageVolumeId (")

The UUID of the software package volume to retrieve.

SoftwarePackageVolumePropertiesRequired
softwarePackageVolumePropertiesRequired (")

This parameter specifies which software package volume properties to retrieve.

Return Values

SoftwarePackageVolumeProperties GetSoftwarePackageVolumeReturn

The software package volume properties retrieved.

Remarks

None

More information:

[SoftwarePackageVolumePropertiesRequired](#) (see page 837)

GetSoftwarePackageVolumeElements

The GetSoftwarePackageVolumeElements method retrieves the elements from a volume's element list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageVolumeElementList.

xsd:long totalNoOfElements (")

The total number of elements to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageVolumeElementList.

Return Values

GetSoftwarePackageVolumeElementsReturn

An array of SoftwarePackageVolumeElementProperties - one for each element requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageVolumeElements to iterate down the list.

More information:

[SoftwarePackageVolumeElementProperties](#) (see page 850)

GetSoftwarePackageVolumes

The GetSoftwarePackageVolumes method retrieves the volumes from a software package volume list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageVolumeList.

xsd:long noOfSoftwarePackageVolumes ("")

The total number of volumes to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePackageVolumeList.

Return Values

ArrayOfSoftwarePackageVolumeProperties

GetSoftwarePackageVolumesReturn

An array of SoftwarePackageVolumeProperties - one for each volume requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackageVolumes to iterate down the list.

More information:

[SoftwarePackageVolumeProperties](#) (see page 836)

OpenSoftwarePackageVolumeElementList

The OpenSoftwarePackageVolumeElementList opens a list of elements for a given software package volume.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePackageVolumeld ("")

The UUID identifying the software package volume whose elements should be listed.

ElementTypeMask elementTypeMask ("")

Indicates the types of element (files and directories) to be returned in the list.

xsd:string folderPath

The subpath of the folder in the volume whose elements should be listed. If folderPath is empty or is a slash (/), the list contains only the elements in the root volume folder.

SoftwarePackageVolumeElementPropertiesRequired

softwarePackageVolumeElementPropertiesRequired ("")

The properties that are required when getting a software policy from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of volume elements that can be passed to GetSoftwarePackageVolumeElements

xsd:long totalNoOfSoftwarePackageVolumeElements

The total number of elements in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetSoftwarePackageVolumeElements method.

More information:

[SoftwarePackageVolumeElementPropertiesRequired](#) (see page 836)

OpenSoftwarePackageVolumeList

The OpenSoftwarePackageVolumeList opens a list of volumes for a given software package.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageId (")

The UUID of the software package whose volumes should be listed.

SoftwarePackageVolumePropertiesRequired

softwarePackageVolumePropertiesRequired (")

The properties that will be required when getting a volume from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package volumes that can be passed to GetSoftwarePackageVolumes

xsd:long totalNoOfSoftwarePackageVolumes

The total number of volumes in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetSoftwarePackageVolumes method.

More information:

[SoftwarePackageVolumePropertiesRequired](#) (see page 837)

SetSoftwarePackageVolume

The SetSoftwarePackageVolume method updates the properties of a software package volume.

Parameters

xsd:string sessionId (")

The session identifier.

SetSoftwarePackageVolumeProperties

setSoftwarePackageVolumeProperties (")

The properties that should be updated for the specified software package volume.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePackageVolumeProperties](#) (see page 757)

Software Packages

The following are Software Package methods.

CloseSoftwarePackageInstallationList

The CloseSoftwarePackageInstallationList method closes a software package installation list that was opened by OpenSoftwarePackageProcedureInstallationList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePackageInstallationList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePackageList

The CloseSoftwarePackageList method closes a software package list that was opened by OpenSoftwarePackageList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle (^m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to `OpenSoftwarePackageList`.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateSoftwarePackage

The `CreateSoftwarePackage` method creates a software package

Parameters**xsd:string sessionId (^m)**

The session identifier.

CreateSoftwarePackageProperties createSoftwarePackageProperties (^m)

The properties required to create a software package.

Return Values**xsd:string CreateSoftwarePackageReturn**

If a software package was successfully created, the UUID of the new package is returned.

Remarks

None

More information:

[CreateSoftwarePackageProperties](#) (see page 690)

DeleteSoftwarePackages

The `DeleteSoftwarePackages` method deletes software packages

Parameters**xsd:string sessionId (^m)**

The session identifier.

ArrayOfstring softwarePackageIds (^m)

An array of software package UUIDs identifying the software packages to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteSoftwarePackagesByName

The DeleteSoftwarePackagesByName method deletes software packages.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfSoftwarePackage softwarePackages (m)

An array list of sequence SoftwarePackage. Each member of type SoftwarePackage contains details specifying a particular software package.

Return Values

None

Remarks

None

ExportSoftwarePackage

The ExportSoftwarePackage exports the software package to a location on a local computer or a network path.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePackageId (m)

The UUID of the software package to be exported.

dsm:ExportSoftwarePkgProperties softwareProperties (m)

This parameter specifies which software package properties are required to export.

Return Values

None

Remarks

None

ExportSoftwarePackageByName

The ExportSoftwarePackageByName exports the software package identified by name and version to a location on a local computer or a network path.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePkgName (m)

The name of the software package to be exported.

xsd:string softwarePkgVersion (m)

The version of the software package to be exported.

dsm:ExportSoftwarePkgProperties softwareProperties (m)

This parameter specifies which software package properties are required to export.

Return Values

None

Remarks

None

FindSoftwarePackage

The FindSoftwarePackage method finds a software package given its name and version

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePackageName (m)

The software package name to search for.

xsd:string softwarePackageVersion (")

The software package version to search for.

Return Values**xsd:string FindSoftwarePackageReturn**

If a software package is found its UUID is returned.

Remarks

The name and version must be specified exactly. This method does not allow wildcards. The software package name is case insensitive.

GetSoftwarePackage

The GetSoftwarePackage method retrieves a software package identified by its UUID.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePackageId (")

The UUID of the software package to retrieve.

**SoftwarePackagePropertiesRequired
softwarePackagePropertiesRequired (")**

This parameter specifies which software package properties to retrieve.

Return Values**SoftwarePackageProperties GetSoftwarePackageReturn**

The software package properties retrieved.

Remarks

None

More information:

[SoftwarePackageProperties](#) (see page 847)

[SoftwarePackagePropertiesRequired](#) (see page 835)

GetSoftwarePackage2

The GetSoftwarePackage2 method retrieves a software package identified by its UUID. This method also retrieves software package format (regular or virtual).

Parameters

xsd:string sessionId (")

The session identifier

xsd:string softwarePackageId (")

The UUID of the software package to retrieve

**SoftwarePackagePropertiesRequired
softwarePackagePropertiesRequired (")**

This parameter specifies which software package properties to retrieve.

Return Values

SoftwarePackageProperties2 GetSoftwarePackagesReturn

The software package properties retrieved.

Remarks

None

More information:

[SoftwarePackagePropertiesRequired](#) (see page 835)

[SoftwarePackageProperties2](#) (see page 849)

GetSoftwarePackageInstallations

The GetSoftwarePackageInstallations method retrieves the installations from a software package installation list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePackageInstallationList.

xsd:long noOfSoftwarePackageInstallations (")

The total number of installations to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwarePackageInstallationList`.

Return Values**ArrayOfUnitSoftwareJobProperties
GetSoftwarePackageInstallationsReturn**

An array of `UnitSoftwareJobProperties` - one for each software package installation requested from the list.

Remarks

You can make multiple calls to `GetSoftwarePackageInstallations` to iterate down the list.

More information:

[UnitSoftwareJobProperties](#) (see page 880)

GetSoftwarePackageInstallations2

The `GetSoftwarePackageInstallations2` method retrieves the installations from a software package installation list. This method also retrieves software package format (regular or virtual).

Parameters**xsd:string sessionId (")**

The session identifier

xsd:long listHandle (")

A numeric handle to a software jobs list, as returned by a previous call to `OpenSoftwarePackageInstallationList`.

xsd:long noOfSoftwarePackageInstallations (")

The total number of software jobs to get from the list. This number should be less than or equal to the total number in the list, as returned by a previous call to `OpenSoftwarePackageInstallationList`.

Return Values**ArrayOfUnitSoftwareJobProperties2
GetSoftwarePackageInstallationsReturn**

An array of `UnitSoftwareJobProperties` - one for each software package installation requested from the list

Remarks

You can make multiple calls to `GetSoftwarePackageInstallations2` to iterate down the list.

More information:

[UnitSoftwareJobProperties2](#) (see page 883)

GetSoftwarePackages

The `GetSoftwarePackages` method retrieves the software packages from a software package list.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to `OpenSoftwarePackageList`.

xsd:long noOfSoftwarePackages (")

The total number of software packages to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to `OpenSoftwarePackageList`.

Return Values**ArrayOfSoftwarePackageProperties GetSoftwarePackagesReturn**

An array of `SoftwarePackageProperties` - one for each software package requested from the list.

Remarks

You can make multiple calls to `GetSoftwarePackages` to iterate down the list.

More information:

[SoftwarePackageProperties](#) (see page 847)

GetSoftwarePackages2

The GetSoftwarePackages2 method retrieves the software packages from a software package list. This method also retrieves software package format (regular or virtual).

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a software packages list as returned by a previous call to OpenSoftwarePackageList.

xsd:long noOfSoftwarePackages (")

The total number of software packages to get from the list. This should be less than or equal to the total number in the list, as returned by a previous call to OpenSoftwarePackageList.

Return Values

ArrayOfSoftwarePackageProperties2 GetSoftwarePackagesReturn

An array of SoftwarePackageProperties - one for each software package requested from the list.

Remarks

You can make multiple calls to GetSoftwarePackages2 to iterate down the list.

More information:

[SoftwarePackageProperties2](#) (see page 849)

ImportSoftwarePackage

The ImportSoftwarePackage method imports software packages

Parameters

xsd:string sessionId (^m)

The session identifier.

ImportSoftwarePackageProperties importSoftwarePackageProperties (^m)

This parameter specifies the software package properties of the software package to import.

Return Values

xsd:string ImportSoftwarePackageReturn

Returns the softwarePackageID after importing the software package.

Remarks

None

More information:

[ImportSoftwarePackageProperties](#) (see page 711)

ImportSoftwarePackage2

Imports the software package into the software library

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

**dsm:ImportSoftwarePackageProperties2
importSoftwarePackageProperties2 (m)**

The properties required to import software package. Setting all the structure members is not mandatory.

Return Values

xsd:string softwarePackageId

Returns the softwarePackageID after importing the software package.

Remarks

None

More information:

[ImportSoftwarePackageProperties2](#) (see page 712)

ImportSoftwarePackageByName

The ImportSoftwarePackageByName method imports the software package into the software library identified by the name and version supplied in the sequence ImportSoftwarePackagePropertiesByName.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

**dsm:ImportSoftwarePackagePropertiesByName
importSoftwarePackagePropertiesByName (m)**

Specifies the properties required to import a software package. Setting all the structure members is not mandatory.

Return Values

xsd:string softwarePackageId

Returns the softwarePackageID after importing the software package.

Remarks

None

More information:

[ImportSoftwarePackagePropertiesByName](#) (see page 713)

OpenSoftwarePackageInstallationList

The OpenSoftwarePackageInstallationList opens a list of installations that have been made for a given software package.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePackageId (m)

The UUID identifying the software package whose installations should be listed.

xsd:string unitGroupId

The UUID identifying a group of units by which to filter the list of installations. You can use this parameter to return a list of software package installations to only those units within a specific group.

UnitSoftwareJobPropertiesRequired

unitSoftwareJobPropertiesRequired (")

The properties that will be required when getting an individual installation from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package installations that can be passed to GetSoftwarePackageInstallations

xsd:long totalNoOfSoftwarePackages

The total number of installations in the list whose handle has been returned.

Remarks

The method returns the size of the list, and a handle to the list that can be used with the GetSoftwarePackageInstallations method.

More information:

[UnitSoftwareJobPropertiesRequired](#) (see page 886)

OpenSoftwarePackageList

The OpenSoftwarePackageList opens a list of software packages.

Parameters

xsd:string sessionId (")

The session identifier.

SoftwarePackageMaskType softwarePackageMaskType (")

The structure can be used to set which types of software package should be returned in the list.

SoftwarePackagePropertiesRequired
softwarePackagePropertiesRequired (")

The properties that will be required when getting a software package from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of software packages that can be passed to GetSoftwarePackages

xsd:long totalNoOfSoftwarePackages

The total number of software packages in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetSoftwarePackages method.

More information:

[SoftwarePackageMaskType](#) (see page 779)

SealSoftwarePackages

The SealSoftwarePackages method seals one or more software packages. After sealing you cannot modify the software packages.

Parameters**xsd:string sessionId (")**

The session identifier.

ArrayOfstring softwarePackageIds (")

An array of software package UUIDs identifying the software packages to seal.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetSoftwarePackage

The SetSoftwarePackage method updates the properties of a software package

Parameters

xsd:string sessionId (^m)

The session identifier.

SetSoftwarePackageProperties softwarePackageProperties (^m)

The properties that should be updated for the specified software package.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePackageProperties](#) (see page 757)

UnsealSoftwarePackages

The UnsealSoftwarePackages method unseals one or more software packages.

Parameters

xsd:string sessionId (^m)

The session identifier.

ArrayOfstring softwarePackageIds (^m)

An array of software package UUIDs identifying the software packages to unseal.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateSoftwarePackage3

The CreateSoftwarePackage3 method is used to create a package in the SD library and supports the creation of wrapper packages.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:CreateSoftwarePackageProperties3 pkgProps

Includes the software package properties and the properties for wrapper packages.

xsd:Boolean sealAfterCreate

Specifies whether the package must be sealed after creation.

Return Values

xsd:string packageId

Returns the UUID of the created package.

More information:

[CreateSoftwarePackageProperties3](#) (see page 900)

SetSoftwarePackage3

The SetSoftwarePackage3 method is used to edit both the software package and also Debian wrapper package.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:SetSoftwarePackageProperties3 pkgProps

Specifies the properties of the package that you want to edit.

xsd:Boolean sealAfterEdit

Specifies whether the package must be sealed after editing.

Note: If the package is in a sealed state, you can only edit the comments and enableChecksum fields.

This parameter is ignored when the package is in a sealed state. The edit operation fails if the package is sealed and you try to set all the package properties.

More information:

[SetSoftwarePackageProperties3](#) (see page 901)

GetSoftwarePackage3

The GetSoftwarePackage3 method retrieves the software package properties from the Debian wrapper package.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string packageId

Specifies the UUID of the Debian wrapper package, the properties of which are retrieved.

dsm:SoftwarePackagePropertiesRequired3 propertiesRequired

Specifies the software package properties to retrieve.

Return Values

dsm:SoftwarePackageProperties3

Returns the properties of the software package.

More information:

[SoftwarePackageProperties3](#) (see page 902)

[SoftwarePackagePropertiesRequired3](#) (see page 906)

GetSoftwarePackageList3

The GetSoftwarePackageList3 method retrieves the list of software packages and their properties from a specified software package group.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string softwarePackageGroupId

Specifies the UUID of the software package group from which the packages list is retrieved.

Note: Pass NULL to retrieve all softwares.

**dsm:SoftwarePackagePropertiesRequired3
softwarePackagePropertiesRequired (m)**

Specifies the properties that are required when getting a software package from the list.

dsm:ArrayOfSoftwarePackageFilter3 filterArray

Specifies the filter on software package properties

dsm:SoftwarePackageProperty sortProperty

Specifies the property based on which you want to sort the packages.

xsd:boolean ascending

Specifies that the sorting takes place in ascending order.

Xsd:long index

Specifies the starting index for the list.

xsd:long numRequired

Specifies the total number of packages to get from the list.

Xsd:boolean refreshList

Specifies that the result set is recomputed for each call.

Return Values

xsd:long totalNumPackages

Returns the total number of packages available for the specified filter condition.

dsm:ArrayOfSoftwarePackageProperties3 swPkgArray

Returns the array of SoftwarePackages properties.

More information:

[ArrayOfSoftwarePackageFilter3](#) (see page 986)

[SoftwarePackagePropertiesRequired3](#) (see page 906)

[ArrayOfSoftwarePackageProperties3](#) (see page 986)

GetSoftwarePackageInstallationList

The GetSoftwarePackageInstallationList method retrieves the list of installations for a given software package. This method is supported only on the domain manager.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string softwarePackageId

Specifies the UUID of the software package for which you want the installations list.

xsd:string unitGroupId

Specifies the UUID the group for which you want the installations list. You can also use this parameter to return a list of software package installations to units in a specific group.

**dsm:UnitSoftwareJobPropertiesRequired
unitSoftwareJobPropertiesRequired**

Specifies the properties that want to get from each installation in the list.

Xsd:long index

Specifies the starting index for the list.

xsd:long numInstallationsRequired

Specifies the number of installations to get from the list.

Return Values

Xsd:int totalNumInstallations

Returns the total number of installations in the list.

**dsm:ArrayOfUnitSoftwareJobProperties3
arrOfUnitSoftwareJobProperties**

Returns an array of UnitSoftwareJobProperties, one for each software package installation that are requested from the list.

More information:

[ArrayOfUnitSoftwareJobProperties3](#) (see page 984)

GetSoftwarePackageList

The GetSoftwarePackageList method returns the registered software packages in a domain manager. This method supports the sort and filter operations also.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string softwarePackageGroupId

Specifies the UUID of the software package group from which the software packages have to be listed. Specify NULL to list all software packages.

**dsm:SoftwarePackagePropertiesRequired
softwarePackagePropertiesRequired**

Specifies the software package properties you want retrieve from the list.

dsm:ArrayOfSoftwarePackageFilter filterArray

Specifies an array of filter conditions on software package properties.

dsm:SoftwarePackageProperty sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

xsd:long numRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Xsd:boolean refreshList

Specifies whether you want to recompute the result set for each call.

Return Values

xsd:long totalNumPackages

Contains the total number of packages available for the specified filter condition.

dsm:ArrayOfSoftwarePackageProperties2 swPkgArray

Contains an array of software packages properties.

Software Policies

The following are software policy methods.

This section contains the following topics:

[CloseSoftwarePolicyList](#) (see page 399)
[CloseSoftwarePolicyListRecursive](#) (see page 399)
[CloseSoftwarePolicySoftwareJobList](#) (see page 400)
[CloseSoftwarePolicySoftwarePolicyExceptionList](#) (see page 400)
[CreateSoftwarePolicy](#) (see page 401)
[CreateSoftwarePolicy2](#) (see page 401)
[CreateSoftwarePolicy3](#) (see page 402)
[CreateSoftwarePolicyByName](#) (see page 403)
[CreateSoftwarePolicyByName3](#) (see page 404)
[DeleteSoftwarePolicies](#) (see page 405)
[DeleteSoftwarePoliciesByName](#) (see page 406)
[EvaluateSoftwarePolicy](#) (see page 406)
[DeleteSoftwarePoliciesInUnitGroup](#) (see page 407)
[DeleteSoftwarePoliciesInUnitGroupByName](#) (see page 407)
[FindSoftwarePolicy](#) (see page 408)
[GetSoftwarePolicies](#) (see page 409)
[GetSoftwarePolicies3](#) (see page 410)
[GetSoftwarePoliciesRecursive](#) (see page 411)
[GetSoftwarePolicy](#) (see page 411)
[GetSoftwarePolicy3](#) (see page 412)
[GetSoftwarePolicySoftwareJobs](#) (see page 413)
[GetSoftwarePolicySoftwareJobs3](#) (see page 414)
[GetSoftwarePolicySoftwarePolicyExceptions](#) (see page 415)
[LinkSoftwarePoliciesToUnitGroup](#) (see page 415)
[LinkSoftwarePoliciesToUnitGroupByName](#) (see page 416)
[OpenSoftwarePolicyList](#) (see page 416)
[OpenSoftwarePolicyList3](#) (see page 417)
[OpenSoftwarePolicyListRecursive](#) (see page 418)
[OpenSoftwarePolicySoftwareJobList](#) (see page 419)
[OpenSoftwarePolicySoftwarePolicyExceptionList](#) (see page 420)
[SealSoftwarePolicies](#) (see page 421)
[SetSoftwarePolicy](#) (see page 422)
[SetSoftwarePolicy2](#) (see page 422)
[SetSoftwarePolicy3](#) (see page 423)
[SetSoftwarePolicyByName](#) (see page 424)
[SetSoftwarePolicyByName3](#) (see page 425)
[UnlinkSoftwarePoliciesFromUnitGroup](#) (see page 426)
[UnlinkSoftwarePoliciesFromUnitGroupByName](#) (see page 426)
[UnsealSoftwarePolicies](#) (see page 427)
[UnsealSoftwarePoliciesByName](#) (see page 427)

CloseSoftwarePolicyList

The CloseSoftwarePolicyList method closes a software policy list that was opened by OpenSoftwarePolicyList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePolicyList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePolicyListRecursive

The CloseSoftwarePolicyListRecursive method closes a software policy list that was opened by OpenSoftwarePolicyListRecursive and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePolicyListRecursive.

Return Values

This is a void method that does not expose a return value.

Remarks

This API behaves identically to CloseSoftwarePolicyList.

CloseSoftwarePolicySoftwareJobList

The CloseSoftwarePolicySoftwareJobList method closes a software job list that was opened by OpenSoftwarePolicySoftwareJobList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePolicySoftwareJobList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePolicySoftwarePolicyExceptionList

The CloseSoftwarePolicySoftwarePolicyExceptionList method closes a list of policy exception units that was opened by OpenSoftwarePolicySoftwarePolicyExceptionList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePolicySoftwarePolicyExceptionList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateSoftwarePolicy

The CreateSoftwarePolicy method creates a new software policy.

Parameters

xsd:string sessionId (^m)

The session identifier.

CreateSoftwarePolicyProperties softwarePolicy (^m)

The properties required to create a software policy.

Return Values

xsd:string CreateSoftwarePolicyReturn

If a software policy was successfully created, the UUID of the new policy is returned.

Return Values

None

More information:

[CreateSoftwarePolicyProperties](#) (see page 779)

CreateSoftwarePolicy2

This method creates a software policy.

It takes UUID of a unit group as an optional parameter.

If a valid unit Group ID is supplied, the created policy is linked to it.

If not supplied, the software policy is created without a link to any unit group.

Parameters

xsd: string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd: string unitGroupId

UUID of the unit group to which the created policy is linked. If an invalid ID is supplied, the policy is not created.

CreateSoftwarePolicyProperties2 createSoftwarePolicyProperties2 (m)

The sequence *CreateSoftwarePolicyProperties2* contains the properties associated with software policy. Providing all the members of the sequence is not mandatory.

Return Values

xsd:string softwarePolicyId

If the software policy is successfully created, it returns a string which contains the UUID of the newly created policy.

Remarks

None

More information:

[CreateSoftwarePolicyProperties2](#) (see page 693)

CreateSoftwarePolicy3

The CreateSoftwarePolicy3 method creates a software policy. The method takes the UUID of a unit group as an optional parameter. If a valid unit Group ID is provided, the created policy is linked to it. If a valid value is not provided, the software policy is created without a link to any unit group.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence CreateSoftwarePlicyProperties3. The sequence includes two parameters: userMessage, which lets you provide the custom administrator message information, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd: string sessionId (m)

Specifies the session identifier obtained from the login method of the web service.

xsd: string unitGroupId

Specifies the UUID of the unit group to which the created policy is linked. If an invalid ID is supplied, the policy is not created.

CreateSoftwarePolicyProperties3 createSoftwarePolicyProperties3 (m)

Specifies the sequence *CreateSoftwarePolicyProperties3* that contains the properties associated with the software policy. Although the sequence is mandatory, but providing all the members of the sequence is not mandatory.

Return Values

xsd:string softwarePolicyId

Returns a string that contains the UUID of the newly created policy if the software policy is successfully created.

Remarks

None

More information:

[CreateSoftwarePolicyProperties3](#) (see page 781)

CreateSoftwarePolicyByName

Creates the software policy. This method takes the name of the software policy which is to be created as a mandatory parameter.

Name of the unit group (unitGroupName) to which the policy is attached is a mandatory parameter.

If unitGroupName is supplied the API searches for it, and if found, links it with the created policy.

If unitGroupName is supplied the API searches for it, if not found, API creates the unitGroup and links it with the created SoftwarePolicy.

If unitGroupName is not supplied the software policy is created without a link to any unit group.

Parameters

xsd: string sessionId (m)

Specifies the session id obtained from the login to the web service.

UnitGroupPropertiesForSoftwarePolicy

unitGroupPropertiesForSoftwarePolicy

This sequence provides the details of the unit group to which the created software policy is linked. If not supplied the software policy is created without a link to any unit group.

CreateSoftwarePolicyProperties2 createSoftwarePolicyProperties2 (m)

The sequence CreateSoftwarePolicyProperties2 contains the properties associated with software policy. Although the sequence is mandatory, providing all the members of the sequence is not mandatory.

Return Values

xsd:string softwarePolicyId

If the software policy is successfully created, it returns a string which contains the UUID of the newly created policy.

Remarks

None

More information:

[CreateSoftwarePolicyProperties2](#) (see page 693)

CreateSoftwarePolicyByName3

The CreateSoftwarePolicyByName3 method creates a software policy based on the specified name of the software policy. The name of the unit group (unitGroupName) to which the policy is attached is a mandatory parameter. Make sure that you take into consideration the following points related to unitGroupName:

- If unitGroupName is provided, the method searches for it. If the method finds the unit group name, the method links the unit group name with the created policy.
- If unitGroupName is provided, the method searches for it. If the method does not find the unit group name, the method creates the unitGroup and links it with the created software policy.
- If unitGroupName is not provided, the software policy is created without a link to any unit group.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence CreateSoftwarePlicyProperties3. This sequence includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters**xsd:string sessionId (m)**

Specifies the session id obtained from the login to the web service.

UnitGroupPropertiesForSoftwarePolicy**unitGroupPropertiesForSoftwarePolicy**

Specifies the details of the unit group to which the created software policy is linked. If not supplied, the software policy is created without a link to any unit group.

CreateSoftwarePolicyProperties3 createSoftwarePolicyProperties3 (m)

Specifies the properties associated with the software policy. The sequence is mandatory, but providing all the members of the sequence is not mandatory.

Return Values**xsd:string softwarePolicyId**

Returns a string that contains the UUID of the newly created policy if the software policy is successfully created.

Remarks

None

More information:

[CreateSoftwarePolicyProperties3](#) (see page 781)

DeleteSoftwarePolicies

The DeleteSoftwarePolicies method deletes one or more software policies.

Parameters**xsd:string sessionId (m)**

The session identifier.

ArrayOfstring softwarePolicyIds (m)

An array of software policy UUIDs identifying the software policies to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteSoftwarePoliciesByName

The DeleteSoftwarePoliciesByName method deletes one or more software policies. (Corresponding to the existing API DeleteSoftwarePolicies)

DeleteSoftwarePoliciesByName implements a functionality corresponding to the existing WS API DeleteSoftwarePolicies. The difference is that in the new api, the software policies are specified by name instead of UUID. The api deletes the policies corresponding to the list of policy names provided by the user.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring softwarePolicyNames (m)

An array of software policy Names identifying the software policies to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

EvaluateSoftwarePolicy

The EvaluateSoftwarePolicy method evaluates a software policy identified by its UUID.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePolicyId (m)

The UUID identifying the software policy to evaluate.

SoftwarePolicyEvaluationType evaluateType

The type of evaluation to be used for the software policy.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SoftwarePolicyEvaluationType](#) (see page 100)

DeleteSoftwarePoliciesInUnitGroup

This method deletes the software policies available in the unit group. If recursive is set to true, it would delete the policies of the nested unit groups too. This method will not be deleting the unit groups that contain the policies.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupId (m)

Specifies the UUID of the unit group from which the software policies need to be deleted.

xsd:boolean isRecursive (m)

If recursive is set to true then this api would check for policies in all the sub unit groups of the specified unit group. If policies exist in sub unit groups, they too are deleted. By default the parameter is set to false.

Return Values

This is a void method that does not expose a return value.

This method deletes a policy and does not delete a unitGroup. To delete a unitGroup, use DeleteUnitGroups.

DeleteSoftwarePoliciesInUnitGroupByName

This method deletes the software policies available in a unit group. If recursive is set to true, it would delete the policies of the nested unit groups too. This method will not be deleting the unit groups that contain the policies.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupName (m)

Specifies the name of the unit group from which the software policies need to be deleted.

xsd:boolean isRecursive

If recursive is set to true then this api would check for deletion, policies in all the sub unit groups of the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

This method deletes a policy and does not delete a unitGroup. To delete a unitGroup, use DeleteUnitGroups.

FindSoftwarePolicy

The FindSoftwarePolicy method finds a software policy given certain information.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string parentSoftwarePolicyId

The UUID identifying the parent software policy of the policy that should be found.

SoftwarePolicyScope parentSoftwarePolicyScope ("")

The type of the parent software policy of the policy that should be found.

xsd:string softwarePolicyName ("")

The display name of the policy to search for.

Return Values

xsd:string FindSoftwarePolicyReturn

If a software policy is found, its UUID is returned.

Remarks

If the parent UUID is not supplied, this method finds the SoftwarePackageProcedureGroup under the superfolder (for example, Software Library).

More information:

[SoftwarePolicyScope](#) (see page 100)

GetSoftwarePolicies

The GetSoftwarePolicies method retrieves the policies from a software policy list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to OpenSoftwarePolicyList.

xsd:long noOfSoftwarePolicies ("")

The total number of software policies to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePolicyList.

Return Values

ArrayOfSoftwarePolicyProperties GetSoftwarePoliciesReturn

An array of SoftwarePolicyProperties - one for each policy requested from the list.

Remarks

You can make multiple calls to GetSoftwarePolicies to iterate down the list.

More information:

[SoftwarePolicyProperties](#) (see page 851)

GetSoftwarePolicies3

The GetSoftwarePolicies3 method retrieves the policies from a software policy list.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved array of software policy properties in the sequence SoftwarePolicyProperties3. This sequence includes the userMessage parameter that specifies the custom administrator message information linked with the software policy in SD.

Parameters

xsd:string sessionId (")

Specifies the session identifier obtained from the login method of the web service.

xsd:long listHandle (")

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwarePolicyList3.

xsd:long noOfSoftwarePolicies (")

Specifies the total number of software policies to get from the list. This number must be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePolicyList3.

Return Values

ArrayOfSoftwarePolicyProperties3 arrayOfSoftwarePolicyProperties3

Returns an array of SoftwarePolicyProperties3—one for each policy requested from the list.

Remarks

None

More information:

[SoftwarePolicyProperties3](#) (see page 854)

GetSoftwarePoliciesRecursive

The GetSoftwarePoliciesRecursive method retrieves the policies from a software policy list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list as returned by a previous call to OpenSoftwarePoliciesRecursive.

xsd:long noOfSoftwarePolicies ("")

The number of software policies.

Return Values

ArrayOfSoftwarePolicyProperties GetSoftwarePoliciesRecursiveReturn

An array of SoftwarePolicyProperties - one for each policy requested from the list.

Remarks

You can make multiple calls to GetSoftwarePoliciesRecursive to iterate down the list. This API behaves identically to GetSoftwarePolicies.

More information:

[SoftwarePolicyProperties](#) (see page 851)

GetSoftwarePolicy

The GetSoftwarePolicy method retrieves a software policy identified by its UUID.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string softwarePolicyId ("")

The UUID of the software policy to retrieve.

SoftwarePolicyPropertiesRequired softwarePolicyPropertiesRequired (")

This parameter specifies which software policy properties to retrieve.

Return Values

SoftwarePolicyProperties GetSoftwarePolicyReturn

The software policy properties retrieved.

Remarks

None

More information:

[SoftwarePolicyPropertiesRequired](#) (see page 856)

GetSoftwarePolicy3

The GetSoftwarePolicy3 method retrieves a software policy identified by its UUID.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved array of software policy properties in the sequence SoftwarePolicyProperties3. This sequence includes the userMessage parameter that specifies the custom administrator message information linked with the software policy in SD. Additionally, this method takes the sequence SoftwarePolicyPropertiesRequired3 that includes a parameter to set the requirement for the userMessage field in the retrieved software job container properties.

Parameters

xsd:string sessionId (")

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwarePolicyId (")

Specifies the UUID of the software policy to retrieve.

SoftwarePolicyPropertiesRequired3 softwarePolicyPropertiesRequired3 (")

Specifies which software policy properties to retrieve.

Return Values**SoftwarePolicyProperties3 softwarePolicyProperties3**

Returns the retrieved software policy properties for the specified software policy ID.

Remarks

None

More information:

[SoftwarePolicyPropertiesRequired3](#) (see page 858)

[SoftwarePolicyProperties3](#) (see page 854)

GetSoftwarePolicySoftwareJobs

The GetSoftwarePolicySoftwareJobs method retrieves the software jobs from a software policy's job list.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePolicySoftwareJobList.

xsd:long noOfSoftwareJobs (")

The total number of jobs to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePolicySoftwareJobList.

Return Values**ArrayOfSoftwareJobProperties GetSoftwarePolicySoftwareJobsReturn**

An array of SoftwareJobProperties - one for each job requested from the list.

Remarks

You can make multiple calls to GetSoftwarePolicySoftwareJobs to iterate down the list.

More information:

[SoftwareJobProperties](#) (see page 768)

GetSoftwarePolicySoftwareJobs3

The GetSoftwarePolicySoftwareJobs3 method retrieves the software jobs from a job list of a software policy.

Additionally, this method retrieves the ShutdownAfterLast property that includes information on whether a target computer is to be shut down after completion of the last SD job. The method returns a retrieved list of software job properties in the sequence SoftwareJobProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast.

This method also retrieves the userMessage property that includes the custom administrator message linked with the software job in SD. The sequence SoftwareJobProperties3 includes the option userMessage.

Parameters

xsd:string sessionId (")

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (")

Specifies a numeric handle to a list as returned by a previous call to OpenSoftwarePolicySoftwareJobList.

xsd:long noOfSoftwareJobs (")

Specifies the total number of jobs to get from the list. The number must be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePolicySoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties3 arrayOfSoftwareJobProperties3

Returns an array of SoftwareJobProperties3—one for each job requested from the list.

Remarks

None

More information:

[SoftwareJobProperties3](#) (see page 774)

GetSoftwarePolicySoftwarePolicyExceptions

The GetSoftwarePolicySoftwarePolicyExceptions method retrieves the exception units from a software policy's exception list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list as returned by a previous call to OpenSoftwarePolicySoftwarePolicyExceptionList.

xsd:long noOfSoftwarePolicyExceptions (")

The total number of exception units to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePolicySoftwarePolicyExceptionList.

Return Values

ArrayOfSoftwarePolicyExceptionProperties

GetSoftwarePolicySoftwarePolicyExceptionsReturn

An array of SoftwarePolicyExceptionProperties - one for each exception unit requested from the list.

Remarks

You can make multiple calls to GetSoftwarePolicySoftwarePolicyExceptions to iterate down the list.

More information:

[SoftwarePolicyExceptionProperties](#) (see page 851)

LinkSoftwarePoliciesToUnitGroup

Links a list of software policies, specified by their respective uuids to a target unit group also specified by its uuid.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupId (m)

The uuid of the unit group to which the software policies have to be linked.

ArrayOfstring softwarePolicyIds (m)

The list of software policy uuids that have to be linked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkSoftwarePoliciesToUnitGroupByName

Links a list of software policies, specified by their respective names to a target unit group also specified by its name.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupName (m)

The name of the unit group to which the software policies are to be linked.

ArrayOfstring softwarePolicyNames (m)

The list of software policy names that have to be linked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

OpenSoftwarePolicyList

The OpenSoftwarePolicyList method opens a list of software policies.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwarePolicyId

Ignore this parameter.

SoftwarePolicyPropertiesRequired softwarePolicyPropertiesRequired (")

The properties that will be required when getting a software policy from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of software policies that can be passed to GetSoftwarePolicies

xsd:long totalNoOfSoftwarePolicies

The total number of software policies in the list whose handle has been returned.

Remarks

The method returns the total number of software policies in the list, and a handle to the list that can be used for further calls to GetSoftwarePolicies.

More information:

[SoftwarePolicyPropertiesRequired](#) (see page 856)

OpenSoftwarePolicyList3

The OpenSoftwarePolicyList3 method returns the total number of software policies in the list, and a handle to the list that you can use for further calls to GetSoftwarePolicies3.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence SoftwarePolicyPropertiesRequired3. This sequence uses a parameter to set the requirement for the userMessage field in the retrieved software job container properties.

Parameters**xsd:string sessionId (")**

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwarePolicyId

Ignore this parameter.

SoftwarePolicyPropertiesRequired3 softwarePolicyPropertiesRequired3 (")

Specifies the properties that are required when getting a software policy from the list.

Return Values

xsd:long OpenListHandle

Returns a numeric handle to a list of software policies that you can pass to GetSoftwarePolicies.

xsd:long totalNoOfSoftwarePolicies

Returns the total number of software policies in the list for which the handle has been returned.

Remarks

None

More information:

[SoftwarePolicyPropertiesRequired3](#) (see page 858)

OpenSoftwarePolicyListRecursive

The OpenSoftwarePolicyListRecursive method opens a list of software policies, and of any software policies recursively contained within that policy (if any).

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePolicyId

Ignore this parameter.

SoftwarePolicyPropertiesRequired softwarePolicyPropertiesRequired (")

The properties that will be required when getting a software policy from the list.

Return Values**xsd:long OpenListHandletotalNoOfSoftwarePolicies**

A numeric handle to a list of software policies that can be passed to GetSoftwarePolicyListRecursive

Remarks

This API behaves identically to OpenSoftwarePolicyList.

More information:

[SoftwarePolicyPropertiesRequired](#) (see page 856)

OpenSoftwarePolicySoftwareJobList

The OpenSoftwarePolicySoftwareJobList method opens a list of the software jobs that are contained by the given software policy.

The method returns the total number of software jobs in the list, and a handle to the list that can be used for further calls to GetSoftwarePolicySoftwareJobs

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string softwarePolicyId (")

The UUID of the software policy whose member jobs should be listed.

SoftwareJobPropertiesRequired softwareJobPropertiesRequired (^m)

The properties that will be required when getting a job from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of jobs that can be passed to
GetSoftwarePolicySoftwareJobsResponse

xsd:long totalNoOfSoftwareJobs

The total number of jobs in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobPropertiesRequired](#) (see page 777)

OpenSoftwarePolicySoftwarePolicyExceptionList

The OpenSoftwarePolicySoftwarePolicyExceptionList method opens a list of the units that have violated the given software policy.

The method returns the total number of units in the list, and a handle to the list that can be used for further calls to
GetSoftwarePolicySoftwarePolicyExceptions.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string softwarePolicyId (^m)

The UUID of the software policy whose exceptions should be listed.

**SoftwarePolicyExceptionPropertiesRequired
softwarePolicyExceptionPropertiesRequired (")**

The properties that will be required when getting a unit from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of units that can be passed to
GetSoftwarePolicySoftwarePolicyExceptions

xsd:long totalNoOfSoftwarePolicyExceptions

The total number of units in the list whose handle has been returned.

Remarks

None

More information:

[SoftwarePolicyExceptionProperties](#) (see page 851)

SealSoftwarePolicies

The SealSoftwarePolicies method seals and activates one or more software job policies.

Parameters**xsd:string sessionId (")**

The session identifier.

ArrayOfstring softwarePolicyIds (")

An array of UUIDs identifying the software policies to be sealed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetSoftwarePolicy

The SetSoftwarePolicy method updates the properties of a software policy.

Parameters

xsd:string sessionId (^m)

The session identifier.

SetSoftwarePolicyProperties softwarePolicy (^m)

The properties that should be updated for the specified software policy.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePolicyProperties](#) (see page 758)

SetSoftwarePolicy2

This method updates the properties of a specified software policy.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwarePolicyId (m)

The UUID of the software policy which needs to be updated.

SetSoftwarePolicyProperties2 setSoftwarePolicyProperties (m)

The properties of the software policy which needs to be updated. Though providing the sequence is mandatory, providing values for all its members is not mandatory. The software policy properties which are not specified, retain their original values.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePolicyProperties2](#) (see page 732)

SetSoftwarePolicy3

The SetSoftwarePolicy3 method updates the properties of a specified software policy.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence SetSoftwarePolicyProperties3. This sequence includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (m)

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwarePolicyId (m)

Specifies the UUID of the software policy that needs to be updated.

SetSoftwarePolicyProperties3 setSoftwarePolicyProperties3 (m)

Specifies the properties of the software policy that needs to be updated. Though providing the sequence is mandatory, but providing values for all of its members is not mandatory. The software policy properties which are not specified, retain their original values.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePolicyProperties3](#) (see page 759)

SetSoftwarePolicyByName

This method updates specified software policy properties. The software policy is specified by name.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

String softwarePolicyName (m)

The name of the Software Policy. In a particular domain manager or enterprise manager software policy name is unique if it is local to the manager.

SetSoftwarePolicyPropertiesByName

setSoftwarePolicyPropertiesByName (m)

The properties of the software policy that needs to be updated. Though providing the sequence is mandatory, providing values for all its members is not mandatory. The software policy properties that are not specified retain their original values.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePolicyPropertiesByName](#) (see page 761)

SetSoftwarePolicyByName3

The SetSoftwarePolicyByName3 method updates the specified software policy properties. The software policy is specified by name.

This method also includes the functionality that helps you manage the custom administrator message. The method takes the sequence SetSoftwarePolicyPropertiesByName3. This sequence includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (m)

Specifies the session identifier obtained from the login method of the web service.

xsd:string softwarePolicyName (m)

Specifies the name of the software policy. In a particular domain manager or an enterprise manager, the software policy name is unique if it is local to the manager.

SetSoftwarePolicyPropertiesByName3

setSoftwarePolicyPropertiesByName3 (m)

Specifies the properties of the software policy that you need to update. Though providing the sequence is mandatory, but providing values for all of its members is not mandatory. The software policy properties that are not specified retain their original values.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwarePolicyPropertiesByName3](#) (see page 763)

UnlinkSoftwarePoliciesFromUnitGroup

Unlinks a list of software policies specified by their respective UUIDs from a target unit group, also specified by its UUID.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string unitGroupId (m)

Specifies the UUID of the group from which to unlink the software policies.

ArrayOfstring softwarePolicyIds (m)

Specifies the list of software policy IDs to be unlinked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkSoftwarePoliciesFromUnitGroupByName

Unlinks a list of software policies specified by their respective names from a target unit group, also specified by its name.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string unitGroupName (m)

Specifies the name of the group from which to unlink the software policies.

ArrayOfstring softwarePolicyNames (m)

Specifies the list of software policy names to be unlinked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnsealSoftwarePolicies

The UnsealSoftwarePolicies method unseals one or more software policies.

Parameters

xsd:string sessionId (m)

The session identifier.

ArrayOfstring softwarePolicyIds (m)

An array of UUIDs identifying the software policies to be unsealed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnsealSoftwarePoliciesByName

The UnsealSoftwarePoliciesByName method unseals one or more software policies.(Corresponding to the existing API UnSealSoftwarePolicies)

Unseals the specified software policy. This Api does the same functionality as UnsealSoftwarePolicies.

Parameters

xsd: string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring softwarePolicyNames (m)

An array of Names identifying the software policies to be unsealed.

Return Values

This is a void method that does not expose a return value.

Remarks

None

Priority, Reinstall, and Policy Enhancement

This section contains the following topics:

[CloseParentSoftwareJobContainerList2](#) (see page 428)
[CloseSoftwareJobContainerListRecursive2](#) (see page 429)
[CloseSoftwarePolicyList2](#) (see page 429)
[CreateInstallSoftwareJob2](#) (see page 430)
[CreateInstallSoftwareJobInSoftwarePolicy2](#) (see page 431)
[CreateInstallSoftwareJobInSoftwarePolicyByName](#) (see page 432)
[CreateInstallSoftwareJobInSoftwarePolicyByName3](#) (see page 433)
[CreateSoftwareJobContainer2](#) (see page 434)
[GetParentSoftwareJobContainers2](#) (see page 435)
[GetSoftwareJob2](#) (see page 436)
[GetSoftwareJobContainer2](#) (see page 436)
[GetSoftwareJobContainers2](#) (see page 437)
[GetSoftwareJobContainerSoftwareJobs2](#) (see page 438)
[GetSoftwareJobContainersRecursive2](#) (see page 439)
[GetSoftwareJobs2](#) (see page 440)
[GetSoftwarePolicies2](#) (see page 441)
[GetSoftwarePolicy2](#) (see page 441)
[GetSoftwarePolicySoftwareJobs2](#) (see page 442)
[OpenParentSoftwareJobContainerList2](#) (see page 443)
[OpenSoftwareJobContainerList2](#) (see page 444)
[OpenSoftwareJobContainerListRecursive2](#) (see page 445)
[OpenSoftwarePolicyList2](#) (see page 446)
[SetSoftwareJob2](#) (see page 447)
[SetSoftwareJobContainer2](#) (see page 447)
[SetSoftwareJobContainerPriority](#) (see page 448)

CloseParentSoftwareJobContainerList2

The CloseParentSoftwareJobContainerList2 method closes a software job container list that was opened by OpenParentSoftwareJobContainerList2 and releases the list handle.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenParentSoftwareJobContainerList2.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwareJobContainerListRecursive2

The CloseSoftwareJobContainerListRecursive2 method closes a software job container list that was opened by OpenSoftwareJobContainerListRecursive2 and releases the list handle.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwareJobContainerListRecursive2.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseSoftwarePolicyList2

The CloseSoftwarePolicyList2 method closes a software policy list that was opened by OpenSoftwarePolicyList2 and releases the list handle.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenSoftwarePolicyList2.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateInstallSoftwareJob2

The CreateInstallSoftwareJob2 method creates a new software job of type 'Install'. The method takes SoftwareJobOperationMask2 sequence that has an extra flag reinstall. Please check SoftwareJobOperationMask2 sequence for more details.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId (m)

The UUID identifying the software package procedure that should be used by the job.

CreateInstallSoftwareJobOrderProperties createInstallSoftwareJobOrderProperties (m)

The properties required to create and order a software job.

xsd:string softwarejobContainerId (m)

The UUID identifying the job container associated with this new software job.

ArrayOfstring unitGroupIds

An array of UUIDs identifying the unit groups to which the job should be applied.

ArrayOfstring unitIds

An array of UUIDs identifying the units to which the job should be applied.

Return Values

xsd:string softwareJobId

If a software Install job was successfully created, the UUID of the new job is returned.

Remarks

You must specify UnitIds or UnitGroupids or both.

More information:

[CreateInstallSoftwareJobOrderProperties](#) (see page 636)

CreateInstallSoftwareJobInSoftwarePolicy2

The SetSoftwareJob2 method updates the properties of a software job. The method takes the SoftwareJobOperationMask2 sequence, which has a reinstall flag.

Parameters**xsd:string sessionId (m)**

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePackageProcedureId (m)

The UUID identifying the software package procedure that should be used by the job.

**CreateInstallSoftwareJobOrderPolicyProperties
createInstallSoftwareJobOrderPolicyProperties (m)**

The properties required to create and order a software job in a software policy.

xsd:string softwarePolicyId (m)

The UUID identifying the software policy in which the software job should be created.

Return Values**xsd:string softwareJobId**

If an 'Install' software job was successfully created, the UUID of the new job is returned.

Remarks

None

More information:

[CreateInstallSoftwareJobOrderPolicyProperties](#) (see page 632)

CreateInstallSoftwareJobInSoftwarePolicyByName

The CreateInstallSoftwareJobInSoftwarePolicyByName method creates a new software job of type 'Install' within a software policy.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

SoftwarePackage softwarePackage (m)

Specifies the software package details.

xsd:string softwarePackageProcedureName (m)

Specifies the name of install procedure for the software package.

CreateInstallSoftwareJobOrderPolicyProperties

installSoftwareJobOrderPolicyProperties (m)

Software job order policy properties are specified.

xsd:string softwarePolicyName (m)

Name of the Software policy.

Return Values

xsd:string installSoftwareJobId

The UUID of the install software job is returned to the user.

Remarks

None

More information:

[CreateInstallSoftwareJobOrderPolicyProperties](#) (see page 632)

CreateInstallSoftwareJobInSoftwarePolicyByName3

The CreateInstallSoftwareJobInSoftwarePolicyByName3 method creates a new software job of type Install within a software policy.

Additionally, this method provides the functionality to shut down a target computer after completion of the jobs. The method takes the sequence CreateInstallSoftwareJobOrderPolicyProperties3, which includes another sequence SoftwareJobBootMask3. The SoftwareJobBootMask3 sequence contains the option ShutdownAfterLast that lets you shut down the target computer after completion of the jobs.

This method also includes the functionality that helps you manage the custom administrator message. For the custom administrator message, the sequence CreateInstallSoftwareJobOrderPolicyProperties3 includes two parameters: userMessage, which takes the message input from the user, and userMessageSupplied, which checks the presence of the userMessage field for the custom administrator message in SD.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

SoftwarePackage softwarePackage (m)

Specifies the software package details.

xsd:string softwarePackageProcedureName (m)

Specifies the name of the install procedure for the software package.

CreateInstallSoftwareJobOrderPolicyProperties3

installSoftwareJobOrderPolicyProperties3 (m)

Specifies the properties required to create and order a software job in a software policy.

xsd:string softwarePolicyName (m)

Specifies the name of the software policy in which you must create the software job.

Return Values

xsd:string softwareJobId

Returns the UUID of the new job if a software Install job is successfully created.

Remarks

None

More information:

[CreateInstallSoftwareJobOrderPolicyProperties](#) (see page 632)

CreateSoftwareJobContainer2

This method creates a new software job container. The difference between the new and old (CreateSoftwareJobContainer) method is that the new method takes a new structure, CreateSoftwareJobContainerProperties2, which has a new member 'priority'. Using this you can set the priority of the container.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

CreateSoftwareJobContainerProperties2 softwarejobContainer2(m)

The properties required to create a software job container.

Return Values

xsd:string softwareJobContainerId

If a software job container was successfully created, the UUID of the new container is returned.

Remarks

None

More information:

[CreateSoftwareJobContainerProperties2](#) (see page 216)

GetParentSoftwareJobContainers2

The GetParentSoftwareJobContainers2 method retrieves the parent software job containers from a software job containers's parent group list.

The difference between the new and old (GetParentSoftwareJobContainers) API is that the new API retrieves the priority of the container also.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenParentSoftwareJobContainerList2.

xsd:long noOfParentSoftwareJobContainers (m)

The total number of software job containers to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenParentSoftwareJobContainerList2.

Return Values

ArrayOfSoftwareJobContainerProperties2

arrayOfSoftwareJobContainerProperties2

An array of SoftwareJobContainerProperties2 - one for each parent container requested from the list.

Remarks

You can make multiple calls to GetParentSoftwareJobContainers2 to iterate down the list.

More information:

[SoftwareJobContainerProperties2](#) (see page 860)

GetSoftwareJob2

The GetSoftwareJob2 method retrieves a software job identified by its UUID. The method returns SoftwareJobOperationMask2 sequence that has an extra flag reinstall. Please check SoftwareJobOperationMask2 sequence for more details.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobId (m)

The UUID of the software package to retrieve.

SoftwareJobPropertiesRequired softwareJobPropertiesRequired (m)

Specifies which software job properties to retrieve.

Return Values

SoftwareJobProperties2 softwareJobProperties2

The software job properties retrieved.

Remarks

None

More information:

[SoftwareJobPropertiesRequired](#) (see page 777)

[SoftwareJobProperties2](#) (see page 771)

GetSoftwareJobContainer2

This method retrieves a software job container identified by its UUID. The difference between the new and old (GetSoftwareJobContainer) API is that the new API retrieves the 'priority' of the container also.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobContainerId (m)

The UUID of the software job container to retrieve.

SoftwareJobContainerPropertiesRequired2
softwareJobContainerPropertiesRequired2 (m)

Specifies which software job container properties to retrieve.

Return Values**SoftwareJobContainerProperties2 softwareJobContainerProperties2**

The software job container properties retrieved.

Remarks

None

More information:

[SoftwareJobContainerProperties2](#) (see page 860)

[SoftwareJobContainerPropertiesRequired2](#) (see page 741)

GetSoftwareJobContainers2

The GetSoftwareJobContainers2 method retrieves the job containers from a software job container list. The difference between the new and old (GetSoftwareJobContainers) API is that the new API retrieves the 'priority' of the container also.

Parameters**xsd:string sessionId (m)**

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenSoftwareJobContainerList2.

xsd:long noOfSoftwareJobContainers (m)

The total number of software job containers to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobContainerList2.

Return Values**ArrayOfSoftwareJobContainerProperties2**
arrayOfSoftwareJobContainerProperties2

An array of SoftwareJobContainerProperties2 - one for each container requested from the list.

Remarks

You can make multiple calls to GetSoftwareJobContainers2 to iterate down the list.

More information:

[SoftwareJobContainerProperties2](#) (see page 860)

GetSoftwareJobContainerSoftwareJobs2

The GetSoftwareJobContainerSoftwareJobs2 method retrieves the software jobs from a software job container's job list. The method returns SoftwareJobOperationMask2 sequence that has an extra flag reinstall. See SoftwareJobOperationMask2 sequence for more details.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenSoftwareJobContainerSoftwareJobList.

xsd:long noOfSoftwareJobs (m)

The total number of jobs to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobContainerSoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties2 arrayOfSoftwareJobProperties2

An array of SoftwareJobProperties2 - one for each job requested from the list.

Remarks

You can make multiple calls to GetSoftwareJobContainerSoftwareJobs2 to iterate down the list.

More information:

[SoftwareJobProperties](#) (see page 768)

GetSoftwareJobContainersRecursive2

The GetSoftwareJobContainersRecursive2 method retrieves the job containers from a software job container list.

The difference between the new and old (GetSoftwareJobContainerRecursive) API is that the new API retrieves the 'priority' of the container also.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenSoftwareJobContainersRecursive2.

xsd:long noOfSoftwareJobContainers (m)

The number of software job containers.

Return Values

ArrayOfSoftwareJobContainerProperties2

arrayOfSoftwareJobContainerProperties2

An array of SoftwareJobContainerProperties2 - one for each container requested from the list.

Remarks

You can make multiple calls to GetSoftwareJobContainersRecursive2 to iterate down the list.

More information:

[SoftwareJobContainerProperties2](#) (see page 860)

GetSoftwareJobs2

The GetSoftwareJobs2 method retrieves the software jobs from a software job list. The method returns SoftwareJobOperationMask2 sequence that has an extra flag reinstall. Please check SoftwareJobOperationMask2 sequence for more details.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenSoftwareJobList.

xsd:long noOfSoftwareJobs (m)

The total number of software jobs to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties2 arrayOfSoftwareJobProperties2

An array of SoftwareJobProperties2 - one for each software job requested from the list.

Remarks

You can make multiple calls to GetSoftwareJobs2 to iterate down the list.

More information:

[SoftwareJobProperties2](#) (see page 771)

More Information:

[CreateSoftwarePolicyProperties2](#) (see page 693)

GetSoftwarePolicies2

The GetSoftwarePolicies2 method retrieves the policies from a software policy list.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenSoftwarePolicyList.

xsd:long noOfSoftwarePolicies (m)

The total number of software policies to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePolicyList2.

Return Values

ArrayOfSoftwarePolicyProperties2 arrayOfSoftwarePolicyProperties2

An array of SoftwarePolicyProperties2 - one for each policy requested from the list.

Remarks

You can make multiple calls to GetSoftwarePolicies2 to iterate down the list.

GetSoftwarePolicy2

The GetSoftwarePolicy2 method retrieves a software policy identified by its UUID.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePolicyId (m)

The UUID of the software policy to retrieve.

SoftwarePolicyPropertiesRequired2 softwarePolicyPropertiesRequired2 (m)

Specifies which software policy properties to retrieve.

Return Values

SoftwarePolicyProperties2 softwarePolicyProperties2

The software policy properties retrieved.

Remarks

None

More information:

[SoftwarePolicyPropertiesRequired2](#) (see page 857)

[SoftwarePolicyProperties2](#) (see page 865)

GetSoftwarePolicySoftwareJobs2

The GetSoftwarePolicySoftwareJobs2 method retrieves the software jobs from a software policy's job list. The method returns SoftwareJobOperationMask2 sequence that has an extra flag reinstall. Please check SoftwareJobOperationMask2 sequence for more details.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenSoftwarePolicySoftwareJobList.

xsd:long noOfSoftwareJobs (m)

The total number of jobs to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenSoftwarePolicySoftwareJobList.

Return Values

ArrayOfSoftwareJobProperties2 arrayOfSoftwareJobProperties2

An array of SoftwareJobProperties2 - one for each job requested from the list.

Remarks

You can make multiple calls to GetSoftwarePolicySoftwareJobs2 to iterate down the list.

OpenParentSoftwareJobContainerList2

The OpenParentSoftwareJobContainerList2 method opens a list of the software job containers that are parent containers of the software job container.

The method returns the total number of software job containers in the list, and a handle to the list that you can use for further calls to GetParentSoftwareJobContainers2.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobContainerId (m)

The UUID of the software job container whose parent containers should be listed.

SoftwareJobContainerPropertiesRequired2

softwareJobContainerPropertiesRequired2 (m)

The properties that are required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software job containers that can be passed to GetParentSoftwareJobContainers2.

xsd:long totalNoOfParentSoftwareJobContainers

The total number of parent software job containers in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired2](#) (see page 741)

OpenSoftwareJobContainerList2

The OpenSoftwareJobContainerList2 method opens a list of software job containers.

The method returns the total number of software job containers in the list, and a handle to the list that you can use for further calls to GetSoftwareJobContainers2.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobContainerId (m)

The software job container identifier. If not supplied, the root container is opened.

SoftwareJobContainerPropertiesRequired2

softwareJobContainerPropertiesRequired2 (m)

The properties that are required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software job containers that can be passed to GetSoftwareJobContainers.

xsd:long totalNoOfSoftwareJobContainers

The total number of software job containers in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired2](#) (see page 741)

OpenSoftwareJobContainerListRecursive2

This method opens a list of software job containers.

The method returns the total number of software job containers in the list, and a handle to the list that you can use for further calls to `GetSoftwareJobContainersRecursive2`.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobContainerId

The software job container identifier. If not supplied, the root container is opened.

SoftwareJobContainerPropertiesRequired2

softwareJobContainerPropertiesRequired2 (m)

The properties that are required when getting a software job container from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software job containers that can be passed to `GetSoftwareJobContainersRecursive2`.

totalNoOfSoftwareJobContainers

The total number of software job containers in the list whose handle has been returned.

Remarks

None

More information:

[SoftwareJobContainerPropertiesRequired2](#) (see page 741)

OpenSoftwarePolicyList2

The OpenSoftwarePolicyList2 method opens a list of software policies.

The method returns the total number of software policies in the list, and a handle to the list that you can use for further calls to GetSoftwarePolicies2.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwarePolicyId

Ignore this parameter.

SoftwarePolicyPropertiesRequired2 softwarePolicyPropertiesRequired2 (m)

The properties that are required when getting a software policy from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software policies that can be passed to GetSoftwarePolicies.

xsd:long totalNoOfSoftwarePolicies

The total number of software policies in the list whose handle has been returned.

Remarks

None

More information:

[SoftwarePolicyPropertiesRequired2](#) (see page 857)

SetSoftwareJob2

The SetSoftwareJob2 method updates the properties of a software job. The method takes the SoftwareJobOperationMask2 sequence, which has an extra flag reinstall. See the sequence SoftwareJobOperationMask2 for more details.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string softwareJobId (m)

The UUID of the software job to be updated.

SetSoftwareJobProperties2 setSoftwareJobProperties2 (m)

The properties that should be updated for the specified software job.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareJobProperties2](#) (see page 743)

SetSoftwareJobContainer2

This method updates the properties of a software job container. The difference between the new and old (SetSoftwareJobContainer) method is that the new method takes a new structure, SetSoftwareJobContainerProperties2, which has a new member 'priority'. Using this you can set the priority of the container.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobContainerId (m)

The UUID of the container whose properties are to be modified.

SetSoftwareJobContainerProperties2 softwareJobContainer2(m)

The properties that should be updated for the specified software job container.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareJobContainerProperties2](#) (see page 731)

SetSoftwareJobContainerPriority

This method sets the priority of a software job container.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string softwareJobContainerId (m)

The UUID of the software job container whose priority is to be set.

xsd:long priority (m)

Can be 1 (highest) to 10 (lowest).

Return Values

This is a void method that does not expose a return value.

Remarks

None

OSIM

The DSM Web Services uses the methods listed in the subsequent section for OSIM functionality.

Considerations for Using OSIM Web Services

Consider the following factors when you use the OSIM web services:

- To use the ByName version of OSIM methods, the OSIM registered computers must have unique labels.
- A single session cannot have multiple listhandles opened simultaneously by a given method. You must close the listhandles opened by a method before opening it again using the same method. For example, within a given session, if you have opened a listhandle by calling the `OpenOSImageList` method, you must call the `CloseOSImageList` method before you make a call to the `OpenOSImageList` method.

CreateOSIMComputer

The `CreateOSIMComputer` method is used to create a new computer for OS deployment.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

dsm:createOSIMComputerProperties

Defines the [properties to create a computer](#) (see page 812).

Return Values

xsd:string computerId

Returns the UUID if a computer is created.

Remarks

This web service preregisters a new computer.

RegisterComputerWithOSIM

The `RegisterComputerWithOSIM` method is used to register the CA ITCM computer for OSIM deployment.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:string OSIMTargetID

Specifies the UUID of the computer.

xsd:string osimImage

Specifies the OSIM image.

Return Values

None

Remarks

None

RegisterComputerWithOSIMByName

The RegisterComputerWithOSIMByName method is used to register the CA ITCM computer for OSIM deployment using the host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string osimTargetName

Specifies the host name of the target computer.

xsd:string osimImage

Specifies the OSIM image.

Return Values

None

Remarks

None

ActivateOSInstallation

The ActivateOSInstallation method activates a planned installation request for execution.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

xsd:boolean bWakeOnLAN

Specifies that the boot server must send a wake on LAN to the target computer.

xsd:boolean bReboot

Specifies that the boot server must send a reboot request to the target computer.

Note: For the target computer to reboot, the software delivery agent must be installed.

xsd:boolean bWaitForBootServer

Specifies that the execution must wait until the target computer is assigned a boot server.

dsm:WS_OSIM_DEPLOY bWaitForImages

Specifies if the manager must [wait for the OS images](#) (see page 111).

dsm:DateTime activateTime

Specifies the date and time for the installation. If start date is not specified, the installation begins immediately.

Return Values

None

Remarks

None

ActivateOSInstallationByName

The ActivateOSInstallationByName method activates a planned installation request for execution by using the host name.

Parameters**xsd:string sessionId**

Specifies the session id obtained from the login to the Web Service.

xsd:String OSIMTargetComputerLabel

Specifies the computer label of the target computer.

xsd:boolean bWakeOnLAN

Specifies the boot server to send a wake on LAN to the target computer.

xsd:boolean bReboot

Specifies the boot server to send a reboot request to the target computer.

Note: For the target computer to reboot, the software delivery agent must be installed.

xsd:boolean bWaitForBootServer

Specifies that the execution must wait until the target computer is assigned a boot server.

dsm:WS_OSIM_DEPLOY bWaitForImages

Specifies if the manager must [wait for the OS images](#) (see page 111).

dsm:DateTime activateTime

Specifies the date and time for the installation. If the start date is not specified, the installation begins immediately.

Return Values

None

Remarks

None

CancelOSInstallation

The CancelOSInstallation method cancels the OS installation.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

xsd:bool bForceCancel

Specifies that the OS installation must be canceled.

Note: You cannot cancel an OS installation job that in *Installing* state.

Return Values

None

Remarks

None

CancelOSInstallationByName

The CancelOSInstallationByName method cancels the OS installation of the give host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:String OSIMTargetComputerLabel

Specifies the computer label of the target computer.

xsd:bool bForceCancel

Specifies that the OS installation must be canceled.

Note: You cannot cancel an OS installation job that in *Installing* state.

Return Values

None

Remarks

None

ReActivateOSInstallation

The ReActivateOSInstallation method reactivates a canceled or a failed OS installation.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

xsd:boolean bWakeOnLAN

Specifies that the boot server must send a wake on LAN to the target computer.

xsd:boolean wakeOnLANSupplied

Specifies if the bWakeOnLAN value is specified.

xsd:boolean bReboot

Specifies that the boot server must send a reboot request to the target computer.

Note: For the target computer to reboot, the software delivery agent must be installed.

xsd: boolean rebootSupplied

Specifies if the bReboot value is specified.

xsd:boolean bWaitForBootServer

Specifies that the execution must wait until the target computer is assigned a boot server.

xsd:boolean waitForBootServerSupplied

Specifies if the bWaitForBootServer value is specified.

dsm:WS_OSIM_DEPLOY bWaitForImages

Specifies if the manager must [wait for the OS images](#) (see page 111).

xsd:boolean waitForImagesSupplied

Specifies if the bWaitForImages value is specified.

dsm:DateTime activateTime

Specifies the date and time for the installation. If start date is not specified, the installation begins immediately.

Return Values

None

Remarks

None

ReActivateOSInstallationByName

The ReActivateOSInstallationByName method reactivates a canceled or a failed OS installation of the given target computer.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the Web Service.

xsd:String OSIMTargetComputerLabel

Specifies the computer label of the target computer.

xsd:boolean bWakeOnLAN

Specifies the boot server to send a wake on LAN to the target computer.

xsd:boolean wakeOnLANSupplied

Specifies if the bWakeOnLAN value is specified.

xsd:boolean bReboot

Specifies the boot server to send a reboot request to the target computer.

Note: For the target computer to reboot, the software delivery agent must be installed.

xsd: boolean rebootSupplied

Specifies if the bReboot value is specified.

xsd:boolean bWaitForBootServer

Specifies that the execution must wait until the target computer is assigned a boot server.

xsd:boolean waitForBootServerSupplied

Specifies if the bWaitForBootServer value is specified.

dsm:WS_OSIM_DEPLOY bWaitForImages

Specifies if the manager must [wait for the OS images](#) (see page 111).

xsd:boolean waitForImagesSupplied

Specifies if the bWaitForImages value is specified.

dsm:DateTime activateTime

Specifies the date and time for the installation. If the start date is not specified, the installation begins immediately.

Return Values

None

Remarks

None

ReInstallOS

The ReInstallOS method reinstalls the current OS.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

xsd:boolean bWakeOnLAN

Specifies that the boot server must send a wake on LAN to the target computer.

xsd:boolean wakeOnLANSupplied

Specifies if the bWakeOnLAN value is specified.

xsd:boolean bReboot

Specifies that the boot server must send a reboot request to the target computer.

Note: For the target computer to reboot, the software delivery agent must be installed.

xsd: boolean rebootSupplied

Specifies if the bReboot value is specified.

xsd:boolean bWaitForBootServer

Specifies that the execution must wait until the target computer is assigned a boot server.

xsd:boolean waitForBootServerSupplied

Specifies if the bWaitForBootServer value is specified.

dsm:WS_OSIM_DEPLOY bWaitForImages

Specifies if the manager must [wait for the OS images](#) (see page 111).

xsd:boolean waitForImagesSupplied

Specifies if the bWaitForImages value is specified.

dsm:DateTime activateTime

Specifies the date and time for the installation. If start date is not specified, the installation begins immediately.

Return Values

None

Remarks

None

ReInstallOSByName

The ReInstallOSByName method reinstalls the current OS by the host name of the target computer.

Parameters**xsd:string sessionId**

Specifies the session id obtained from the login to the Web Service.

xsd:String OSIMTargetComputerLabel

Specifies the computer label of the target computer.

xsd:boolean bWakeOnLAN

Specifies the boot server to send a wake on LAN to the target computer.

xsd:boolean wakeOnLANSupplied

Specifies if the bWakeOnLAN value is specified.

xsd:boolean bReboot

Specifies the boot server to send a reboot request to the target computer.

Note: For the target computer to reboot, the software delivery agent must be installed.

xsd: boolean rebootSupplied

Specifies if the bReboot value is specified.

xsd:boolean bWaitForBootServer

Specifies that the execution must wait until the target computer is assigned a boot server.

xsd:boolean waitForBootServerSupplied

Specifies if the bWaitForBootServer value is specified.

dsm:WS_OSIM_DEPLOY bWaitForImages

Specifies if the manager must [wait for the OS images](#) (see page 111).

xsd:boolean waitForImagesSupplied

Specifies if the bWaitForImages value is specified.

dsm:DateTime activateTime

Specifies the date and time for the installation. If the start date is not specified, the installation begins immediately.

Return Values

None

Remarks

None

DeregisterOSIMComputer

The DeRegisterOSIMComputer method removes a target computer from OSIM managed systems using the MAC address of the target computer.

Note: This method does not delete the target computer from the CA ITCM MDB. If you do not remove the computer from the network and do not disable PXE, the computer reregisters with OSIM during the boot process.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the OSIM computer.

Return Values

None

Remarks

None

DeRegisterOSIMComputerByName

The DeRegisterOSIMComputerByName method removes a target computer from the OSIM managed systems using the host name of the target computer.

Note: This method does not delete the target computer from the CA ITCM MDB. If you do not remove the computer from the network and do not disable PXE, the computer re-registers with OSIM during the boot process.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string osimTargetLabel

Specifies the computer label of the target computer.

Return Values

None

Remarks

None

GetUnnamedOSIMRegisteredComputers

The GetUnnamedOSIMRegisteredComputers method fetches the list of computers that are registered with OSIM from the network but not during discovered hardware. These computers show a MAC address and not the host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Return Values

dsm:ArrayOfOSIMTargetProperties

[Array of OSIM target properties](#) (see page 992), one for each OSIM registered computer. The array contains the following properties of the target computers:

- Name—Name of the Target. For unnamed OSIM targets, this parameters returns empty string.
- MAC address—MAC Address of the target
- Host name—Host Name of the target
- Network address—Network address of the target

- Created at—Creation date
- Boot status—Boot status of the target
- Name of attached boot server—Name of the boot server the target is associated with.

Remarks

None

GetUnmanagedOSIMRegisteredComputers

The GetUnmanagedOSIMRegisteredComputers fetches the list of computers that are in an unmanaged state.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Return Values

dsm:ArrayOfOSIMTargetProperties

[Array of the OSIM target properties](#) (see page 992), one for each unmanaged computer. The array contains the following properties of the target computers:

- Name—Name of the Target. For unnamed OSIM targets, this parameters returns the MAC address of the computer.
- MAC address—MAC Address of the target
- Host name—Host Name of the target
- Network address—Network address of the target
- Created at—Creation date
- Boot status—Boot status of the target
- Name of attached boot server—Name of the boot server the target is associated with.

Remarks

None

GetOSInstallationRequest

The GetOSInstallationRequest method is used to retrieve the installation request details for the target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

dsm:WS_OSIM_CONF_TYPE

Specifies the [type of installation request](#) (see page 111).

xsd:bool confTypeSupplied

Specifies if the configuration type is specified.

Return Values

dsm:ArrayOfOSIMProperties

Displays a list of [OSIM installation request attributes](#) (see page 992). The array contains the following properties of the target computers:

- Target name
- Target UUID
- Image name
- Image UUID
- Boot status
- Configuration state
- Date of last status change
- Activation time

Note: Activation time is not available for Planned requests.

The following details are available only for scheduled requests:

- Wake on LAN required
- Reboot required
- Wait for boot server required
- Wait for images

Remarks

None.

GetOSInstallationRequestByName

The GetOSInstallationRequestByName method is used to retrieve the installation request details for the target computer by using the host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:String OSIMTargetLabel

Specifies the computer label of the target computer.

dsm:WS_OSIM_CONF_TYPE

Specifies the [type of installation request](#) (see page 111).

xsd:bool confTypeSupplied

Specifies if the configuration type is specified.

Return Values

dsm:ArrayOfOSIMProperties

Displays the list of [OSIM installation request attributes](#) (see page 992). The array contains the following properties of the target computers:

- Target name
- Target UUID
- Image name
- Image UUID
- Boot status
- Configuration state
- Date of last status change
- Activation time

Note: Activation time is not available for Planned requests.

The following details are available only for scheduled requests:

- Wake on LAN required
- Reboot required
- Wait for boot server required
- Wait for images

Remarks

None.

GetInstallationRequestList

The GetInstallationRequestList method is used to request the list of installation requests for the target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

Return Values

dsm:ArrayOfOSIMTargetInstallationRequestsProperties

Displays the [installation requests for the target computer](#) (see page 993). The array contains the installation request properties, one for each Installation Request:

- Target name
- Target UUID
- Image name
- Image UUID
- Boot status
- Configuration state
- Date of last status change
- Activation time

Note: Activation time is not available for Planned requests.

The following details are available only for scheduled requests:

- Wake on LAN required
- Reboot required
- Wait for boot server required
- Wait for images

Remarks

None.

GetInstallationRequestListByName

The GetInstallationRequestListByName method is used to retrieve the list of installation requests for the target computer by using the host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:String OSIMTargetLabel

Specifies the computer label of the target computer.

Return Values

dsm:ArrayOfOSIMTargetInstallationRequestsProperties

Displays the [installation requests for the target computer](#) (see page 993). The array contains the installation request properties for the target computer:

- Target name
- Target UUID
- Image name
- Image UUID
- Boot status
- Configuration state
- Date of last status change
- Activation time

Note: Activation time is not available for Planned requests.

The following details are available only for scheduled requests:

- Wake on LAN required
- Reboot required
- Wait for boot server required
- Wait for images

Remarks

None

DeleteOSInstallationRequest

The DeleteOSInstallationRequest method is used to delete a planned or scheduled installation of an OSIM target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

dsm:WS_OSIM_IR_STATUS irStatus

Specifies the [installation request status of the OSIM computer](#) (see page 111).

Return Values

None

Remarks

None

DeleteOSInstallationRequestByName

The DeleteOSInstallationRequestByName method is used to delete a planned or scheduled installation of an OSIM target computer by using the host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:String OSIMTargetLabel

Specifies the computer label of the target computer.

dsm:WS_OSIM_IR_STATUS irStatus

Specifies the [installation request status of the OSIM computer](#) (see page 111).

Return Values

None

Remarks

None

SetOSIMComputerMACAddressByName

The SetOSIMComputerMACAddressByName method is used to change the MAC address information and the installation requests by using the host name of a target computer.

Note: You can apply this method only when there is no scheduled request. If there is a scheduled request, it must be of Failed or Stopped status .

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:String OSIMTargetLabel

Specifies the computer label of the target computer.

xsd:string MACAddress

Specifies the new MAC address of the target computer.

Return Values

None

Remarks

None

SetOSInstallationParameters

The SetOSInstallationParameters method updates multiple OS installation parameters for a particular target identified by MAC address.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string osimTargetMACAddress

Specifies the MAC address of the target whose OS installation parameters are to be updated.

ArrayOfOSIMParameters arrParams

Specifies an array of installation parameters to be updated.

SetOSInstallationParametersByName

The SetOSInstallationParametersByName method updates multiple OS installation parameters for a particular target identified by computer label.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string string osimTargetComputerLabel

Specifies the computer label of the target whose OS installation parameters are to be updated.

ArrayOfOSIMParameters arrParams

Specifies an array of installation parameters to be updated.

GetOSInstallationParameters

The GetOSInstallationParameters method is used to fetch the installation parameter settings with the descriptions of planned, scheduled, or current installation request for the target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

Return values

dsm:ArrayOfOSIMTargetPropertiesWithDesc

Array of the [installation parameters of the target computer](#) (see page 994).

Remarks

None

GetOSInstallationParametersByName

The GetOSInstallationParametersByName method is used to fetch the installation parameter settings with the descriptions of planned, scheduled, or current installation request by using the host name of the target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:String OSIMTargetLabel

Specifies the computer label of the target computer.

Return values

dsm:ArrayOfOSIMTargetPropertiesWithDesc

Array of the [installation parameters of the target computer](#) (see page 994).

Remarks

None

SetOSInstallationParameter

The SetOSInstallationParameter method changes the OS installation parameter of the target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the target computer.

xsd:string installParameterName

Specifies the parameter name of the target computer.

xsd:string installParameterValue

Specifies the new parameter value of the target computer.

Return Values

None

Remarks

None

SetOSInstallationParameterByName

The SetOSInstallationParameterByName method changes the OS installation parameter by the host name of the target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:String OSIMTargetComputerLabel

Specifies the computer label of the target computer.

xsd:string installParameterName

Specifies the parameter name of the target computer.

xsd:string installParameterValue

Specifies the new parameter value of the target computer.

Return Values

None

Remarks

None

OpenOSImageList

The OpenOSImageList method fetches the list of all available OS images.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:bool isBootServerNameSupplied

Specifies if the boot server name is entered.

xsd:string bootServerName

Specifies the name of the boot server.

dsm:ArrayOfOSImageFilter

Specifies the [filter on the OS image properties](#) (see page 995).

xsd:bool matchAll

Specifies the condition on the filter properties.

dsm:OSIMAGEPROPERTY sortProperty

Specifies the [OS image property](#) (see page 802) that you want to sort.

xsd:bool ascending

Specifies if the sorting is in ascending order.

Return Values

xsd:long OpenListHandle

Specifies the numeric handle to a list of OS images that can be passed to GetOSImages.

xsd:long totalNoOfImages

Specifies the total number of images in the list returned by the OpenListHandle parameter.

GetOSImages

TheGetOSImages method retrieves the properties of the OS images from OpenOSImageList.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:long listHandle

Specifies the numeric handle to a list as returned by a previous call to OpenOSImageList.

xsd:long noOfImages

Specifies the total number of OS images to fetch from the list. This must be less than or equal to the total number in the list as returned by a previous call to OpenOSImageList

Return Values

dsm:ArrayOfOSImageProperties

Displays the list of the [open OS images](#) (see page 995).

Remarks

None

CloseOSImageList

The CloseOSImageList method closes the list handle.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:long listHandle

Specifies the list handle for the list of OS images.

Return Values

None

Remarks

None

GetBootImageList

The GetBootImageList method fetches all the available boot images.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

xsd:bool isBootServerNameSupplied

Specifies if the boot server name is entered.

xsd:string bootServerName

Specifies the name of the boot server.

Return Values**dsm:ArrayOfOSImageProperties**

[Array of list of boot image properties](#) (see page 995). The array contains the following boot image properties:

- Image name—Name of the image
- Type indicator—Boot image type
- Platform—boot image platform
- Registered at—Date when the boot image was registered at the domain manager
- SD package name—Name of the related SD package
- SD package version—Version of the related SD package
- SD package comment—Comment related with the SD package
- Image support information—Information about the feature support of the image, for example, FIPS 140-2

AssignOSIMTargetsToBootServer

The AssignOSIMTargetsToBootServer method assigns a list of target computers using the MAC address to a boot server.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string bootServerName

Specifies the name of the boot server.

dsm:ArrayOfstring osimTargetMACAddress

Specifies the list of MAC addresses of target computers.

Return Values

None

Remarks

None

AssignOSIMTargetsToBootServerByName

The AssignOSIMTargetsToBootServerByName method assigns a list of target computers using the host names to a boot server.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string bootServerName

Specifies the name of the boot server.

dsm:ArrayOfstring osimTargetLabel

Specifies the list of computer labels of target computers.

Return Values

None

Remarks

None

OpenScalabilityServerList

The OpenScalabilityServerList fetches a list of all scalability servers registered with a domain manager.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ScalabilityServerPropertiesRequired

Specifies the [properties required to fetch a scalability server](#) (see page 800) from the list.

Return Values

xsd:long OpenListHandle

Specifies the numeric handle to a list of servers that is used by GetScalabilityServers.

xsd:long totalNoOfQueries

Specifies the total number of images in the list returned by the OpenListHandle parameter..

Remarks

This method returns the size of the list and a handle to the list that can be used with the [GetScalabilityServers method](#) (see page 473).

GetScalabilityServers

The GetScalabilityServers method retrieves the properties of the scalability servers from scalability server list.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the Web Service.

xsd:long listHandle

Specifies the numeric handle to a list, returned by a previous call to OpenScalabilityServerList.

xsd:long noOfServers(m)

Specifies the total number of servers to get from the list. This must be less than or equal to the total number in the list, returned by a previous call to OpenScalabilityServerList

Return Values

dsm:ArrayOfServerProperties GetServerReturn

Specifies an [Array of scalability server properties](#) (see page 995), one for each server.

Remarks

None

CloseScalabilityServerList

The CloseScalabilityServerList closes the list of open scalability servers.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:long listHandle

Specifies the total number of images in the list returned by the OpenListHandle parameter.

Return Values

None

Remarks

None

SearchComputerProperty

The SearchComputerProperty method checks if a computer with the given property value exists.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string propertyValue

Specifies the computer property value that you want to search for.

dsm:PROPERTY-TYPE propType

Identifies the [type of property](#) (see page 112).

Return Values

None

Remarks

None

GetOSImageParameters

The GetOSImageParameters method fetches the default parameter settings of an OS image.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

xsd:string osImage

Specifies the OS image name. The default parameter settings of the OS image specified will be retrieved.

Return Values**dsm:ArrayOfOSIMPropertiesWithDesc**

Array of the [default parameters of the OS image](#) (see page 994).

Remarks

None

GetBootServerList

The GetBootServerList method fetches the list of boot servers registered with the domain manager.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Return Values

dsm:ArrayOfBootServerProperties

Array of [boot server properties](#) (see page 995) registered with the domain manager. The array contains the following boot server properties:

- BootServer Name—Name of the Bootserver
- Access method—Access method configured
- Boot server status—Boot server status
- Last report—Timestamp last report
- Last report requested—Timestamp last report request
- Last response—Timestamp last heard
- MPC status—MPC status
- NFS support—NFS support
- PXE interfaces
- SD agent package (Linux)—SD agent package (Linux)
- SD agent package (Windows)—SD agent package (Windows)
- SD agent package r11 (Windows)—SD agent package r11 (Windows)

Remarks

None

GetOSImageDeployedBootServers

The GetOSImageDeployedBootServers fetches the list of the boot servers where the OS image is registered.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string osImage

Specifies the name of the OS image.

Return Values

dsm:ArrayOfBootServerProperties

Array of [boot server properties](#) (see page 995) registered with the domain manager. The array contains the following boot server properties:

- BootServer Name—Name of the Bootserver
- Access method—Access method configured
- Boot server status—Boot server status
- Last report—Timestamp last report
- Last report requested—Timestamp last report request
- Last response—Timestamp last heard
- MPC status—MPC status
- NFS support—NFS support
- PXE interfaces
- SD agent package (Linux)—SD agent package (Linux)
- SD agent package (Windows)—SD agent package (Windows)
- SD agent package r11 (Windows)—SD agent package r11 (Windows)

Remarks

None

GetBootImageDeployedBootServers

The GetBootImageDeployedBootServers method fetches the list of all the boot servers where the Boot image is registered.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string bootImage

Specifies the name of the boot image.

Return Values

dsm:ArrayOfBootServerProperties

Array of [boot server properties](#) (see page 995) registered with the domain manager. The array contains the following boot server properties:

- BootServer Name—Name of the Bootserver
- Access method—Access method configured
- Boot server status—Boot server status
- Last report—Timestamp last report
- Last report requested—Timestamp last report request
- Last response—Timestamp last heard
- MPC status—MPC status
- NFS support—NFS support
- PXE interfaces
- SD agent package (Linux)—SD agent package (Linux)
- SD agent package (Windows)—SD agent package (Windows)
- SD agent package r11 (Windows)—SD agent package r11 (Windows)

Remarks

None

DeRegisterOSImage

The DeRegisterOSImage method deregisters the OS image from the domain manager.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string osImage

Specifies the name of the OS image.

Return Values

None

Remarks

None

DeRegisterBootImage

The DeRegisterBootImage method deregisters the boot image from the domain manager.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string bootImage

Specifies the name of the boot image.

Return Values

None

Remarks

None

AssignOSImageToOSIMTargets

The AssignOSImageToOSIMTargets method assigns an OS image to target computers.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string osImage

Specifies the name of the OS image.

dsm:ArrayOfstring osimTargetNames

Specifies an array of the list of computer labels for target computers.

Return Values

None

Remarks

None

OpenOSIMTargetList

The OpenOSIMTargetList method lists the handle of the registered OSIM target computers.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:bool isBootServerNameSupplied

Specifies if the boot server name is entered.

xsd:string bootServerName

Specifies the name of the boot server.

dsm:ArrayOfOSImageFilter

Specifies the [filter on the OS image properties](#) (see page 995).

xsd:bool matchAll

Specifies the condition on the filter properties.

dsm:OSIMAGEPROPERTY sortProperty

Specifies the [OS Image property](#) (see page 802) that you want to sort.

xsd:bool ascending

Specifies sorting in an ascending order.

Return Values

None

Remarks

None

GetOSIMTargets

The GetOSIMTargets method fetches the list of OSIM target computers.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

xsd:long listHandle

Specifies the list handle of the list of OS images.

long totalNoOfOSIMTargets

Specifies the number Of OSIM target computers.

Return Values**dsm:ArrayOfTargetProperties arrayOfTargetProperties**

[Returns an array of target properties](#) (see page 995).

Remarks

None

CloseOSIMTargetList

The CloseOSIMTargetList method is used to close the OSIM target computers list handle.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:long listHandle

Specifies the list handle of the list of OS images.

Return Values

None

Remarks

None

GetOSIMTarget

The GetOSIMTarget method fetches the properties of an OSIM Computer based on OSIM ComputerMACAddress

Parameters

xsd:string sessionId (m)

Specifies the session identifier.

xsd:string OSIMTargetMACAddress

Specifies the MAC address of the OSIM computer.

Return Values

dsm:ArrayOfOSIMProperties arrayOfOSIMTargetProperties

Returns an array that includes the following properties for the target computer:

- Name—Name of the Target
- MAC address—MAC Address of the target
- Host name—Host Name of the target
- Network address—Network address of the target
- Created at—Creation date

- Boot status—Boot status of the target
- Name of attached boot server—Name of the boot server the target is associated with
- Name of OS image in the planned installation request—Name of the OS image involved with the planned installation request if there is any
- Name of OS image in the scheduled installation request—Name of the OS image involved with the scheduled installation request if there is any
- The status of the scheduled installation request—Status of the scheduled installation request if there is any scheduled installation

Remarks

None

GetOSIMTargetByName

The GetOSIMTargetByName method fetches the properties of OSIM Computer based on OSIM ComputerLabel.

Parameters

xsd:string sessionId (m)

Specifies the session identifier.

xsd:string OSIMTargetLabel

Specifies the label of the OSIM computer.

Return Values

dsm:ArrayOfOSIMProperties arrayOfOSIMTargetProperties

Returns an array that includes the following properties for the target computer:

- Name—Name of the Target
- MAC address—MAC Address of the target
- Host name—Host Name of the target
- Network address—Network address of the target
- Created at—Creation date

- Boot status—Boot status of the target
- Name of attached boot server—Name of the boot server the target is associated with
- Name of OS image in the planned installation request—Name of the OS image involved with the planned installation request if there is any
- Name of OS image in the scheduled installation request—Name of the OS image involved with the scheduled installation request if there is any
- The status of the scheduled installation request—Status of the scheduled installation request if there is any scheduled installation

Remarks

None

GetOSImages2

The `GetOSImages2` method returns a list of OS images from a specified index number. For example, if the complete list of OS images has 10 images (from 0-9) and if you want to read from the fifth position, this method can return a list with images from 5 to 9.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:long listHandle

Specifies the numeric handle to a list as returned by a previous call to `OpenOSImageList`.

xsd:long noOfImages

Specifies the number of objects you want to retrieve from the `startIndex` position. This number must be less than or equal to the total number in the list as returned by a previous call to `OpenOSImageList`.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

Return Values

dsm:ArrayOfOSImageProperties

Displays an array of OS image properties for the images that are returned from the result set.

Remarks

None

GetOSIMTargets2

The GetOSIMTargets2 method returns a list of OSIM registered computers from a specified index number. For example, if the complete list of OSIM registered computer has 10 computers (from 0-9) and if you want to read from the fifth position, this method can return a list with computers from 5 to 9.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:long listHandle

Specifies the list handle of the list of OS images.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

Xsd:long noOfTargets

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

dsm:ArrayOfTargetProperties

Array of OSIM target properties, one for each OSIM target computer.

Remarks

None

GetOSIMTargetList

The GetOSIMTargetList method returns the OSIM registered computers.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:bool isBootServerNameSupplied

Specifies whether the boot server name is entered.

xsd:string bootServerName

Specifies the name of the boot server.

dsm:ArrayOfOSIMTargetFilter2 filterArray

Specifies the filter on the OSIM computer properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

dsm:OSIMTARGETPROPERTY2 sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

Xsd:long noOfTargetsRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Xsd:boolean refreshList

Specifies whether you want to recompute the result set for each call.

Return Values

dsm:ArrayOfTargetProperties2 arrayOfOSIMTargets

Contains an array of OSIM target properties, one for each OSIM target computer.

xsd:long totalNumberOfTargets

Contains the total number of computers matching the filter condition.

Remarks:

Even though the interface contains parameters related to filtering, the method does not support filtering operations.

SetOSInstallationParameter2

The SetOSInstallationParameter2 method changes the OS installation parameters for an array of computers identified by MACAddress.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfstring osimTargetMACAddress

Specifies list of MAC addresses of target computers.

xsd:string installParameterName

Specifies the parameter name of the OS installation parameter you want to change.

xsd:string installParameterValue

Specifies the OS installation parameter value you want to change. To update the default value, specify NULL.

Return Values

None

Remarks

None

SetOSInstallationParameterByName2

The SetOSInstallationParameterByName2 method changes the OS installation parameter for an array of computers identified by the host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfstring osimTargetComputerLabel (m)

Specifies the list of computer labels of the target computers.

xsd:string installParameterName

Specifies the parameter name of the OS installation parameter you want to change.

xsd:string installParameterValue

Specifies the OS installation parameter value you want to change. To update the default value, specify NULL.

Return Values

None

Remarks

None

GetOSIMParameterDefaults

The GetOSIMParameterDefaults method returns the supported values for an OS Installation parameter for OSIM Computers identified by MACAddress. For example, some of the OSIM parameters like, Time Zone, Language, and so on, have a set of valid values list for OS Image.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfstring osimTargetMACAddress

Specifies list of MAC addresses of target computers.

xsd:string paramName

Specifies the name of the parameter for which you want to retrieve the valid values.

Return Values

dsm:ArrayOfstring defaults

Returns an array of valid values for the specified parameter.

GetOSIMParameterDefaultsByName

The GetOSIMParameterDefaultsByName method returns the supported values for an OS Installation parameter for OSIM Computers identified by the host name. For example, some of the OSIM parameters like, Time Zone, Language, and so on, have a set of valid values list for OS Image.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfstring osimTargetComputerLabel

Specifies the list of computer labels of target computers.

xsd:string paramName

Specifies the name of the parameter for which you want to retrieve the valid values.

Return Values

dsm:ArrayOfstring defaults

Returns an array of valid values for the specified parameter.

GetBootImageList2

The GetBootImageList2 method fetches all the registered boot images in a domain manager. This method also supports the sort and filter operations.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:bool isBootServerNameSupplied

Specifies if the boot server name is entered.

xsd:string bootServerName

Specifies the name of the boot server.

dsm: ArrayOfBootImageFilter filterArray

Specifies an array of filter conditions on boot image properties.

dsm:BootImageProperty sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

xsd:long numRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Xsd:boolean refreshList

Specifies whether you want to recompute the result set for each call.

Return Values

xsd:long totalNumImages

Contains the total number of images available for the specified filter condition.

dsm:ArrayOfBootImageProperties biPropsArray

Contains an array of boot image properties.

GetBootServerList2

The GetBootServerList2 method returns the boot servers in the domain manager. This method supports the sort and filter operations also.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfBootServerFilter filterArray

Specifies an array of filter conditions on boot server properties.

dsm:BootServerProperty sortProperty

Specifies the boot server property based on which you want to sort the servers.

xsd:boolean ascending

Specifies whether you want to sort the boot servers in ascending order. If set to True, the servers are sorted in ascending order based on the sort property you specified.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

xsd:long numRequired

Specifies the number of boot servers to get from the specified index number.

Xsd:boolean refreshList

Specifies whether you want to recompute the result set for each call.

Return Values**xsd:long totalNumServers**

Contains the total number of boot server available for the specified filter condition.

dsm: ArrayOfBootServerProperties biPropsArray

Contains an array of boot server properties.

Software Groups

This section contains the following topics:

[CreateSoftwareGroup](#) (see page 491)

[CreateSoftwareGroupByName](#) (see page 492)

[DeleteSoftwareGroups](#) (see page 493)

[DeleteSoftwareGroupsByName](#) (see page 493)

[LinkSoftwareGroupsToSoftwarePackageGroup](#) (see page 494)

[LinkSoftwareGroupsToSoftwarePackageGroupByName](#) (see page 495)

[SetSoftwareGroup](#) (see page 496)

[SetSoftwareGroupByName](#) (see page 496)

[UnlinkSoftwareGroupsFromSoftwarePackageGroup](#) (see page 497)

[UnlinkSoftwareGroupsFromSoftwarePackageGroupByName](#) (see page 498)

CreateSoftwareGroup

Creates software package group or software package procedure group based on the given group type.

Parameters**xsd:string sessionId (m)**

Session Id obtained from the login to the web service.

xsd:string parentSoftwarePackageGroupId

Parent software group Id under which new group is created By default, the group is created under software Package Library.

CreateSoftwareGroupProperties softwareGroupProperties (m)

The properties required to create software group. Setting all the structure members is not mandatory.

Return Values

xsd:string softwareGroupId

Returns the software group UUID after creating the software group.

Remarks

None

More information:

[CreateSoftwareGroupProperties](#) (see page 643)

CreateSoftwareGroupByName

The CreateSoftwareGroupByName method creates a software package group or Software Package Procedure Group based on the group type.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string parentSoftwarePackageGroupName

Specifies the name of parent software group under which the new group is created. When not specified the group is created under the root software group.

CreateSoftwareGroupProperties createSoftwareGroupProperties (m).

The properties required to create software group. Setting all the structure members is not mandatory

Return Values

xsd:string softwareGroupId

Returns the software group UUID after creating the software group.

Remarks

None

More information:

[CreateSoftwareGroupProperties](#) (see page 643)

DeleteSoftwareGroups

The DeleteSoftwareGroups method deletes the group identified by its UUID. Software group may be software package group or software package procedure group. If recursive is set to true, the child software groups are recursively deleted.

Note: Recursive deletion of software package procedure groups is not needed because this api does not allow the creation of child software package procedure groups.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwareGroupId (m)

UUID of software group to delete.

xsd:boolean isRecursive

If recursive is true, it recursively deletes the software group. The default value is false.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteSoftwareGroupsByName

The DeleteSoftwareGroupsByName method deletes the group identified by its Name. Software group may be software package group or software package procedure group. If recursive is set to true, the child software groups are recursively deleted.

Note: Recursive deletion of software package procedure groups is not needed because this api does not allow the creation of child software package procedure groups.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwareGroupName (m)

Name of the software group to delete.

xsd:boolean isRecursive

If recursive is true, it recursively deletes the software group. The default value is false.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkSoftwareGroupsToSoftwarePackageGroup

The LinkSoftwareGroupsToSoftwarePackageGroup method links one or more software groups to a software package group. Software group can be software package group or software package procedure group

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwarePackageGroupId

Specifies the UUID of the software package group to which the software procedure groups are to be linked.

If this UUID is not supplied the software procedure groups and software package group link to the root location.

ArrayOfstring softwareGroupIds (m)

List of software package group UUIDs and/or software package procedure group UUIDs.

Return Values

This is a void method that does not expose a return value.

Remarks

The same functionality is already achieved in the current implementation of webservises by using `LinkSoftwarePackageProcedureGroupsToSoftwarePackageGroup` and `LinkSoftwarePackageGroupsToSoftwarePackageGroup`.

LinkSoftwareGroupsToSoftwarePackageGroupByName

This method links one or more software package procedures from a software package procedure group Parameters.

Parameters**xsd:string sessionId (m)**

Specifies the session id obtained from the login to the web service.

xsd:string softwarePackageProcedureGroupName (m)

Package procedure group name to which software package procedures are to be linked.

ArrayOfSoftwarePackageProcedure softwarePackageProcedures (m)

Set of software package procedures to be linked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetSoftwareGroup

Sets the properties of the software group by giving the softwareGroupld (package Group/ ProcedureGroup)

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwareGroupld (m)

Specifies the UUID of the software group whose properties need to be set.

SetSoftwareGroupProperties setSoftwareGroupProperties (m)

Specifies the values of the software group properties that need to be changed. Specifying values for all the members of the sequence *SetSoftwareGroupProperties* is not mandatory.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareGroupProperties](#) (see page 734)

SetSoftwareGroupName

Sets the properties of the software group. The target software group (package Group or procedure group) is specified by name

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string softwareGroupName (m)

Specifies the name of the software group whose properties need to be set.

SetSoftwareGroupProperties setSoftwareGroupProperties (m)

Specifies the values of the software group properties that need to be changed. Specifying values for all the members of the sequence *SetSoftwareGroupProperties* is not mandatory.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetSoftwareGroupProperties](#) (see page 734)

UnlinkSoftwareGroupsFromSoftwarePackageGroup

The UnlinkSoftwareGroupsFromSoftwarePackageGroup method unlinks one or more software groups from the specified software package group. Software groups can be software package groups and software package procedure groups. If the software package group is not given it unlinks from the software Package library.

Parameters**xsd:string sessionId (m)**

Specifies the session id obtained from the login to the web service.

xsd: string softwarePackageGroupId

If the software package group UUID is not specified, the system attempts to unlink software procedure groups and software package group from the root location. When specified, the software procedure groups and software package group is unlinked from the specified software package group.

ArrayOfstring softwareGroupIds (m)

List of UUIDs of subgroups to be unlinked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkSoftwareGroupsFromSoftwarePackageGroupByName

This method unlinks the software groups (software package group and software package procedure groups) from a software package group. If the software package group is not given the method unlinks from the software Package library.

Parameters

xsd:string sessionId (m)

Session Id. got from the login to the web service.

xsd:string softwarePackageGroupName

If the software package group name is not specified then the system will attempt to unlink software procedure groups and software package group from the root location. If specified then the software procedure groups and software package group would be unlinked from the specified software package group.

ArrayOfstring softwareGroupNames (m)

List of subgroup names to be unlinked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

Units and Groups

The following sections describe unit and group methods.

This section contains the following topics:

[Unit Groups](#) (see page 499)

[Unit Inventory](#) (see page 534)

[Units](#) (see page 543)

[User Profiles](#) (see page 571)

Unit Groups

This section describes unit group methods.

This section contains the following topics:

[CloseUnitGroupComputerMembersList](#) (see page 500)
[CloseUnitGroupUnitGroupMembersList](#) (see page 500)
[CloseUnitGroupUnitMembersList](#) (see page 501)
[CloseUnitGroupUserAccountMembersList](#) (see page 501)
[CloseUnitGroupUserProfileMembersList](#) (see page 502)
[CreateUnitGroup](#) (see page 502)
[CreateUnitGroup2](#) (see page 503)
[CreateUnitGroupByName](#) (see page 504)
[DeleteUnitGroups](#) (see page 506)
[DeleteUnitGroupsByName](#) (see page 506)
[GetSystemGroupUUID](#) (see page 507)
[GetUnitGroup](#) (see page 507)
[GetUnitGroupComputerMembers](#) (see page 508)
[GetUnitGroupUnitGroupMembers](#) (see page 508)
[GetUnitGroupUnitMembers](#) (see page 509)
[GetUnitGroupUserAccountMembers](#) (see page 510)
[GetUnitGroupUserProfileMembers](#) (see page 511)
[LinkComputersToUnitGroup](#) (see page 511)
[LinkComputersToUnitGroupByName](#) (see page 512)
[LinkUnitGroupsToUnitGroup](#) (see page 513)
[LinkUnitGroupsToUnitGroupByName](#) (see page 513)
[LinkUserAccountsToUnitGroup](#) (see page 514)
[LinkUserProfilesToUnitGroup](#) (see page 515)
[OpenUnitGroupComputerMembersList](#) (see page 515)
[OpenUnitGroupUnitGroupMembersList](#) (see page 516)
[OpenUnitGroupUnitMembersList](#) (see page 517)
[OpenUnitGroupUserAccountMembersList](#) (see page 518)
[OpenUnitGroupUserProfileMembersList](#) (see page 519)
[SetUnitGroup](#) (see page 520)
[SetUnitGroup2](#) (see page 521)
[EvaluateUnitGroupQuery](#) (see page 521)
[UnlinkComputersFromUnitGroup](#) (see page 522)
[UnlinkComputersFromUnitGroupByName](#) (see page 522)
[UnlinkUnitGroupsFromUnitGroup](#) (see page 523)
[UnlinkUnitGroupsFromUnitGroupByName](#) (see page 523)
[UnlinkUserAccountsFromUnitGroup](#) (see page 524)
[UnlinkUserProfilesFromUnitGroup](#) (see page 525)
[DeleteUnitGroups2](#) (see page 525)
[DeleteUnitGroupsByName2](#) (see page 526)
[CreateUnitGroup3](#) (see page 527)
[GetUnitGroup2](#) (see page 528)
[LinkItemsToUnitGroup](#) (see page 528)
[UnLinkItemsFromUnitGroup](#) (see page 529)
[DeleteItems](#) (see page 529)
[MoveItemsToUnitGroup](#) (see page 530)
[GetUnitGroupComputerMembers2](#) (see page 530)
[OpenUnitGroupComputerMembersList2](#) (see page 531)

[GetUnitGroupComputerMembers3](#) (see page 532)

[GetUnitGroupByName](#) (see page 532)

[GetUnitGroupsXML](#) (see page 533)

CloseUnitGroupComputerMembersList

The CloseUnitGroupComputerMembersList method closes a unit group list that was opened by OpenUnitGroupComputerMembersList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitGroupComputerMembersList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitGroupUnitGroupMembersList

The CloseUnitGroupUnitGroupMembersList method closes a unit group list that was opened by OpenUnitGroupUnitGroupMembersList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitGroupUnitGroupMembersList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitGroupUnitMembersList

The CloseUnitGroupUnitMembersList method closes a unit group list that was opened by OpenUnitGroupUnitMembersList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitGroupUnitMembersList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitGroupUserAccountMembersList

The CloseUnitGroupUserAccountMembersList method closes a unit group list that was opened by OpenUnitGroupUserAccountMembersList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitGroupUserAccountMembersList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitGroupUserProfileMembersList

The CloseUnitGroupUserProfileMembersList method closes a unit group list that was opened by OpenUnitGroupUserProfileMembersList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitGroupUserProfileMembersList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateUnitGroup

The CreateUnitGroup method creates a unit group

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string parentUnitGroupId (")

The UUID identifying the parent group of which the newly created group is a child. In order to create a top-level group, this is the UUID of the Computer And Users group.

CreateUnitGroupProperties createUnitGroupProperties (")

The properties required to create a unit group.

Return Values

xsd:string CreateUnitGroupReturn

If a unit group was successfully created, the UUID of the new group will be returned.

Remarks

If the parentUnitGroupId is supplied, the method will create a sub-group of the specified parent group.

More information:

[CreateUnitGroupProperties](#) (see page 783)

CreateUnitGroup2

Creates a unit group. If you do not specify the the parent group, the new unit group is created under the group *Computers and Users*. You can link existing computers to the newly created unit group by specifying their UUIDs, or you can link all discovered computers to the unit group. You can attach a query, specified by its uuid, to the newly created group for evaluating the group members.

Note: When the parameter `computerUUIDS` is specified and `linkAllComputers` is set to true, the parameter `linkAllComputers` takes precedence, which means that `computerIds` are ignored and all computers discovered under the domain or enterprise are linked to the newly created group. When the query to be linked to the group is specified (to create a *dynamic group*) and the computers to be associated with the group are also specified (to create a *static group*), the API returns an error.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string parentUnitGroupId

Specifies the UUID of the parent group for the unit group to be created. By default, the “Computer and User” group is set as the parent group.

CreateUnitGroupProperties2 createUnitGroupProperties (m)

Specifies the properties required to create a unit group. Setting all the structure members is not mandatory.

ArrayOfstring computerUUIDS

Specifies the list of computer uuids to be linked to the group.

xsd:boolean linkAllComputers

Links all registered computers in the domain or enterprise to the group. If set to true the value of parameter `computerUUIDs` is ignored.

Return Values

xsd:string uuidOfCreatedUnitGroup

Indicates the UUID of the new unit group, if created successfully.

Remarks

If parameter `computerUUIDS` is assigned a value (that is, the list of computer uuids is provided and the parameter `linkAllComputers` is set to true) the value of `linkAllComputers` overrides the value of parameter `computerUUIDS`

If the query to be linked to the group (which creates a *dynamic group*) and the computers to be associated with the group (which creates a *static group*) are both specified at the same time the API returns an error.

More information:

[Structure CreateUnitGroupProperties2](#) [CreateUnitGroupProperties2](#) (see page 746)

CreateUnitGroupByName

Creates a unit group. You can specify the parent group by its name; if it is not specified the new unit group is created under the group *Computers and Users*. You can link all discovered computers or existing computers specified by their respective names to the newly created unit group. You can attach a query, specified by its name, to the newly created group for evaluating the group members.

Note: When parameter `computerNames` is specified and `linkAllComputers` is set to 1 (true), the parameter `linkAllComputers` takes precedence; this means that `computerNames` are ignored and all computers discovered under the domain/enterprise are linked to the newly created group. When query to be linked to the group is specified (to create a *dynamic group*) and the computers to be associated with the group are also specified (to create a *static group*), the API returns an error.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string parentUnitGroupName

Specifies the name of the parent group under which the unit group is to be created. By default the “Computer and User” group is set as the parent group.

CreateUnitGroupPropertiesByName unitGroupPropertiesByName (m)

Specifies the properties required to create a unit group. Setting all the structure members is not mandatory.

ArrayOfstring computerNames

Specifies the list of computer names to be linked to the group.

xsd:boolean linkAllComputers

Links all registered computers in the domain or enterprise to the group. If set to true, the value of parameter computerNames is ignored.

Return Values

xsd:string uuidOfCreatedUnitGroup

Indicates the UUID of the new unit group, if successfully created.

Remarks

If parameter computerNames is assigned a value (that is, the list of computer names is provided) and the parameter linkAllComputers is also set to true the value of linkAllComputers overrides the value of parameter computerNames.

If the query to be linked to the group (which creates a *dynamic group*) and the computers to be associated with the group (which creates a *static group*) are specified at the same time, the API returns an error.

More information:

[CreateUnitGroupPropertiesByName](#) (see page 695)

DeleteUnitGroups

The DeleteUnitGroups method deletes one or more groups identified by their UUIDs.

Parameters

xsd:string sessionId (^m)

The session identifier.

ArrayOfstring unitGroupIds (^m)

An array of UUIDs that identify the groups to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteUnitGroupsByName

The DeleteUnitGroupsByName method deletes one or more unit groups identified by their Names.

Note: When the web service is connected to a domain manager, and the user tries to delete a group replicated from the enterprise manager, an error is returned to the user. On a domain manager only local computer groups can be deleted.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring unitGroupNames (m)

Specifies the array of names that identify the unit groups to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

GetSystemGroupUUID

The GetSystemGroupUUID method retrieves the identifying UUID for the specified CA ITCM system group.

Parameters

xsd:string sessionId (^m)

The session identifier.

SystemGroup systemGroup (^m)

The identifier for the system group whose UUID is to be retrieved.

Return Values

xsd:string systemGroupUUID

If the system group was successfully found, the UUID of the group will be returned.

Remarks

None

GetUnitGroup

The GetUnitGroup method retrieves a unit group identified by its UUID.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string unitGroupId (^m)

The UUID of the unit group to retrieve.

UnitGroupPropertiesRequired unitGroupPropertiesRequired (^m)

This parameter specifies which unit group properties to retrieve.

Return Values

UnitGroupProperties GetUnitGroupReturn

The unit group properties retrieved.

Remarks

None

More information:

[UnitGroupPropertiesRequired](#) (see page 876)

GetUnitGroupComputerMembers

The GetUnitGroupComputerMembers method retrieves the computers from a unit group's member list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list returned by a previous call to OpenUnitGroupComputerMembersList.

xsd:long noOfComputers ("")

The total number of computers to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenUnitGroupComputerMembersList.

Return Values

ArrayOfComputerProperties GetUnitGroupComputerMembersReturn

An array of ComputerProperties - one for each computer requested from the list.

Remarks

You can make multiple calls to GetUnitGroupComputerMembers to iterate down the list.

More information:

[ComputerProperties](#) (see page 669)

GetUnitGroupUnitGroupMembers

The GetUnitGroupUnitGroupMembers method retrieves the unit groups from a unit group's member list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list returned by a previous call to OpenUnitGroupUnitGroupMembersList.

xsd:long noOfUnitGroups (")

The total number of groups to get from the list. This is less than or equal to the total number in the list returned by a previous call to `OpenUnitGroupUnitGroupMembersList`.

Return Values**ArrayOfUnitGroupProperties GetUnitGroupUnitGroupMembersReturn**

An array of `UnitGroupProperties` - one for each group requested from the list.

Remarks

You can make multiple calls to `GetUnitGroupUnitGroupMembers` to iterate down the list.

More information:

[UnitGroupProperties](#) (see page 870)

GetUnitGroupUnitMembers

The `GetUnitGroupUnitMembers` method retrieves the units (computers, user accounts, and user profiles) from a unit group's member list.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:long listHandle (")

A numeric handle to a list returned by a previous call to `OpenUnitGroupUnitMembersList`.

xsd:long noOfUnits (")

The total number of units to get from the list. This is less than or equal to the total number in the list returned by a previous call to `OpenUnitGroupUnitMembersList`.

Return Values**ArrayOfUnitProperties GetUnitGroupUnitMembersReturn**

An array of `UnitProperties` - one for each unit requested from the list.

Remarks

You can make multiple calls to `GetUnitGroupUnitMembers` to iterate down the list.

More information:

[UnitProperties](#) (see page 877)

GetUnitGroupUserAccountMembers

The GetUnitGroupUserAccountMembers method retrieves the user accounts (computers, user accounts and user profile) from a unit group's member list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list returned by a previous call to OpenUnitGroupUserAccountMembersList.

xsd:long noOfUserAccounts (")

The total number of user accounts to get from the list. This is than or equal to the total number in the list returned by a previous call to OpenUnitGroupUserAccountMembersList.

Return Values

ArrayOfUserAccountProperties

GetUnitGroupUserAccountMembersReturn

An array of UserAccountProperties - one for each user account requested from the list.

Remarks

You can make multiple calls to GetUnitGroupUserAccountMembers to iterate down the list.

More information:

[UserAccountProperties](#) (see page 890)

GetUnitGroupUserProfileMembers

The GetUnitGroupUserProfileMembers method retrieves the user profiles from a unit group's member list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list returned by a previous call to OpenUnitGroupUserProfileMembersList.

xsd:long noOfUserProfiles ("")

The total number of user profiles to get from the list. This is than or equal to the total number in the list returned by a previous call to OpenUnitGroupUserProfileMembersList.

Return Values

ArrayOfUserProfileProperties GetUnitGroupUserProfileMembersReturn

An array of UserProfileProperties - one for each user profile requested from the list

Remarks

You can make multiple calls to GetUnitGroupUserProfileMembers to iterate down the list.

More information:

[UserProfileProperties](#) (see page 892)

LinkComputersToUnitGroup

The LinkComputersToUnitGroup method links one or more computers to a given unit group

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string unitGroupId ("")

The UUID identifying the unit group to which the given computers is to be linked.

ArrayOfstring computerIds (m)

An array of UUIDs identifying the computers to link to the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkComputersToUnitGroupName

The LinkComputersToUnitGroupName API is used for linking array of computers identified by hostnames to a computer group.

Note: When the web service is connected to a domain manager, if you use this api to link a computer to a group replicated from the enterprise manager, an error is returned. On a domain manager the composition of only local groups (groups created in the domain manager) can be altered.

Parameters

xsd: string sessionId (m)

Specifies the session id obtained from the login to the web service.

dsm: unitGroupPropertiesByName (m)

Specifies the unit group name to which the given computers need to be linked. It contains two fields: (a) unitGroupName - Used for specifying the name of the computer group to link (b) ObjectPreference - Used for specifying the preference. If set to LOCALLY, this parameter searches for groups that are created locally. If set to REPLICATE, searches for replicated groups. This is applicable when DM is linked with EM.

arrayOfComputerPropertiesByName (m)

An array of computer hostnames that need to be linked to a specified unit group. It contains three fields: (a) computerName - hostname of the computer to be linked to group (b) ObjectPreference (c) domainName - This field is applicable when ObjectPreference is set to REPLICATE.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkUnitGroupsToUnitGroup

The LinkUnitGroupsToUnitGroup method links one or more unit groups to a given unit group

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string unitGroupId (")

The UUID identifying the unit group to which the given groups are to be linked.

ArrayOfstring unitGroupIds (")

An array of UUIDs identifying the groups that are to be linked to the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkUnitGroupsToUnitGroupByName

The LinkUnitGroupsToUnitGroupByName method links one or more unit groups specified by name to a given unit group also specified by name. The default value of the target group is *Computers and Users*.

Note: When the web service is connected to a domain manager, and the user tries to link a local group to a group replicated from the enterprise manager, or vice versa, an error is returned to the user. On a domain manager, the composition or properties of only local computer groups can be modified.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupName

Specifies the name of group to which the unit groups are to be linked. The default value is *Computer and User* group.

ArrayOfstring unitGroupNames (m)

Defines an array of names identifying the groups that are to be linked to the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkUserAccountsToUnitGroup

The LinkUserAccountsToUnitGroup method links one or more user accounts to a given unit group

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string unitGroupId ("")

The UUID identifying the unit group to which the given user accounts is to be linked.

ArrayOfstring userAccountIds ("")

An array of UUIDs identifying the user accounts that are to be linked to the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

LinkUserProfilesToUnitGroup

The LinkUserProfilesToUnitGroup method links one or more user profiles to a given unit group

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string unitGroupId (^m)

The UUID identifying the unit group to which the given user profiles are to be linked.

ArrayOfstring userProfileIds (^m)

An array of UUIDs identifying the user profiles that are to be linked to the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

OpenUnitGroupComputerMembersList

The OpenUnitGroupComputerMembersList method opens a list of the computers that are members of the given unit group.

The method returns the total number of computers in the list, and a handle to the list that can be used for further calls to GetUnitGroupComputerMembers.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string unitGroupId (^m)

The UUID of the unit group whose member computers are to be listed.

ArrayOfComputerPropertyFilter computerPropertyFilters

An array of computer property filter definitions to limit the computers returned by the search.

ComputerPropertiesRequired computerPropertiesRequired (")

The properties that will be required when getting a computer from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of unit groups that can be passed to GetUnitGroupComputerMembers

xsd:long totalNoOfComputers

The total number of computers in the list whose handle has been returned.

Remarks

None

More information:

[ComputerPropertiesRequired](#) (see page 670)

[ComputerPropertyFilter](#) (see page 671)

OpenUnitGroupUnitGroupMembersList

The OpenUnitGroupUnitGroupMembersList method opens a list of the unit groups that are members of the given unit group.

The method returns the total number of groups in the list, and a handle to the list that can be used for further calls to GetUnitGroupUnitGroupMembers

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string unitGroupId (")

The UUID of the unit group whose member groups are to be listed.

UnitGroupPropertiesRequired unitGroupPropertiesRequired (")

The properties that will be required when getting a group from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of unit groups that can be passed to GetUnitGroupUnitGroupMembers

xsd:long totalNoOfUnitGroups

The total number of groups in the list whose handle has been returned.

Remarks

None

More information:

[UnitGroupPropertiesRequired](#) (see page 876)

OpenUnitGroupUnitMembersList

The OpenUnitGroupUnitMembersList method opens a list of the units (user accounts, user profiles, and computers) that are members of the given unit group.

The method returns the total number of units in the list, and a handle to the list that can be used for further calls to GetUnitGroupUnitMembers

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string unitGroupId (")

The UUID of the unit group whose member units are to be listed.

ArrayOfUnitPropertyFilter unitPropertyFilters

An array of unit property filter definitions to limit the units returned by the search.

UnitPropertiesRequired unitPropertiesRequired (")

The properties that will be required when getting a unit from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of units that can be passed to GetUnitGroupUnitMembers

xsd:long totalNoOfUnits

The total number of units in the list whose handle has been returned.

Remarks

None

More information:

[UnitPropertiesRequired](#) (see page 877)

[UnitPropertyFilter](#) (see page 877)

OpenUnitGroupUserAccountMembersList

The OpenUnitGroupUserAccountMembersList method opens a list of the user accounts that are members of the given unit group.

The method returns the total number of user accounts in the list, and a handle to the list that can be used for further calls to GetUnitGroupUserAccountMembers

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string unitGroupId (")

The UUID of the unit group whose member user accounts are to be listed.

ArrayOfUserAccountPropertyFilter userAccountPropertyFilters (")

An array of user account property filter definitions to limit the user accounts returned by the search.

UserAccountPropertiesRequired userAccountPropertiesRequired (")

The properties that will be required when getting a user account from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of unit groups that can be passed to GetUnitGroupUserAccountMembers

xsd:long totalNoOfUserAccounts

The total number of user accounts in the list whose handle has been returned.

Remarks

None

More information:

[UserAccountPropertiesRequired](#) (see page 891)

[UserAccountPropertyFilter](#) (see page 891)

OpenUnitGroupUserProfileMembersList

The OpenUnitGroupUserProfileMembersList method opens a list of the user profiles that are members of the given unit group.

The method returns the total number of user profiles in the list, and a handle to the list that can be used for further calls to GetUnitGroupUserProfileMembers

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string unitGroupId (")

The UUID of the unit group whose member user profiles are to be listed.

ArrayOfUserProfilePropertyFilter userProfilePropertyFilters

An array of user profile property filter definitions to limit the user profiles returned by the search.

UserProfilePropertiesRequired userProfilePropertiesRequired (")

The properties that will be required when getting a user profile from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of unit groups that can be passed to GetUnitGroupUserProfileMembers

xsd:long totalNoOfUserProfiles

The total number of user profiles in the list whose handle has been returned.

Remarks

None

More information:

[UserProfilePropertiesRequired](#) (see page 896)

[UserProfilePropertyFilter](#) (see page 897)

More information:

[SetUnitGroupProperties](#) (see page 767)

SetUnitGroup

The SetUnitGroup method updates the properties of a unit group

Parameters

xsd:string sessionId (")

The session identifier.

SetUnitGroupProperties unitGroupProperties (")

The properties that are updated for the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetUnitGroup2

The SetUnitGroup2 method updates the properties of a unit group

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string unitGroupUUID (m)

UUID of unit group whose properties need to be modified.

SetUnitGroupProperties2 unitGroupProperties (m)

The properties that are updated for the specified unit group.

Return Values

This is a void method that does not expose a return value.

More information:

[SetUnitGroupProperties2](#) (see page 874)

EvaluateUnitGroupQuery

The EvaluateUnitGroupQuery method re-evaluates the members of a unit group based on the query that defines the members of that group.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string unitGroupId (m)

The UUID identifying the unit group whose associated query should be re-evaluated.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkComputersFromUnitGroup

The UnlinkComputersFromUnitGroup method unlinks one or more computers from the specified unit group

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string unitGroupId (m)

The UUID identifying the group from which the computers are to be unlinked.

ArrayOfstring computerIds (m)

An array of UUIDs identifying the computers that are to be unlinked from the specified group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkComputersFromUnitGroupName

This method unlinks a list of computers, specified by computer name, from a unit group. A computer can be uniquely identified by its name.

Note: When the web service is connected to a domain manager, and using this api, the user tries to unlink a computer from a group replicated from the enterprise manager, an error is returned to the user. On a domain manager the composition of only local groups (groups created in the domain manager) can be changed.

Parameters

xsd: string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd: string unitGroupName (m)

The Unit Group Name where computers need to be unlinked,

ArrayOfstring computerNames (m)

List of all computer names that need to be unlinked from UnitGroup.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkUnitGroupsFromUnitGroup

The `UnlinkUnitGroupsFromUnitGroup` method unlinks one or more unit groups from the specified unit group

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string unitGroupId ("")

The UUID identifying the unit group from which the given groups are to be unlinked.

ArrayOfstring unitGroupIds ("")

An array of UUIDs identifying the groups that are to be unlinked from the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkUnitGroupsFromUnitGroupName

The `UnlinkUnitGroupsFromUnitGroupName` method unlinks one or more unit groups specified by name from the specified unit group, also specified by name. The default value of the target group is *Computers and Users*.

Note: When the web service is connected to a domain manager, and the user tries to unlink from a group replicated from the enterprise manager, the action is not allowed and an error is returned. On a domain manager the composition of only local computer groups can be altered.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupName

Identifies the name of the unit group from which the given groups are to be unlinked. The default value is *Computer and User* group.

ArrayOfstring unitGroupNames (m)

Defines an array of names identifying the groups that are to be unlinked from the specified unit group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkUserAccountsFromUnitGroup

The UnlinkUserAccountsFromUnitGroup method unlinks one or more user accounts from the specified unit group

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string unitGroupId (m)

The UUID identifying the group from which the user accounts are to be unlinked.

ArrayOfstring userAccountIds (m)

An array of UUIDs identifying the user accounts that are to be unlinked from the specified group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UnlinkUserProfilesFromUnitGroup

The UnlinkUserProfilesFromUnitGroup method unlinks one or more user profiles from the specified unit group

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string unitGroupId (")

The UUID identifying the group from which the user profiles are to be unlinked.

ArrayOfstring userProfileIds (")

An array of UUIDs identifying the user profiles that are to be unlinked from the specified group.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteUnitGroups2

Parameters

xsd:string sessionId (m)

Specifies the session Id obtained from the login to the web service.

dsm:ArrayOfstring unitGroupIds

An array of UUIDs that identify the groups to delete.

xsd:boolean forceDelete

Deletes all the unit groups. If set to true, the relationship with the unit groups is ignored.

If the value is set to false, the API returns the objects that are related to the unit groups and deletes the groups that are not related to any objects.

dsm:LinkedObjectTypeRequired objectType

Specifies the object type. If forceDelete is set to true, set the ObjectType as CONSTRAINT_NONE. Else, set it to CONSTRAINT_ALL.

Return Values

None

More information:

[LinkedObjectTypeRequired](#) (see page 76)

DeleteUnitGroupsByName2

Parameters

xsd:string sessionId (m)

Specifies the session Id obtained from the login to the web service.

dsm:ArrayOfstring unitGroupNames

An array of names that identify the groups to delete.

xsd:boolean forceDelete

Deletes all the unit groups. If set to true, the relationship with the unit groups is ignored.

If the value is set to false, the API returns the objects that are related to the unit groups and deletes the groups that are related to any objects.

dsm:LinkedObjectTypeRequired objectType

Specifies the object type. If forceDelete is set to true, set the ObjectType as CONSTRAINT_NONE. Else, set it to CONSTRAINT_ALL.

Return Values

None

More information:

[LinkedObjectTypeRequired](#) (see page 76)

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateUnitGroup3

This method creates a unit group. If you do not specify the parent group, the new unit group is created under the group Computers and Users. You can link existing items (Groups/Computers/user accounts/user profiles) to the newly created unit group by specifying their UUIDs and type.

Parameters

xsd:string sessionId

Session Id comes from the login to the web service.

xsd:string parentUnitGroupId

Specifies the UUID of the parent group for which you can create a unit group. By default, the "Computer and User" group is set as the parent group.

dsm:CreateUnitGroupProperties3 groupProperties

Specifies the properties required to create a unit group.

dsm:ArrayOfItems linkItems

Specify the items to be linked to newly created group. You can link items to static groups only.

Return Values

xsd:string groupUuid

If a unit groups is successfully created, the UUID of the new group is returned.

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

GetUnitGroup2

This method retrieves a unit group properties identified by its UUID.

Parameters

xsd:string sessionId

Session Id is from the login to the web service.

xsd:string unitGroupId

The uuid of the group for which the properties must be retrieved.

Return Values

dsm:UnitGroupProperties2 groupProperties

The properties of the unit group.

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

LinkItemsToUnitGroup

This method links heterogeneous items (groups/computers/user accounts/user profiles) to a unit group.

Note:you can link items to only static group.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

xsd:string unitGroupId

The uuid of the group to which the items should be linked.

dsm:ArrayOfItems linkItems

Array of items to be linked to unit group

Return Values

None.

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

UnLinkItemsFromUnitGroup

This method unlinks heterogeneous items (groups/computers/user accounts/user profiles) from a unit group. Note that, you can unlink items to only static group.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

xsd:string unitGroupId

The uuid of the group to which the items should be linked.

dsm:ArrayOfItems unLinkItems

Array of items to be unlinked to unit group

Return Values

None.

DeleteItems

This method deletes heterogeneous items (groups/computers/user accounts/user profiles).

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

dsm:ArrayOfItems deleteItems

Array of items that must be deleted.

Return Values

None.

MoveItemsToUnitGroup

This method moves an item from one unit group to another.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

xsd:string oldGroupUuid

The uuid of the group to which the items are initially linked

xsd:string newGroupUuid

The uuid of the group to which the items need to be moved.

dsm:ArrayOfItems moveItems

Array of item UUIDs to be moved.

Return Values

None.

GetUnitGroupComputerMembers2

The GetUnitGroupComputerMembers2 method fetches the computers from a unit groups member list.

Parameters

xsd:string sessionId

Specifies the session identifier.

xsd:long listHandle

Specifies the numeric handle to a list returned by a previous call to OpenUnitGroupComputerMembersList2.

xsd:long noOfComputers

Specifies the total number of computers to fetch from the list. This is less than or equal to the total number in the list returned by a previous call to OpenUnitGroupComputerMembersList2.

Return Values

dsm:ArrayOfComputerProperties2

GetUnitGroupComputerMembersReturn

Specifies an [Array of ComputerProperties](#) (see page 988). One for each computer requested from the list.

Remarks

You can make multiple calls to GetUnitGroupComputerMembers2 to iterate down the list.

OpenUnitGroupComputerMembersList2

The OpenUnitGroupComputerMembersList2 method fetches a list of the computers that are members of the given unit group.

This method returns the total number of computers in the list and a handle to the list that can be used for further calls to GetUnitGroupComputerMembers2.

Parameters

xsd:string sessionId

Specifies the session identifier.

xsd:string unitGroupId

Specifies the UUID of the unit group member computers to be listed.

dsm:ArrayOfComputerPropertyFilter2 computerPropertyFilters

Specifies an Array of computer property filter definitions to limit the computers returned by the search.

xsd:boolean matchAll

Specifies the condition on the filter properties.

dsm: ComputerProperty2 sortProperty

Specifies the [Computer property](#) (see page 626) on which sorting takes place.

xsd:bool ascending

Specifies sorting in an ascending order.

dsm:ComputerPropertiesRequired2 computerPropertiesRequired (m)

Specifies the [properties required to get a computer](#) (see page 806) from the list.

Return Values

xsd:long OpenListHandle

Specifies the numeric handle to a list of unit groups that can be passed to GetUnitGroupComputerMembers2.

xsd:long totalNoOfComputers

Specifies the total number of computers in the list whose handle has been returned.

Remarks

None

GetUnitGroupComputerMembers3

The GetUnitGroupComputerMembers3 method fetches the computers in a unit group member list from a specified index number.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:long listHandle

Specifies the numeric handle to a list returned by a previous call to OpenUnitGroupComputerMembersList2.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

xsd:long noOfComputers

Specifies the number of objects you want to retrieve from the startIndex position. This number must be less than or equal to the total number in the list returned by a previous call to OpenUnitGroupComputerMembersList2.

Return Values

dsm:ArrayOfComputerProperties2**GetUnitGroupComputerMembersReturn**

Specifies an [Array of ComputerProperties](#) (see page 988). One for each computer requested from the list.

Remarks

You can make multiple calls to GetUnitGroupComputerMembers3 to iterate down the list. You must call the CloseUnitGroupComputerMembersList method to close the list.

GetUnitGroupByName

The GetUnitGroupByName method retrieves the properties of a specified unit group, identified by its name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Xsd:string unitGroupName

Specifies the name of the unit group, the properties of which you want to retrieve.

dsm:ObjectPreference objectPreference

Specifies whether the group must be a local or replicated group, or either of the two.

dsm: UnitGroupPropertiesRequired unitGroupPropertiesRequired

Specifies the group properties you want to retrieve.

Return Values**dsm: UnitGroupProperties GetUnitGroupReturn**

Contains the unit group properties retrieved.

Remarks

None

GetUnitGroupsXML

The GetUnitGroupsXML method returns the CA ITCM computer groups in an XML tree.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

Xsd:string parentGroup

Specifies the root group for which you need the xml tree. Specify NULL to return all the computer groups.

xsd:string excludeGroupName

Specifies the group name to be excluded from the XML, if any. The subgroups under the specified group are also excluded.

xsd:boolean excludeSystemGroups

Specifies whether you want to exclude system groups such as All Computers, All User Accounts, and All User Profiles.

Return Values**xsd: string groupsXML**

Returns the computer groups XML as a string.

Unit Inventory

This section describes unit inventory methods.

This section contains the following topics:

[CloseUnitInventoryComponentGroupList](#) (see page 534)

[CloseUnitInventoryComponentList](#) (see page 535)

[CloseUnitInventoryGroupList](#) (see page 535)

[CloseUnitInventoryItemList](#) (see page 536)

[GetUnitInventoryComponentGroups](#) (see page 536)

[GetUnitInventoryComponents](#) (see page 537)

[GetUnitInventoryGroups](#) (see page 538)

[GetUnitInventoryItems](#) (see page 538)

[OpenUnitInventoryComponentGroupList](#) (see page 539)

[OpenUnitInventoryComponentList](#) (see page 541)

[OpenUnitInventoryGroupList](#) (see page 542)

CloseUnitInventoryComponentGroupList

The `CloseUnitInventoryComponentGroupList` method closes an inventory component group list that was opened by `OpenUnitInventoryComponentGroupList` and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to the list to close. The handle was generated by a previous call to `OpenUnitInventoryComponentGroupList`.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitInventoryComponentList

The CloseUnitInventoryComponentList method closes an inventory component list that was opened by OpenUnitInventoryComponentList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitInventoryComponentList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitInventoryGroupList

The CloseUnitInventoryGroupList method closes an inventory group list that was opened by OpenUnitInventoryGroupList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitInventoryGroupList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitInventoryItemList

The CloseUnitInventoryItemList method closes an inventory item list that was opened by OpenUnitInventoryItemList and releases the list handle.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitInventoryItemList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

GetUnitInventoryComponentGroups

The GetUnitInventoryComponentGroups method retrieves the inventory component groups from a unit's list.

Parameters

xsd:string sessionId ("")

The session identifier..

xsd:long listHandle ("")

A numeric handle to a list returned by a previous call to OpenUnitInventoryComponentGroupList.

xsd:long noOfUnitInventoryComponentGroups ("")

The total number of component groupss to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenUnitInventoryComponentGroupList.

Return Values

ArrayOfUnitInventoryGroupProperties

GetUnitInventoryComponentGroupsReturn

An array of UnitInventoryGroupProperties - one for each group requested from the list.

Remarks

You can make multiple calls to GetUnitInventoryComponentGroups to iterate down the list.

More information:

[UnitInventoryGroupProperties](#) (see page 869)

GetUnitInventoryComponents

The GetUnitInventoryComponents method retrieves the inventory components from a unit's list.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list returned by a previous call to OpenUnitInventoryComponentList.

xsd:long noOfUnitInventoryComponents (")

The total number of components to get from the list. This is than or equal to the total number in the list returned by a previous call to OpenUnitInventoryComponentList.

Return Values

ArrayOfUnitInventoryComponentProperties

GetUnitInventoryComponentsReturn

An array of UnitInventoryComponentProperties - one for each component requested from the list.

Remarks

You can make multiple calls to GetUnitInventoryComponents to iterate down the list.

More information:

[UnitInventoryComponentProperties](#) (see page 868)

GetUnitInventoryGroups

The GetUnitInventoryGroups method retrieves the inventory groups from a unit's list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list returned by a previous call to OpenUnitInventoryGroupList.

xsd:long noOfUnitInventoryGroups ("")

The total number of groups to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenUnitInventoryGroupList.

Return Values

ArrayOfUnitInventoryGroupProperties GetUnitInventoryGroupsReturn

An array of UnitInventoryGroupProperties - one for each group requested from the list.

Remarks

Make multiple calls to GetUnitInventoryGroups to iterate down the list.

GetUnitInventoryItems

The GetUnitInventoryItems method retrieves the inventory items from a unit's list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list returned by a previous call to OpenUnitInventoryItemList.

xsd:long noOfUnitInventoryItems (")

The total number of inventory items to get from the list. This is less than or equal to the total number in the list returned by a previous call to `OpenUnitInventoryItemList`.

Return Values**ArrayOfUnitInventoryItemProperties GetUnitInventoryItemsReturn**

An array of `UnitInventoryItemProperties` - one for each item requested from the list.

Remarks

You can make multiple calls to `GetUnitInventoryItems` to iterate down the list.

More information:

[UnitInventoryItemProperties](#) (see page 869)

OpenUnitInventoryComponentGroupList

The `OpenUnitInventoryComponentGroupList` method opens a list of inventory component groups on a given unit.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string unitId (")

The UUID of the unit whose inventory component groups are to be listed.

UnitInventoryComponentProperties
unitInventoryComponentProperties ("")

The properties of the UnitInventoryComponent whose groups should be listed.

UnitInventoryGroupPropertiesRequired
unitInventoryGroupPropertiesRequired ("")

The properties that will be required when getting a group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of inventory component groups that can be passed to GetUnitInventoryComponentGroups

xsd:long totalNoOfUnitInventoryComponentGroups

The total number of inventory component groups in the list whose handle has been returned.

Remarks

The method returns the size of the list, and a handle to the list that you can use with the GetUnitInventoryComponentGroups method.

Asset Inventory is reported in a tree like structure. The root of this tree is called the *inventory component*, any groups under this root are called *inventory component groups*, and the subgroups in each component group are called *inventory groups*. The elements under inventory groups are called *inventory items*, as follows:

"General Inventory"	[Inventory Component]
"System"	[Inventory Component Group]
"Processors"	[Inventory Group]
"Speed" = "slow"	[Inventory Item]

More information:

[UnitInventoryComponentProperties](#) (see page 868)

[UnitInventoryGroupPropertiesRequired](#) (see page 869)

OpenUnitInventoryComponentList

The OpenUnitInventoryComponentList method opens a list of inventory components on a given unit.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string unitId (")

The UUID of the unit whose inventory components are to be listed.

UnitInventoryComponentPropertiesRequired

unitInventoryComponentPropertiesRequired (")

The properties that will be required when getting an inventory component from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of inventory components that can be passed to GetUnitInventoryComponents

xsd:long totalNoOfUnitInventoryComponents

The total number of inventory components in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that you can use with the GetUnitInventoryComponents method.

Asset Inventory is reported in a tree like structure. The root of this tree is called the *inventory component*, any groups under this root are called *inventory component groups*, and the subgroups in each component group are called *inventory groups*. The elements under inventory groups are called *inventory items*, as follows:

"General Inventory"	[Inventory Component]
"System"	[Inventory Component Group]
"Processors"	[Inventory Group]
"Speed" = "slow"	[Inventory Item]

More information:

[UnitInventoryComponentPropertiesRequired](#) (see page 868)

OpenUnitInventoryGroupList

The OpenUnitInventoryGroupList method opens a list of inventory groups on a given unit.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string unitId ("")

The UUID of the unit whose inventory groups are to be listed.

UnitInventoryGroupProperties unitInventoryGroupProperties ("")

The properties of the unitInventoryGroup whose member groups are to be listed.

**UnitInventoryGroupPropertiesRequired
unitInventoryGroupPropertiesRequired ("")**

The properties that will be required when getting a group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of inventory groups that can be passed to GetUnitInventoryGroups.

xsd:long totalNoOfUnitInventoryGroups

The total number of inventory groups in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that you can use with the GetUnitInventoryGroups method.

Asset Inventory is reported in a tree like structure. The root of this tree is called the *inventory component* and any groups under this root are called *inventory component groups* and the subgroups in each component group are called *inventory groups* .The elements under inventory groups are called *inventory items*, as follows:

"General Inventory"	[Inventory Component]
"System"	[Inventory Component Group]
"Processors"	[Inventory Group]
"Speed" = "slow"	[Inventory Item]

More information:

[UnitInventoryGroupProperties](#) (see page 869)

[UnitInventoryGroupPropertiesRequired](#) (see page 869)

Units

This section describes unit methods.

This section contains the following topics:

[CloseUnitDiscoveredSoftwareList](#) (see page 544)

[CloseUnitQueryList](#) (see page 544)

[CloseUnitUnitGroupList](#) (see page 545)

[CloseUnitUnitSoftwareInstallationJobList](#) (see page 545)

[CloseUnitUnitSoftwareJobList](#) (see page 546)

[CreateComputer](#) (see page 546)

[CreateComputer2](#) (see page 547)

[DeleteComputers](#) (see page 547)

[DeleteComputersByName](#) (see page 548)

[FindComputer](#) (see page 548)

[FindUserAccount](#) (see page 549)

[FindUserProfile](#) (see page 550)

[GetComputer](#) (see page 551)

[SetComputer](#) (see page 552)

[SetComputerByName](#) (see page 552)

[GetUnitDiscoveredSoftware](#) (see page 553)

[GetUnitQueries](#) (see page 554)

[GetUnitUnitGroups](#) (see page 555)

[GetUnitUnitSoftwareInstallationJobs](#) (see page 556)

[GetUnitUnitSoftwareInstallationJobs2](#) (see page 556)

[GetUnitUnitSoftwareJobs](#) (see page 557)

[GetUnitUnitSoftwareJobs2](#) (see page 558)

[GetUserAccount](#) (see page 559)

[GetUserProfile](#) (see page 560)

[OpenUnitDiscoveredSoftwareList](#) (see page 560)

[OpenUnitInventoryItemList](#) (see page 561)

[OpenUnitQueryList](#) (see page 563)

[OpenUnitUnitGroupList](#) (see page 563)

[OpenUnitUnitSoftwareInstallationJobList](#) (see page 564)

[OpenUnitUnitSoftwareJobList](#) (see page 565)

[GetUnitSoftwareJobList](#) (see page 567)

[GetComputerOSName](#) (see page 568)

[GetComputerByName](#) (see page 568)

[GetHardwareModelList](#) (see page 569)

[GetComputerHWModel](#) (see page 570)

[GetComputerHWModelByName](#) (see page 570)

[GetComputerSDProperties](#) (see page 570)

[GetComputerSDPropertiesByName](#) (see page 571)

CloseUnitDiscoveredSoftwareList

The CloseUnitDiscoveredSoftwareList method closes a discovered software list that was opened by OpenUnitDiscoveredSoftwareList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitDiscoveredSoftwareList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitQueryList

The CloseQueryList method closes a query list that was opened by OpenUnitQueryList and releases the list handle.

Parameters

xsd:string sessionId (")

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle

Specifies a numeric handle to a list as returned by a previous call to OpenUnitQueryList.

Return Values

None.

Remarks

None.

CloseUnitUnitGroupList

The CloseUnitUnitGroupList method closes a unit group list that was opened by OpenUnitUnitGroupList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitUnitGroupList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitUnitSoftwareInstallationJobList

The CloseUnitUnitSoftwareInstallationJobList method closes a software installation job list that was opened by OpenUnitUnitSoftwareInstallationJobList and releases the list handle.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitUnitSoftwareInstallationJobList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseUnitUnitSoftwareJobList

The CloseUnitUnitSoftwareJobList method closes a software job list that was opened by OpenUnitUnitSoftwareJobList and releases the list handle.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:long listHandle (^m)

A numeric handle to the list to close. The handle was generated by a previous call to OpenUnitUnitSoftwareJobList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CreateComputer

This method creates a computer.

Parameters

xsd:string sessionId (m)

Session Id. got from the login to the web service.

CreateComputerProperties propertiesForComputer (m)

The properties required to create a computer. Although the structure is a mandatory parameter, setting all the structure members is not mandatory

Return Values

xsd:string uuidOfCreatedComputer

If a computer is successfully created, the UUID of the new computer will be returned.

Remarks

None

More information:

[CreateComputerProperties](#) (see page 675)

CreateComputer2

This method creates a computer.

Parameters

xsd:string sessionId

Session Id. got from the login to the web service.

CreateComputerProperties2 computerProperties

The properties required to create a computer.

Return Values

xsd:string computerId

If a computer is successfully created, the UUID of the new computer will be returned.

DeleteComputers

Deletes list of computers specified by uuid.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ComputerType computerType(mandatory)

Type (machine, scalability server, docking device) of the computer to be deleted.

ArrayOfstring computerIds (m)

The list of uuids of computers to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteComputersByName

The web service api *DeleteComputersByName* deletes a list of computers specified by name.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ComputerType computerType(mandatory)

Type (machine, scalability server, docking device) of the computer to be deleted.

ArrayOfstring computerHostNames (m)

The list of host names of computers to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[ComputerProperties](#) (see page 669)

[ComputerPropertiesRequired](#) (see page 670)

FindComputer

The FindComputer method finds a computer unit given its name.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string computerName (m)

The computer name to search for. This may contain wildcard characters (*).

Xsd:long noOfComputers

The number of computers that should be returned from the search if more than one match is found.

ComputerPropertiesRequired computerPropertiesRequired

This parameter specifies which computer properties to retrieve. Mandatory if noOfComputers is greater than 0.

Return Values

ArrayOfComputerProperties FindComputerReturn

An array of computer properties structures, one for each computer matched by the search. This array will either be of the size of the number of computers matched by the search, or of the number of computers requested by the method call - whichever size is the lowest.

xsd:long totalNoOfComputersFound

The total number of computers that were matched by the search. You can use this to indicate that there were more matches found than the requested number of computers.

Remarks

The method supports wildcard pattern matching for the computer name. The name is case insensitive.

FindUserAccount

The FindUserAccount method finds a user account given its name.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string userName (")

The user account name to search for. This may contain wildcard characters (*).

Xsd:long noOfUserAccounts

The number of user accounts that should be returned from the search if more than 1 match is found.

UserAccountPropertiesRequired userAccountPropertiesRequired

This parameter specifies which user account properties to retrieve. Mandatory if noOfUserAccounts is greater than 0.

Return Values

ArrayOfUserAccountProperties FindUserAccountReturn

An array of user account properties structures, one for each account matched by the search. This array will either be of the size of the number of accounts matched by the search, or of the number of accounts requested by the method call - whichever size is the lowest.

xsd:long totalNoOfUserAccountsFound

The total number of user accounts that were matched by the search (this can be used to indicate whether there were more matches found than the requested number of user accounts).

Remarks

The method supports wildcard pattern matching for the user account name. The name is case insensitive.

More information:

[UserAccountProperties](#) (see page 890)

[UserAccountPropertiesRequired](#) (see page 891)

FindUserProfile

The FindUserProfile method finds a user profile given its name.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string userProfileLabel (^m)

The user profile display name to search for. This may contain wildcard characters (*).

xsd:long noOfUserProfiles

This parameter can be 0 and in this case the method retrieves the totalNoOfUserProfilesFound (match found).

The number of user profiles to return from the search if more than 1 match is found.

UserProfilePropertiesRequired userProfilePropertiesRequired

Specifies which user profile properties to retrieve.

Mandatory if noOfUserProfiles is greater than 0.

Return Values

ArrayOfUserProfileProperties FindUserProfileReturn

An array of user profile properties structures, one for each profile matched by the search. This array will either be of the size of the number of profiles matched by the search, or of the number of profiles requested by the method call - whichever size is the lowest.

xsd:long totalNoOfUserProfilesFound

The total number of user profiles that were matched by the search (this can be used to indicate whether there were more matches found than the requested number of user profiles).

Remarks

The method supports wildcard pattern matching for the user profile name. The name is case insensitive.

More information:

[UserProfileProperties](#) (see page 892)

[UserProfilePropertiesRequired](#) (see page 896)

GetComputer

The GetComputer method retrieves a computer identified by its UUID.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string computerId (^m)

The UUID of the computer to retrieve.

ComputerPropertiesRequired computerPropertiesRequired (^m)

This parameter specifies the computer properties to retrieve.

Return Values

ComputerProperties GetComputerReturn

The computer properties retrieved.

Remarks

None

More information:

[ComputerPropertiesRequired](#) (see page 670)

SetComputer

Sets properties of a computer or docking device.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string computerId (m)

UUID of computer whose properties need to be modified.

SetComputerProperties setComputerProperties (m)

The values of the properties that need to be modified. Setting all the structure members is not mandatory

Return Values

This is a void method that does not expose a return value.

Remarks

This method can be used only with computers and docking devices; it cannot be used for setting the properties of a scalability server.

If you use the setComputerProperties method to change the RAC policy parameter of a computer in the offline mode, the SetComputer API generates an error message.

More information:

[SetComputerProperties](#) (see page 730)

SetComputerByName

Sets properties of a computer/docking device.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string compterHostName (m)

Host name of computer whose properties need to be modified.

SetComputerProperties setComputerProperties (m)

The values of the properties that need to be modified. Although the structure is a mandatory parameter, setting all the structure members is not mandatory.

Return Values

This is a void method that does not expose a return value.

Remarks

This method can be used only with computers and docking devices; it cannot be used for setting the properties of a scalability server.

If you use the setComputerProperties method to change the RAC policy parameter of a computer in the offline mode, the SetComputerByName API generates an error message.

More information:

[SetComputerProperties](#) (see page 730)

GetUnitDiscoveredSoftware

The GetUnitDiscoveredSoftware method retrieves the software items from a unit's discovered software list.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:long listHandle (m)

A numeric handle to a list returned by a previous call to OpenUnitDiscoveredSoftwareList.

xsd:long noOfDiscoveredSoftwares (m)

The total number of software items to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenUnitDiscoveredSoftwareList.

Return Values

ArrayOfDiscoveredSoftwareProperties

GetUnitDiscoveredSoftwareReturn

An array of DiscoveredSoftwareProperties - one for each software item requested from the list.

Remarks

You can make multiple calls to GetUnitDiscoveredSoftware to iterate down the list.

More information:

[DiscoveredSoftwareProperties](#) (see page 700)

GetUnitQueries

The GetUnitQueries method retrieves the queries from a query list.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:long listHandle

Specifies a numeric handle to a list as returned by a previous call to OpenUnitQueryList.

xsd:long noOfQueries (m)

Specifies the total number of queries to get from the list. This number must be less than or equal to the total number in the list as returned by a previous call to OpenUnitQueryList.

Return Values

xsd:dsm ArrayOfUnitQueryProperties GetQueriesReturn

Returns an array of UnitQueryProperties—one for each query requested from the list.

UnitQueryProperties queryPropertiesArray

Returns query properties for each query in the list.

Remarks

None.

More information:

[UnitQueryProperties](#) (see page 880)

GetUnitUnitGroups

The GetUnitUnitGroups method retrieves the procedures from a list of unit groups of which a unit is a member.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:long listHandle (")

A numeric handle to a list returned by a previous call to OpenUnitUnitGroupList.

xsd:long noOfUnitGroups (")

The total number of groups to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenUnitUnitGroupList.

Return Values

ArrayOfUnitGroupProperties GetUnitUnitGroupsReturn

An array of UnitGroupProperties - one for each group requested from the list.

Remarks

You can make multiple calls to GetUnitUnitGroups to iterate down the list.

More information:

[UnitGroupProperties](#) (see page 870)

GetUnitUnitSoftwareInstallationJobs

The GetUnitUnitSoftwareInstallationJobs method retrieves the software installation jobs from a unit's software installation job list.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:long listHandle ("")

A numeric handle to a list returned by a previous call to OpenUnitUnitSoftwareInstallationJobList.

xsd:long noOfUnitSoftwareInstallationJobs ("")

The total number of software installation jobs to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenUnitUnitSoftwareInstallationJobList.

Return Values

ArrayOfUnitSoftwareJobProperties

GetUnitUnitSoftwareInstallationJobsReturn

An array of UnitSoftwareJobProperties - one for each software package installation requested from the list.

Remarks

You can make multiple calls to GetUnitUnitSoftwareInstallationJobs to iterate down the list.

More information:

[UnitSoftwareJobProperties](#) (see page 880)

[ArrayOfUnitSoftwareJobProperties2](#) (see page 987)

GetUnitUnitSoftwareInstallationJobs2

The GetUnitUnitSoftwareInstallationJobs2 method retrieves the software installation jobs from a unit's software installation job list. This method also retrieves software package format (regular or virtual).

Parameters

xsd:string sessionId ("")

The session identifier

xsd:long listHandle ("")

A numeric handle to a software jobs list, as returned by a previous call to OpenUnitUnitSoftwareInstallationJobList

xsd:long noOfUnitSoftwareInstallationJobs (")

The total number of software installation jobs to get from the list. This number is less than or equal to the total number in the list, as returned by a previous call to `OpenUnitUnitSoftwareInstallationJobList`.

Return Values**ArrayOfUnitSoftwareJobProperties2****GetUnitUnitSoftwareInstallationJobsReturn**

An array of `UnitSoftwareJobProperties` - one for each software package installation requested from the list

Remarks

You can make multiple calls to `GetUnitUnitSoftwareInstallationJobs2` to iterate down the list.

More information:

[UnitSoftwareJobProperties2](#) (see page 883)

GetUnitUnitSoftwareJobs

The `GetUnitUnitSoftwareJobs` method retrieves the software jobs from a unit's software job list.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:long listHandle (")

A numeric handle to a list returned by a previous call to `OpenUnitUnitSoftwareJobList`.

xsd:long noOfUnitSoftwareJobs

The total number of software jobs to get from the list. This is less than or equal to the total number in the list returned by a previous call to GetUnitUnitSoftwareJobs.

Return Values

ArrayOfUnitSoftwareJobProperties GetUnitUnitSoftwareJobsReturn

An array of UnitSoftwareJobProperties - one for each software job requested from the list.

Remarks

You can make multiple calls to GetUnitUnitSoftwareJobs to iterate down the list.

More information:

[UnitSoftwareJobProperties](#) (see page 880)

More Information:

[ArrayOfUnitSoftwareJobProperties2](#) (see page 987)

GetUnitUnitSoftwareJobs2

The GetUnitUnitSoftwareJobs2 method retrieves the software jobs from a unit's software job list. This method also retrieves software package format (regular or virtual).

Parameters

xsd:string sessionId (")

The session identifier

xsd:long listHandle (")

A numeric handle to a software jobs list, as returned by a previous call to OpenUnitUnitSoftwareJobList

xsd:long noOfUnitSoftwareJobs

The total number of software jobs to get from the list. This number is less than or equal to the total number in the list, as returned by a previous call to `OpenUnitUnitSoftwareJobList`.

Return Values**ArrayOfUnitSoftwareJobProperties2 GetUnitUnitSoftwareJobsReturn**

An array of `UnitSoftwareJobProperties` - one for each software job installation requested from the list

Remarks

You can make multiple calls to `GetUnitUnitSoftwareJobs2` to iterate down the list.

GetUserAccount

The `GetUserAccount` method retrieves a user account identified by its UUID.

Parameters**xsd:string sessionId (^m)**

The session identifier.

xsd:string userAccountId (^m)

The UUID of the user account to retrieve.

UserAccountPropertiesRequired userAccountPropertiesRequired (^m)

This parameter specifies the user account properties to retrieve.

Return Values**UserAccountProperties GetUserAccountReturn**

The user account properties retrieved.

Remarks

None

More information:

[UserAccountProperties](#) (see page 890)

[UserAccountPropertiesRequired](#) (see page 891)

GetUserProfile

The GetUserProfile method retrieves a user profile identified by its UUID.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string userProfileId (^m)

The UUID of the user profile to retrieve.

UserProfilePropertiesRequired userProfilePropertiesRequired (^m)

This parameter specifies the user profile properties to retrieve.

Return Values

UserProfileProperties GetUserProfileReturn

The user profile properties retrieved.

Remarks

None

More information:

[UserProfileProperties](#) (see page 892)

[UserProfilePropertiesRequired](#) (see page 896)

OpenUnitDiscoveredSoftwareList

The OpenUnitDiscoveredSoftwareList opens a list of the software that has been discovered on a given unit.

Parameters

xsd:string sessionId (^m)

The session identifier.

xsd:string unitId (^m)

The UUID identifying the unit whose discovered software are to be listed.

**DiscoveredSoftwarePropertiesRequired
discoveredSoftwarePropertiesRequired (")**

The properties that will be required when getting an individual software item from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of software items that can be passed to GetUnitDiscoveredSoftware

xsd:long totalNoOfDiscoveredSoftwares

The properties that will be required when getting an individual software item from the list.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetUnitDiscoveredSoftware method.

More information:

[DiscoveredSoftwarePropertiesRequired](#) (see page 701)

OpenUnitInventoryItemList

The OpenUnitInventoryItemList method opens a list of inventory items on a given unit.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string unitId (")

The UUID of the unit whose inventory items are to be listed.

UnitInventoryGroupProperties unitInventoryGroupProperties ("")

The properties of the UnitInventoryGroup whose inventory items are to be listed.

**UnitInventoryItemPropertiesRequired
unitInventoryItemPropertiesRequired ("")**

The properties that will be required when getting an inventory item from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of inventory items that can be passed to GetUnitInventoryItems.

xsd:long totalNoOfUnitInventoryItems

The total number of inventory items in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that you can use with the GetUnitInventoryItems method.

Asset Inventory is reported in a tree like structure. The root of this tree is called the *inventory component* and any groups under this root are called *inventory component groups* and the subgroups in each component group are called *inventory groups*. The elements under inventory groups are called *inventory items*, as follows:

"General Inventory"	[Inventory Component]
"System"	[Inventory Component Group]
"Processors"	[Inventory Group]
"Speed" = "slow"	[Inventory Item]

More information:

[UnitInventoryGroupProperties](#) (see page 869)

[UnitInventoryItemPropertiesRequired](#) (see page 870)

OpenUnitQueryList

The `OpenUnitQueryList` method opens a list of queries that can be linked with a unit group. The method returns the total number of queries in the list and a handle to the list that you can use for further calls to `GetUnitQueries` and `CloseUnitQueryList`.

Parameters

xsd:string sessionId ("")

Specifies the session ID obtained from the login to the web service.

UnitQueryPropertiesRequired queryPropertiesRequired

Specifies which query properties to retrieve.

Return Values

xsd:long OpenListHandle

Returns a numeric handle to a list of queries that can be passed to `GetUnitQueries`.

xsd:long totalNoOfQueries

Returns the total number of queries in the list whose handle has been returned.

Remarks

None.

More information:

[UnitQueryPropertiesRequired](#) (see page 879)

OpenUnitUnitGroupList

The `OpenUnitUnitGroupList` opens a list of unit groups of which the given unit is a member.

Parameters

xsd:string sessionId ("")

The session identifier.

xsd:string unitId ("")

The UUID of the unit whose group memberships are to be listed.

UnitType unitType ("")

The type of the given unit.

UnitGroupPropertiesRequired unitGroupPropertiesRequired ("")

The properties that will be required when getting a unit group from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of unit groups that can be passed to GetUnitUnitGroups

xsd:long totalNoOfUnitGroups

The total number of unit groups in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetUnitUnitGroups method.

More information:

[UnitGroupPropertiesRequired](#) (see page 876)

OpenUnitUnitSoftwareInstallationJobList

The OpenUnitUnitSoftwareInstallationJobList method opens a list of software installations associated with a given unit.

Parameters**xsd:string sessionId ("")**

The session identifier.

xsd:string unitId ("")

The UUID identifying the unit whose software installation jobs are to be listed.

UnitSoftwareJobPropertiesRequired
unitSoftwareJobPropertiesRequired (")

The properties that will be required when getting an individual installation from the list.

Return Values**xsd:long OpenListHandle**

A numeric handle to a list of software package installations that can be passed to GetUnitUnitSoftwareInstallationJobs

xsd:long totalNoOfUnitSoftwareInstallationJobs

The total number of installation jobs in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetUnitUnitSoftwareInstallationJobs method.

More information:

[UnitSoftwareJobPropertiesRequired](#) (see page 886)

OpenUnitUnitSoftwareJobList

The OpenUnitUnitSoftwareJobList opens a list of software jobs associated with a given unit.

Parameters**xsd:string sessionId (")**

The session identifier.

xsd:string unitId (")

The UUID identifying the unit whose software jobs are to be listed.

UnitSoftwareJobPropertiesRequired
unitSoftwareJobPropertiesRequired (")

The properties that will be required when getting an individual software job from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of software package installations that can be passed to GetUnitUnitSoftwareJobs

xsd:long totalNoOfUnitSoftwareJobs

The total number of software jobs in the list whose handle has been returned.

Remarks

This method returns the size of the list, and a handle to the list that can be used with the GetUnitUnitSoftwareJobs method.

More information:

[UnitSoftwareJobPropertiesRequired](#) (see page 886)

GetUnitSoftwareJobList

The GetUnitSoftwareJobList method retrieves the list of software jobs deployed on a unit. This method is supported only on the domain manager.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string unitId

Specifies the UUID of the unit for which you want to the installations list.

dsm:UnitSoftwareJobPropertiesRequired**unitSoftwareJobPropertiesRequired**

Specifies the properties that want to get from each installation in the list.

Xsd:long index

Specifies the starting index for the list.

xsd:long numJobsRequired

Specifies the number of jobs to get from the list.

Return Values

Xsd:int totalNumJobs

Returns the total number of jobs in the list.

dsm:ArrayOfUnitSoftwareJobProperties3**arrOfUnitSoftwareJobProperties**

Returns an array of UnitSoftwareJobProperties - one for each software package installation that are requested from the list.

Remarks

None

More information:

[UnitSoftwareJobProperties3](#) (see page 911)

[ArrayOfUnitSoftwareJobProperties3](#) (see page 984)

GetComputerOSName

The GetComputerOSName method returns the operating system name based on specified OS class ID.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string osClassID

Specifies the class ID for which you want to retrieve the operating system name.

Return Values

xsd:string osName

Contains the operating system name for the specified class ID.

GetComputerByName

The GetComputerByName method retrieves the properties of the computer, identified by the host name specified.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string computerHostName

Specifies the host name of the computer to search for. You can include wildcard characters such as * and ?.

Xsd:long noOfComputers

Specifies the number of computers that must be returned from the search, if more than one match is found.

ComputerPropertiesRequired computerPropertiesRequired

Specifies the computer properties to be retrieved. This parameter is mandatory, if noOfComputers is greater than 0.

Return Values

ArrayOfComputerProperties computerArray

Includes an array of computer properties structures, one for each computer matched by the search. The size of this array is either the number of computers matched by the search or the number of computers specified in the noOfComputers parameter, whichever is the lowest.

xsd:long totalNoOfComputersFound

Includes the total number of computers that were matched by the search. This value indicates that there were more matches found than the requested number of computers.

Remarks

The method supports wildcard pattern matching for the computer hostname. The name is case insensitive. Use FindComputer API to search for computers using compute label. Use GetComputerByName API to search for computers using compute hostname.

GetHardWareModelList

The GetHardWareModelList method returns the distinct hardware models in the CA ITCM environment.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfHWModelFilter filterArray

Specifies the filter properties that are supported.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

Xsd:long index

Specifies an index number from which you want the items must be returned from the result set. The index is zero-based.

Xsd:long numRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values**dsm:ArrayOfstring hwModelList**

Contains the distinct hardware models list.

GetComputerHWModel

The GetComputeHWModel method returns the hardware model for a list of computers identified by their UUID.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfstring computeUUIDs (m)

Specifies the array of computer UUIDs, the hardware models of which you want to retrieve.

Return Values

dsm:ArrayOfstring hwModels

Contains the hardware models of the specified computers.

GetComputerHWModelByName

The GetComputeHWModelByName method returns the hardware model for a list of computers identified by host name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

dsm:ArrayOfstring computerHostNames (m)

Specifies the array of host names, the hardware models of which you want to retrieve.

Return Values

dsm:ArrayOfstring hwModels

Contains the hardware models of the specified computers.

GetComputerSDProperties

The GetComputerSDProperties method returns the software delivery related properties for a particular computer identified by UUID.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string computerUUID

Specifies the UUID of the computer.

Return Values

dsm: SDProperties props

Contains the software delivery properties of the computer.

GetComputerSDPropertiesByName

The GetComputerSDPropertiesByName method returns the software delivery related properties for a particular computer identified by hostname.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string computerName

Specifies the name of the computer.

Return Values

dsm: SDProperties props

Contains the software delivery properties of the computer.

User Profiles

This section contains the following topics:

[CreateUserProfile](#) (see page 571)

[DeleteUserProfiles](#) (see page 572)

[DeleteUserProfilesByName](#) (see page 572)

[SetUserProfile](#) (see page 573)

[SetUserProfileByName](#) (see page 573)

CreateUserProfile

The web service api *CreateUserProfile*, creates the common object user profile. It also configure a common and sd agent with it.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

CreateUserProfileProperties createUserProfileProperties (m)

The properties required to create the user profile. Setting all the structure members is not mandatory.

Return Values

xsd:string uuidOfCreatedUserProfile

If the user profile is successfully created, the UUID of the new user profile is returned.

Remarks

None

More information:

[CreateUserProfileProperties](#) (see page 696)

DeleteUserProfiles

The web service api *DeleteUserProfiles* deletes a list of user profiles specified by uuid.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring userProfileIds (m)

The list of uuids of user profiles to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

DeleteUserProfilesByName

The web service api *DeleteUserProfilesByName* deletes a list of user profiles specified by name.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring userProfileNames (m)

The list of names of user profiles to delete.

Return Values

This is a void method that does not expose a return value.

Remarks

None

SetUserProfile

Sets properties of an user profile.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string userProfileId (m)

UUID of user profile whose properties need to be modified.

SetUserProfileProperties setUserProfileProperties (m)

The values of the properties that need to be modified. Setting all the structure members is not mandatory.

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetUserProfileProperties](#) (see page 767)

SetUserProfileByName

Sets properties of an user profile.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string userProfileName (m)

Name of user profile whose properties need to be modified.

SetUserProfileProperties setUserProfileProperties (m)

The values of the properties that need to be modified. Setting all the structure members is not mandatory

Return Values

This is a void method that does not expose a return value.

Remarks

None

More information:

[SetUserProfileProperties](#) (see page 767)

Enterprise Manager Distribution Functionalities

This section contains the following topics:

[AddDeregisterSoftwarePackageOrdersToDistributionContainer](#) (see page 576)

[AddDeregisterSoftwarePackageProcedureOrdersToDistributionContainer](#) (see page 576)

[AddDeregisterSoftwarePolicyOrdersToDistributionContainer](#) (see page 577)

[AddFetchItemJobOrdersToDistributionContainer](#) (see page 578)

[AddRegisterSoftwareJobContainerOrdersToDistributionContainer](#) (see page 578)

[AddDeregisterSoftwareJobContainerOrdersToDistributionContainer](#) (see page 579)

[AddRegisterSoftwarePackageOrdersToDistributionContainer](#) (see page 580)

[AddRegisterSoftwarePackageProcedureOrdersToDistributionContainer](#) (see page 580)

[AddRegisterSoftwarePolicyOrdersToDistributionContainer](#) (see page 581)

[CloseDistributionContainerDistributionDomainList](#) (see page 582)

[CloseDistributionContainerList](#) (see page 582)

[CloseDistributionContainerOrders](#) (see page 583)

[CreateDistributionContainer](#) (see page 583)

[DeleteDistributionContainers](#) (see page 584)

[GetDistributionContainer](#) (see page 585)

[GetDistributionContainerDistributionDomains](#) (see page 585)

[GetDistributionContainerOrder](#) (see page 586)

[GetDistributionContainerOrder3](#) (see page 587)

[GetDistributionContainerOrders](#) (see page 588)

[GetDistributionContainerOrders3](#) (see page 589)

[GetDistributionContainers](#) (see page 590)

[HaltDeliveryDistributionContainers](#) (see page 591)

[OpenDistributionContainerDistributionDomainList](#) (see page 591)

[OpenDistributionContainerList](#) (see page 592)

[OpenDistributionContainerOrderList](#) (see page 593)

[RenewDeliveryDistributionContainers](#) (see page 594)

[ResumeDeliveryDistributionContainers](#) (see page 594)

[SendDistributionContainer](#) (see page 595)

[SetDistributionContainer](#) (see page 596)

AddDeregisterSoftwarePackageOrdersToDistributionContainer

This API adds deregister software package orders to a distribution container.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders is added.

ArrayOfSoftwarePackageOrder softwarePackageOrders (m)

An array of SoftwarePackageOrder to be added to the distribution container.

Return Values

ArrayOfstring orderIds

An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

More information:

[SoftwarePackage](#) (see page 865)

AddDeregisterSoftwarePackageProcedureOrdersToDistributionContainer

This API adds deregister software package procedure orders to a distribution container.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders is added.

**ArrayOfSoftwarePackageProcedureOrder
softwarePackageProcedureOrders (m)**

An array of SoftwarePackageProcedureOrder to be added to the distribution container.

Return Values**ArrayOfstring orderIds**

An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

More information

[SoftwarePackageProcedureOrder](#) (see page 863)

AddDeregisterSoftwarePolicyOrdersToDistributionContainer

This API adds deregister software policy orders to a distribution container.

Parameters**xsd:string sessionId (m)**

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders are added.

ArrayOfSoftwarePolicyOrder softwarePolicyOrders (m)

An array of SoftwarePolicyOrder to be added to the distribution container

Return Values**ArrayOfstring orderIds**

An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

More information:

[SoftwarePolicyOrder](#) (see page 863)

AddFetchItemJobOrdersToDistributionContainer

This webservice interface adds FetchItemJobOrders to DistributionContainer.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

string distributionContainerId (m)

The UUID of the distribution container order.

ArrayOfFetchItemJobOrderProperties fetchItemJobOrders (m)

An Array of FetchItemJobOrders to be added to the Distribution Container.

Return Values

xsd:ArrayOfstring fetchItemJobOrderIds

An array of FetchItemJobOrder UUID's that are successfully added to the Distribution Container.

Remarks

None

More information:

[FetchItemJobOrderProperties](#) (see page 710)

AddRegisterSoftwareJobContainerOrdersToDistributionContainer

This API adds register software job container orders to a distribution container.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders are added.

ArrayOfSoftwareJobContainerOrder softwareJobContainerOrders (m)

An array of SoftwareJobContainerOrder to be added to the distribution container

Return Values**ArrayOfstring orderIds**

An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

More information:

[SoftwareJobContainerOrder](#) (see page 863)

AddDeregisterSoftwareJobContainerOrdersToDistributionContainer

This API adds deregister software job container orders to a distribution container.

Parameters**xsd:string sessionId (m)**

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders are added.

ArrayOfSoftwareJobContainerOrder softwareJobContainerOrders (m)

An array of SoftwareJobContainerOrder to be added to the distribution container

Return Values**ArrayOfstring orderIds**

An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

More information:

[SoftwareJobContainerOrder](#) (see page 863)

AddRegisterSoftwarePackageOrdersToDistributionContainer

This API adds register software package orders to a distribution container.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders is added.

ArrayOfSoftwarePackageOrder softwarePackageOrders (m)

An array of SoftwarePackageOrder to be added to the distribution container .

Return Values

ArrayOfstring orderIds

An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

More information:

[SoftwarePackageOrder](#) (see page 865)

AddRegisterSoftwarePackageProcedureOrdersToDistributionContainer

This API adds register software package procedure orders to a distribution container.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders are added.

**ArrayOfSoftwarePackageProcedureOrder
softwarePackageProcedureOrders (m)**

An array of SoftwarePackageProcedureOrder to be added to the distribution container.

Return Values**ArrayOfstring orderIds**

An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

More information:

[SoftwarePackageProcedureOrder](#) (see page 863)

AddRegisterSoftwarePolicyOrdersToDistributionContainer

This API adds register software policy orders to a distribution container.

Parameters**xsd:string sessionId (m)**

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to which orders are added.

ArrayOfSoftwarePolicyOrder softwarePolicyOrders (m)

An array of SoftwarePolicyOrder to be added to the distribution container

Return Values

ArrayOfstring orderIds An array of UUIDs of successfully added orders.

Remarks

An order is inserted at the requested position. If you try to add an order at an occupied position, all current orders from that position are shifted one position.

CloseDistributionContainerDistributionDomainList

The CloseDistributionContainerDistributionDomainList method closes a distribution domain list that was opened by OpenDistributionContainerDistributionDomainList and releases the list handle.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd: long listHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenDistributionContainerDistributionDomainList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseDistributionContainerList

The CloseDistributionContainerList method closes a distribution container list that was opened by OpenDistributionContainerList and releases the list handle.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to OpenDistributionContainerList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseDistributionContainerOrders

This method closes the Distribution Container Order List opened by `OpenDistributionContainerOrderList` and releases the list handle.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:long openListHandle (m)

A numeric handle to a list that should be closed. The handle was generated by a previous call to `OpenDistributionContainerOrderList`.

Return Values

This is a void method that does not expose a return value

Remarks

None

CreateDistributionContainer

The `CreateDistributionContainer` method creates a new distribution container

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

**CreateDistributionContainerProperties
createDistributionContainerProperties (m)**

The properties required to create a distribution container.

Return Values

xsd:string distributionContainerId

If a distribution container was successfully created, the UUID of the new distribution container is returned.

Remarks

When you send a container, a separate sd task is created that starts and monitors the distribution. SD tasks are handled by the SD Task Manager. The new task gets the provided priority. Valid values are 1-10. 1 is highest priority. The priority controls how often (relative to the other tasks) the SD Task Manager executes a task.

More information:

[CreateDistributionContainerProperties](#) (see page 631)

[DistributionContainerProperties](#) (see page 691)

DeleteDistributionContainers

The DeleteDistributionContainers method deletes one or more distribution containers.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring distributionContainerIds (m)

An array of distribution container UUIDs identifying the distribution containers to delete.

Return Values

None

Remarks

None

GetDistributionContainer

The GetDistributionContainer method retrieves properties of a distribution container identified by its UUID.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

The UUID of the distribution container to retrieve.

Return Values

DistributionContainerProperties distributionContainerProperties

The distribution container properties retrieved.

Remarks

An order is inserted at the requested position. That is, if you try to add an order at an already occupied position, all current orders from that position are shifted one position.

GetDistributionContainerDistributionDomains

The GetDistributionContainersDistributionDomains method retrieves the distribution domains targeted by the distribution container.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd: long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenDistributionContainerDistributionDomainList.

xsd: long fromIndex (m)

A numeric index from where the 'noOfDistributionDomains' is read. If it is less than or equal to zero (0), 'noOfDistributionDomains' is read from previous index location (fromIndex + noOfDistributionDomains supplied in previous call to same Get method).

Note: If fromIndex is supplied as less than zero (0), it is considered as zero (0).

xsd:long noOfDistributionDomains (m)

The total number of distribution domains to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenDistributionContainerDistributionDomainList.

Return Values

ArrayOfDistributionDomainProperties
arrayOfDistributionDomainProperties

An array of DistributionDomainProperties - one for each distribution domain requested from the list.

Remarks

None

More information:

[DistributionDomainProperties](#) (see page 702)

GetDistributionContainerOrder

This webservice interface gets the DistributionContainer order details for the given DistributionContainerOrder. It will take distributionContainerOrderId asInput and returns DistributionContainerOrderProperties

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerOrderId (m)

The UUID of the DistributionContainerOrder whose properties to be retrieved.

DistributionContainerOrderPropertiesRequired
distributionContainerOrderPropertiesRequired (m)

The required properties of the DistributionContainerOrder.

Return Values

DistributionContainerOrderProperties
distributionContainerOrderProperties

The retrieved distribution container order properties.

Remarks

None

More information:

[DistributionContainerOrderPropertiesRequired](#) (see page 692)

GetDistributionContainerOrder3

The GetDistributionContainerOrder3 gets the DistributionContainerOrder details for the given distribution container order. The method takes distributionContainerOrderId as input and returns DistributionContainerOrderProperties.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved DistributionContainerOrderProperties, which has linked job container properties in the sequence SoftwareJobContainerProperties3. This sequence includes the userMessage parameter that specifies the custom administrator message information linked with the job container, that is, associated with the distribution container in SD.

Parameters

xsd:string sessionId (m)

Specifies the session identifier obtained from the login method of the web service.

xsd:string distributionContainerOrderId (m)

Specifies the UUID of the DistributionContainerOrder whose properties are to be retrieved.

DistributionContainerOrderPropertiesRequired distributionContainerOrderPropertiesRequired (m)

Specifies the required properties of the DistributionContainerOrder.

Return Values

DistributionContainerOrderProperties3 distributionContainerOrderProperties3

Returns the retrieved distribution container order properties.

Remarks

None

More information:

[DistributionContainerOrderPropertiesRequired](#) (see page 692)

GetDistributionContainerOrders

This method retrieves the *DistributionContainerOrders* from the *DistributionContainerOrders* list that is opened in previous call to *OpenDistributionContainerOrderList*.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:long openListHandle (m)

A numeric handle to a list as returned by a previous call to *OpenDistributionContainerOrderList*

xsd:long fromIndex (m)

A numeric index from where the *noOfDistributionContainerOrders* is read. If it is less than or equal to zero then *noOfDistributionContainerOrders* is read from previous index location (fromIndex + *noOfDistributionContainerOrders* supplied in previous call to *OpenDistributionContainerOrderList*.)

xsd:long numberOfDistributionContainerOrders (m)

The total number of *DistributionContainerorders* to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to *OpenDistributionContainerOrderList*.

Return Values

**ArrayOfDistributionContainerOrderProperties
distributionContainerOrders**

An array of *DistributionContainerOrderProperties*one for each *Distribution Container Order* requested from the list.

Remarks

None

More information:

[DistributionContainerOrderProperties](#) (see page 701)

GetDistributionContainerOrders3

The GetDistributionContainerOrders3 method retrieves the DistributionContainerOrders properties from the DistributionContainerOrders list that is opened as a result of a previous call to OpenDistributionContainerOrderList.

This method also includes the functionality that helps you manage the custom administrator message. The method returns the retrieved array of DistributionContainerOrderProperties in which each distribution container has one linked job container properties in the sequence SoftwareJobContainerProperties3. The sequence includes the userMessage parameter that specifies the custom administrator message information linked with the job container, that is associated with the distribution container in SD.

Parameters

xsd:string sessionId (m)

Specifies the session identifier obtained from the login method of the web service.

xsd:long openListHandle (m)

Specifies a numeric handle to a list as returned by a previous call to OpenDistributionContainerOrderList.

xsd:long fromIndex (m)

Specifies a numeric index from where the *noOfDistributionContainerOrders* is read. If it is less than or equal to zero, then *noOfDistributionContainerOrders* is read from previous index location (*fromIndex* + *noOfDistributionContainerOrders* supplied in a previous call to OpenDistributionContainerOrderList.)

xsd:long numberOfDistributionContainerOrders (m)

Specifies the total number of *DistributionContainerorders* to get from the list. This number must be less than or equal to the total number in the list as returned by a previous call to OpenDistributionContainerOrderList.

Return Values

ArrayOfDistributionContainerOrderProperties3 distributionContainerOrders3

Specifies an array of DistributionContainerOrderProperties3—one for each distribution container order requested from the list.

Remarks

None

More information:

[DistributionContainerOrderProperties3](#) (see page 702)

GetDistributionContainers

The GetDistributionContainers method retrieves the distribution containers from a distribution container list.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd: long listHandle (m)

A numeric handle to a list as returned by a previous call to OpenDistributionContainerList.

xsd: long fromIndex (m)

A numeric index from where the 'noOfDistributionContainers' is read. If it is less than or equal to zero (0), 'noOfDistributionContainers' is read from previous index location (fromIndex + noOfDistributionContainers supplied in previous call to same Get method).

Note: If fromIndex is supplied as less than zero (0), it is considered as zero (0).

xsd:long noOfDistributionContainers (m)

The total number of distribution containers to get from the list. This should be less than or equal to the total number in the list as returned by a previous call to OpenDistributionContainerList.

Return Values

ArrayOfDistributionContainerProperties

arrayOfDistributionContainerProperties

An array of DistributionContainerProperties one for each container requested from the list.

Remarks

None

More information:

[DistributionContainerProperties](#) (see page 691)

HaltDeliveryDistributionContainers

The HaltDeliveryDistributionContainers method halts one or more ongoing distributions. The distributions can be subsequently deleted or resumed.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring distributionContainerIds (m)

An array of distribution container UUIDs identifying the distribution containers to halt.

Return Values

None

Remarks

A distribution can only be halted when Distribution Status is "Waiting" or "Active".

OpenDistributionContainerDistributionDomainList

The OpenDistributionContainerDistributionDomainList method opens a list of distribution domains associated with a distribution container.

The method returns the total number of distribution domains in the list, and a handle to the list that can be used for further calls to GetDistributionContainerDistributionDomains.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container whose distribution domains is to be listed.

DistributionDomainPropertiesRequired
distributionDomainPropertiesRequired (m)

The properties that are required when getting a distribution domain from the list.

Return Values

xsd:long openListHandle

A numeric handle to a list of distribution domains that can be passed to GetDistributionContainerDistributionDomains

xsd:long totalNoOfDistributionDomains

The total number of distribution domains in the list.

Remarks

None

More information:

[DistributionDomainPropertiesRequired](#) (see page 703)

OpenDistributionContainerList

The OpenDistributionContainerList method opens a list of distribution containers.

The method returns the total number of distribution containers in the list, and a handle to the list that you can use for further calls to GetDistributionContainers

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

DistributionContainerPropertiesRequired
distributionContainerPropertiesRequired (m)

The properties that are required when getting a distribution container from the list.

Return Values

xsd:long openListHandle

A numeric handle to a list of distribution container that can be passed to GetDistributionContainers.

0xsd:long totalNoOfDistributionContainers

The total number of distribution containers in the list whose handle has been returned.

Remarks

None

More information:

[DistributionContainerPropertiesRequired](#) (see page 692)

OpenDistributionContainerOrderList

This OpenDistributionContainerOrderList method opens a list of DistributionContainerOrders for the given DistributionContainer and returns listHandle and totalNoOfDistributionContainerOrders.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string distributionContainerId (m)

The UUID of the distribution container whose orders are to be listed.

**DistributionContainerOrderPropertiesRequired
distributionContainerOrderPropertiesRequired (m)**

The required DistributionContainer Order Properties.

Return Values

xsd:long openListHandle

A numeric handle to a list of distribution container orders that can be passed to GetDistributionContainerOrders.

xsd:long totalNumberOfDistributionContainerOrder

The total number of Distribution Container Orders.

Remarks

None

More information:

[DistributionContainerOrderPropertiesRequired](#) (see page 692)

RenewDeliveryDistributionContainers

The RenewDeliveryDistributionContainers method renews one or more halted or failed distributions.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfDeliveryDistributionContainerProperties

arrayOfDeliveryDistributionContainerProperties (m)

An array of DeliveryDistributionContainerProperties to be renewed.

Return Values

None

ResumeDeliveryDistributionContainers

The ResumeDeliveryDistributionContainers method resumes one or more halted distributions

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfDeliveryDistributionContainerProperties

arrayOfDeliveryDistributionContainerProperties (m)

An array of DeliveryDistributionContainerProperties to be resumed.

Return Values

None

Remarks

The distribution can only be resumed when distribution status is "Halted" or "Error" and you are using DTS (Data Transport Service). You can set a new date and time for the distribution to be resumed.

The difference between Resume and Renew is that when you use Renew, the distribution starts all over again for areas, where status is not "OK".

With Resume, the distribution will either continue with the first parcel transfer that did not yet succeed or (if 100% has been sent) continue waiting for outstanding acknowledgements. The successful parcel transfers so far made in the distribution will not be repeated.

SendDistributionContainer

This method sends associated orders to specified domains and domainGroups.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container.

DateTime distributionDateTime

Specifies the time to send the associated distribution to the addressed domain managers. If not supplied, the default value is the current date and time of the enterprise manager computer ('now').

DateTime haltDateTime

Specifies the date and time when a distribution order should be halted if still ongoing. If not supplied comstore parameter DistMaxDistTime is used.

xsd:Boolean: toAllDomains

If coded the distribution will be sent to all CA ITCM domains registered at the enterprise manager

ArrayOfstring domainIds

Specifies the name of a CA ITCM domain where the distribution is to be sent.

ArrayOfstring domainGroupIds

Specifies the name of a CA ITCM domain group where the distribution is to be sent.

Return Values

None

Remarks

One or more of `addAllDomains`, `domainIds`, and `domainGroupIds` must be supplied. If `addAllDomains` is set to 'true', `domainIds` and `domainGroupIds` are ignored.

SetDistributionContainer

The `SetDistributionContainer` method updates the properties of a distribution container.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string distributionContainerId (m)

UUID of the distribution container to be updated.

SetDistributionContainerProperties setDistributionContainerProperties (m)

The properties that should be updated for the specified distribution container.

Return Values

None

Remarks

None

More information:

[SetDistributionContainerProperties](#) (see page 751)

Enterprise Manager Domain and Domain Group Functionalities

This section contains the following topics:

[CloseDomainGroupDomainMembersList](#) (see page 597)
[CloseDomainGroupDomainGroupMembersList](#) (see page 598)
[CreateDomainGroup](#) (see page 599)
[DeleteDomainGroups](#) (see page 599)
[FindDomain](#) (see page 600)
[FindDomainGroup](#) (see page 601)
[GetDomain](#) (see page 602)
[GetDomainGroup](#) (see page 602)
[GetDomainGroupDomainGroupMembers](#) (see page 603)
[GetDomainGroupDomainMembers](#) (see page 603)
[LinkDomainGroupsToDomainGroup](#) (see page 604)
[LinkDomainsToDomainGroup](#) (see page 605)
[OpenDomainGroupDomainGroupMembersList](#) (see page 605)
[OpenDomainGroupDomainGroupsList](#) (see page 606)
[OpenDomainGroupDomainMembersList](#) (see page 607)
[SetDomainGroup](#) (see page 608)
[UnlinkDomainGroupsFromDomainGroup](#) (see page 609)
[UnlinkDomainsFromDomainGroup](#) (see page 610)

CloseDomainGroupDomainMembersList

The CloseDomainGroupDomainMembersList method closes a domain group list that was opened by OpenDomainGroupDomainMembersList and releases the list handle.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:long listHandle (m)

A numeric handle to the list to close. The handle was generated by a previous call to OpenDomainGroupDomainMembersList.

Return Values

This is a void method that does not expose a return value.

Remarks

None

CloseDomainGroupDomainGroupMembersList

The CloseDomainGroupDomainGroupMembersList method closes a domain group list that was opened by OpenDomainGroupDomainGroupMembersList and releases the list handle

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to the list to close. The handle was generated by a previous call to OpenDomainGroupDomainMembersList.

Return Values

This is a void method that does not expose a return value

Remarks

None

CreateDomainGroup

The CreateDomainGroup method creates a domain group.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string parentDomainGroupid

The UUID identifying the parent group of which the newly created group is a child. In order to create a top-level group, this is the UUID of the Domains group.

If this parameter is not supplied then the domain group will be created under “domains”.

CreateDomainGroupProperties createDomainGroupProperties (m)

The properties required to create a domain group.

Return Values

xsd: string DomainGroupID

If a domain group was successfully created, the UUID of the new group is returned.

Remarks

None

More information:

[CreateDomainGroupProperties](#) (see page 632)

DeleteDomainGroups

The DeleteDomainGroups method deletes one or more groups identified by their UUIDs.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

ArrayOfstring domainGroupIds (m)

An array of UUIDs that identify the groups to delete.

Return Values

None

Remarks

None

FindDomain

The FindDomain method finds a named domain unit.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainName (m)

The domain name to search for. This may contain wildcard characters (*).

Xsd:long noOfDomains

The number of domains that should be returned from the search if more than one match is found.

DomainPropertiesRequired domainPropertiesRequired

This parameter specifies which domain properties to retrieve. Mandatory if noOfDomains is greater than 0.

Return Values

ArrayOfDomainProperties arrayOfDomainProperties

An array of domain properties structures, one for each domain matched by the search. This array will either be of the size of the number of domains matched by the search, or of the number of domains requested by the method call - whichever size is the lowest.

xsd:long totalNoOfDomainsFound

The total number of domains that were matched by the search. You can use this to indicate that there were more matches found than the requested number of domains.

Remarks

The method supports wildcard pattern matching for the domain name. The name is case insensitive.

More information:

[DomainPropertiesRequired](#) (see page 748)

FindDomainGroup

The FindDomainGroup method finds a domain group unit given its name.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupName (m)

The domain group name to search for. This may contain wildcard characters (*).

Xsd:long noOfDomainGroups

The number of domain groups that should be returned from the search if more than one match is found.

DomainGroupPropertiesRequired domainGroupPropertiesRequired

Specifies which domain group properties to retrieve. Mandatory if noOfDomains is greater than 0.

Return Values

ArrayOfDomainGroupProperties arrayOfDomainGroupProperties

An array of domain group properties structures, one for each domain group matched by the search. This array will either be of the size of the number of domain groups matched by the search, or of the number of domain groups requested by the method call - whichever size is the lowest.

xsd:long totalNoOfDomainGroupsFound

The total number of domain groups that were matched by the search. You can use this to indicate that there were more matches found than the requested number of domain groups.

Remarks

The method supports wildcard pattern matching for the domain group name. The name is case insensitive.

More information:

[DomainGroupPropertiesRequired](#) (see page 704)

GetDomain

The GetDomain method retrieves a domain identified by its UUID.

Parameters

xsd:string sessionId (m)

Specifies the session ID obtained from the login to the web service.

xsd:string domainUUID (m)

The UUID of the domain to retrieve.

DomainPropertiesRequired domainPropertiesRequired (m)

Specifies the domain properties to retrieve.

Return Values

DomainProperties domainProperties

The domain properties retrieved.

Remarks

None

More information:

[DomainPropertiesRequired](#) (see page 748)

GetDomainGroup

The GetDomainGroup method retrieves a domain group identified by its UUID.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupId (m)

The UUID of the domain group to retrieve.

DomainGroupPropertiesRequired domainGroupPropertiesRequired (m)

Specifies which domain group properties to retrieve.

Return Values

DomainGroupProperties domainGroupProperties

Remarks

None

More information:

[DomainPropertiesRequired](#) (see page 748)

GetDomainGroupDomainGroupMembers

The GetDomainGroupDomainGroupMembers method retrieves the domain groups from a domain group's member list.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to the list to close. The handle was generated by a previous call to OpenDomainGroupDomainGroupMembersList.

xsd:long fromIndex

Index from where to list the domain groups. This is a reserved parameter and is ignored by the webservice.

xsd:long noOfDomainGroups (m)

The total number of groups to get from the list. This is less than or equal to the total number in the list returned by a previous call to OpenDomainGroupDomainGroupMembersList.

Return Values

dsm: ArrayofDomainGroupProperties

Remarks

None

GetDomainGroupDomainMembers

The GetDomainGroupDomainMembers method retrieves the domains from a domain group's member list.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:long listHandle (m)

A numeric handle to a list returned by a previous call to OpenDomainGroupDomainMembersList.

xsd:long fromIndex

Index from where to list the domains.

xsd:long noOfDomains (m)

The total number of domains to get from the list. This is less than or equal to the total number in the list returned by a previous call to `OpenDomainGroupDomainMembersList`.

Return Values

`ArrayOfDomainProperties`

Remarks

None

LinkDomainGroupsToDomainGroup

The `LinkDomainGroupsToDomainGroup` method links one or more domain groups to a domain group

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupId

The UUID identifying the domain group to which the given groups are to be linked. The default value is the UUID of the system group "domains".

ArrayOfstring domainGroupIds (m)

An array of UUIDs identifying the groups that are to be linked to the specified domain group.

Return Values

None

Remarks

None

LinkDomainsToDomainGroup

The LinkDomainsToDomainGroup method links one or more domains to a given domain group

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupId (m)

The UUID identifying the domain group to which the domains are to be linked.

ArrayOfstring domainIds (m)

An array of UUIDs identifying the domains to link to the specified domain group.

Return Values

None

Remarks

None

OpenDomainGroupDomainGroupMembersList

The OpenDomainGroupDomainGroupMembersList method opens a list of the domain groups that are members of the given domain group.

The method returns the total number of groups in the list, and a handle to the list that you can use for further calls to GetDomainGroupDomainGroupMembers

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupId

The UUID of the domain group whose member groups are to be listed. The default value is the UUID of the system group "domains" .

DomainGroupPropertiesRequired domainGroupPropertiesRequired (m)

The properties that are required when getting a group from the list.

Return Values

xsd:long OpenListHandle

A numeric handle to a list of domain groups that can be passed to `GetDomainGroupDomainGroupMembers`

xsd:long totalNoOfDomainGroups

The total number of groups in the list whose handle has been returned.

Remarks

None

More information:

[DomainGroupPropertiesRequired](#) (see page 704)

OpenDomainGroupDomainGroupsList

The `OpenDomainGroupDomainMembersList` method opens a list of the domains that are members of the given domain group.

The method returns the total number of domains in the list, and a handle to the list that can be used for further calls to `GetDomainGroupDomainMembers`.

Parameters

xsd:string sessionId (m)

The session identifier.

xsd:string domainGroupId (m)

The UUID of the domain group whose member domains are to be listed.

Note: If this parameter is not supplied then the system group “All domains” UUID will be taken.

ArrayOfDomainPropertyFilter domainPropertyFilters

An array of domain property filter definitions to limit the domains returned by the search.

DomainPropertiesRequired domainPropertiesRequired (m)

The properties that will be required when getting a domain from the list.

Return Values

xsd:long openListHandle

A numeric handle to a list of domains that can be passed to GetDomainGroupDomainMembers

xsd:long totalNoOfDomains

The total number of domains in the list whose handle has been returned.

Remarks

None

More information:

[DomainPropertiesRequired](#) (see page 748)

More Information:

[DomainPropertiesRequired](#) (see page 748)

OpenDomainGroupDomainMembersList

The OpenDomainGroupDomainMembersList method opens a list of the domains that are members of the given domain group.

The method returns the total number of domains in the list, and a handle to the list that you can use for further calls to GetDomainGroupDomainMembers.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupId

The UUID of the domain group whose member domains are to be listed.

Note: If this parameter is not supplied then the system group “All domains” UUID will be taken.

ArrayOfDomainPropertyFilter domainPropertyFilters

An array of domain property filter definitions to limit the domains returned by the search.

DomainPropertiesRequired domainPropertiesRequired (m)

The properties that are required when getting a domain from the list.

Return Values

xsd:long openListHandle

A numeric handle to a list of domains that can be passed to GetDomainGroupDomainMembers

xsd:long totalNoOfDomains

The total number of domains in the list whose handle has been returned.

Remarks

None

SetDomainGroup

The SetDomainGroup method updates the properties of a domain group.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string groupUUID (m)

Specifies the domain group UUID.

SetDomainGroupProperties setDomainGroupProperties (m)

The properties that are updated for the specified domain group.

Return Values

None

Remarks

None

More information:

[SetDomainGroupProperties](#) (see page 731)

UnlinkDomainGroupsFromDomainGroup

The UnlinkDomainGroupsFromDomainGroup method unlinks one or more domain groups from the specified domain group.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupId

The UUID identifying the domain group from which the given groups are to be unlinked.

If this parameter is not supplied then the system group “domains” UUID will be taken.

ArrayOfstring domainGroupIds (m)

An array of UUIDs identifying the groups that are to be unlinked from the specified domain group.

Return Values

None

Remarks

None

UnlinkDomainsFromDomainGroup

The UnlinkDomainsFromDomainGroup method unlinks one or more domains from the specified domain group.

Parameters

xsd:string sessionId (m)

Specifies the session id obtained from the login to the web service.

xsd:string domainGroupId (m)

The UUID identifying the group from which the domains are to be unlinked.

ArrayOfstring domainIds (m)

An array of UUIDs identifying the domains that are to be unlinked from the specified group.

Return Values

None

Remarks

None

Asset Script Job Methods

More Information:

[AssetScriptJobProperties](#) (see page 788)

CreateAssetScriptJob

The CreateAssetScriptJob method is used to create an asset script job.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

dsm:AssetScriptJobProperties assetJobScriptProperties

Specifies the properties required to create an asset script job.

Return Values

dsm:CreateAssetScriptJobResponse

Returns a response with the job ID, UUID, and domain ID if the asset script job is created successfully.

Remarks

None

More Information:

[AssetScriptJobPropertiesRequired](#) (see page 796)

GetAssetScriptJob

The GetAssetScriptJob method is used to get the properties of an asset script job.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:long assetjobUID

Specifies the UUID identifying the script job, the properties of which must be retrieved.

**dsm:AssetScriptJobPropertiesRequired
assetScriptJobPropertiesRequired**

Specifies the list of script job properties to be retrieved.

Return Values

dsm:GetAssetScriptJobResponse

Returns a response with the job ID, domain ID, and the value of each property passed in the input parameter AssetScriptJobPropertiesRequired.

Remarks

None

More Information:

[AssetScriptJobPropertiesRequired](#) (see page 796)

OpenAssetScriptJobList

The OpenAssetScriptJobList is used to open the asset script jobs list.

Parameters

xsd:string sessionID

Specifies the session id obtained from the login to the web service.

dsm:AssetScriptJobPropertiesRequired assetJobPropertiesRequired

Specifies the list of script job properties to be retrieved.

Return Values

dsm:OpenAssetScriptJobListResponse

Returns the list handle to the list opened and the number of jobs in the list.

Remarks

None

More Information:

[ArrayOfGetAssetScriptJobsProperties](#) (see page 993)

GetAssetScriptJobs

The GetAssetScriptJobs is used to get the properties for all the asset script jobs.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

Xsd:long listHandle

Specifies the handle to the asset script job list returned by OpenAssetScriptJobList method.

Xsd:long noOfJobs

Specifies the number of asset jobs returned by OpenAssetJobList method.

Return Values

ArrayOfGetAssetScriptJobsProperties getAssetJobsResponse

Contains the job properties of all the script jobs.

Remarks

None

CloseAssetScriptJobList

The CloseAssetScriptJobList method is used to close the list handle of asset script job list.

Parameters

xsd:string sessionID

Specifies the session id obtained from the login to the web service.

Xsd:long listHandle

Specifies the handle to the asset job list returned by OpenAssetScriptJobList method.

Return Values

None

Remarks

None

More Information:

[AssetScriptJobProperties](#) (see page 788)

SetAssetScriptJob

The SetAssetScriptJob is used to set the properties for an asset script job.

Parameters

xsd:string sessionID

Specifies the session id obtained from the login to the web service.

xsd:long assetjobUID

Specifies the UUID identifying the script job, the properties of which must be retrieved.

dsm:AssetScriptJobProperties assetScriptJobProperties

Specifies asset job properties you want to set.

Return Values

None

Remarks

None

DeleteAssetJobs

The DeleteAssetJobs method is used to delete list of asset jobs.

Parameters

xsd:string sessionID

Specifies the session id obtained from the login to the web service.

dsm:ArrayOfstring assetJobIds

Specifies the array of asset job IDs to delete.

Return Values

None

Remarks

None

LinkAssetJobsToUnit

The LinkAssetJobsToUnit method is used to link a list of asset jobs to a unit.

Parameters

xsd:string sessionID

Specifies the session id obtained from the login to the web service.

xsd:string unitId

Specifies the unit ID to which you want to link the asset jobs.

dsm:UnitObjectType unitObjectType

Specifies the object type of the unit.

dsm:ArrayOfstring *arrayofassetJobIds

Specifies the array of asset job IDs you want to link with unit.

Return Values

None

Remarks

None

More information:

[UnitObjectType](#) (see page 110)

LinkAssetJobsToUnitGroup

The LinkAssetJobsToUnitGroup method is used to link a list of asset jobs to a unit group.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupId

Specifies the unit group ID to which you want to link the asset jobs.

dsm:ArrayOfstring assetJobIds

Specifies the array of asset job IDs you want to link with the unit group.

Return Values

None

Remarks

None

More Information:

[UnitObjectType](#) (see page 110)

UnLinkAssetJobsFromUnit

The UnLinkAssetJobsFromUnit method is used to unlink a list of asset jobs from a unit.

Parameters

xsd:string sessionId

Specifies the session id obtained from the login to the web service.

xsd:string unitId

Specifies the unit ID from which you want to unlink the asset jobs.

dsm:UnitObjectType unitObjectType

Specifies the type of the unit.

dsm:ArrayOfstring assetJobIds

Specifies the array of asset job IDs you want to unlink from the unit.

Return Values

None

Remarks

None

UnLinkAssetJobsFromUnitGroup

The UnLinkAssetJobsFromUnitGroup method is used to unlink a list of asset jobs from a unit group.

Parameters**xsd:string sessionId**

Specifies the session id obtained from the login to the web service.

xsd:string unitGroupId

Specifies the unit group ID from which you want to unlink the asset jobs.

dsm:ArrayOfstring assetJobIds

Specifies the array of asset job IDs you want to unlink from the unit group.

Return Values

None

Remarks

None

ActivateJobCheck

The ActivateJobCheck method is used to trigger the asset or software job check.

Parameters**xsd:string sessionId**

Specifies the session id obtained from the login to the web service.

dsm:ArrayOfTriggerInfo triggInfoList

Specifies the list of objects to trigger.

dsm:ArrayOfAMExtraCmdParam cmdParams

Specifies the additional parameters for executing asset management job check.

Return Values

None

Remarks

None

Sequences

The following sections describe all the sequences that the CA ITCM Web Services uses. These structures are a mapping of the underlying layer functionality.

This section contains the following topics:

[ComputerProperties2](#) (see page 626)
[QueryFilter](#) (see page 629)
[QueryProperties2](#) (see page 629)
[QueryProperty](#) (see page 630)
[QueryFolderProperties](#) (see page 630)
[HWModelFilter](#) (see page 630)
[AttributeMask](#) (see page 631)
[AuthenticationData](#) (see page 631)
[CreateDistributionContainerProperties](#) (see page 631)
[CreateDomainGroupProperties](#) (see page 632)
[CreateInstallSoftwareJobOrderPolicyProperties](#) (see page 632)
[CreateInstallSoftwareJobOrderPolicyProperties3](#) (see page 634)
[CreateInstallSoftwareJobOrderProperties](#) (see page 636)
[CreateInstallSoftwareJobOrderProperties3](#) (see page 637)
[CreateQueryProperties](#) (see page 640)
[CreateQueryProperties2](#) (see page 641)
[CreateSecurityAreaProperties](#) (see page 641)
[CreateSecurityProfileProperties](#) (see page 642)
[CreateSoftwareGroupProperties](#) (see page 643)
[CreateSoftwareJobContainerProperties](#) (see page 643)
[CreateSoftwarePackageProcedureDependencyProperties](#) (see page 644)
[CreateSoftwarePackageProcedureGroupProperties](#) (see page 644)
[CreateSoftwarePackageProcedureProperties](#) (see page 644)
[CreateSoftwarePackageProcedureProperties2](#) (see page 646)
[CreateSoftwarePackageProcedureProperties3](#) (see page 650)
[CommonSoftwarePackageProcedureProperties3](#) (see page 651)
[SoftwarePackageProcedureOperationMask3](#) (see page 654)
[computerClassID](#) (see page 655)
[ComputerProperties](#) (see page 669)
[ComputerPropertiesRequired](#) (see page 670)
[ComputerPropertyFilter](#) (see page 671)
[ComputerStatusInfoRequired](#) (see page 673)
[CreateAgentComponents](#) (see page 673)
[CreateAgentComponents](#) (see page 674)
[CreateCatalogSoftwarePackageGroupProperties](#) (see page 674)
[CreateComputerProperties](#) (see page 675)
[CreateComputerProperties2](#) (see page 678)
[CreateSoftwareJobContainerProperties2](#) (see page 679)
[CreateSoftwareJobContainerProperties3](#) (see page 680)
[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)
[CreateSoftwareJobOrderPolicyProperties3](#) (see page 683)
[CreateSoftwareJobOrderProperties](#) (see page 685)
[CreateSoftwareJobOrderProperties3](#) (see page 687)
[CreateSoftwarePackageGroupProperties](#) (see page 689)

[CreateSoftwarePackageProperties](#) (see page 690)
[CreateSoftwarePackageVolumeProperties](#) (see page 690)
[DistributionContainerProperties](#) (see page 691)
[DistributionContainerPropertiesRequired](#) (see page 692)
[DistributionPermissionMask](#) (see page 692)
[DistributionContainerOrderPropertiesRequired](#) (see page 692)
[CreateSoftwarePolicyProperties2](#) (see page 693)
[CreateUnitGroupPropertiesByName](#) (see page 695)
[CreateUserProfileProperties](#) (see page 696)
[DateTime](#) (see page 697)
[DeploymentPackageProperties](#) (see page 697)
[DeploymentPackagePropertiesRequired](#) (see page 698)
[DeploymentInfo](#) (see page 699)
[DeploymentPackageInfo](#) (see page 700)
[DiscoveredSoftwareProperties](#) (see page 700)
[DiscoveredSoftwarePropertiesRequired](#) (see page 701)
[DistributionContainerOrderProperties](#) (see page 701)
[DistributionContainerOrderProperties3](#) (see page 702)
[DistributionDomainProperties](#) (see page 702)
[DistributionDomainPropertiesRequired](#) (see page 703)
[DomainGroupProperties](#) (see page 704)
[DomainGroupPropertiesRequired](#) (see page 704)
[DomainProperties](#) (see page 705)
[MachineCredentials](#) (see page 706)
[NoOfDistributionDomains](#) (see page 707)
[ParameterInfo](#) (see page 707)
[ParameterValue](#) (see page 708)
[QueryFragment](#) (see page 708)
[QueryResultProperties](#) (see page 708)
[QueryResultPropertiesRequired](#) (see page 709)
[ResumeDeliveryDistributionContainerProperties](#) (see page 709)
[ExportSoftwarePkgProperties](#) (see page 709)
[FetchItemJobOrderProperties](#) (see page 710)
[FetchItemOrderProperties](#) (see page 710)
[FileSystemElementPath](#) (see page 710)
[GetSecurityAreaConfiguration](#) (see page 710)
[GetSecurityGroupPermissions](#) (see page 711)
[GetSecurityObjectsPermissions](#) (see page 711)
[ImportSoftwarePackageProperties](#) (see page 711)
[ImportSoftwarePackageProperties2](#) (see page 712)
[ImportSoftwarePackagePropertiesByName](#) (see page 713)
[QueryFragment2](#) (see page 714)
[QueryProperties](#) (see page 716)
[QueryProperties3](#) (see page 717)
[QueryPropertiesRequired](#) (see page 718)
[QueryPropertiesRequired3](#) (see page 718)
[ScanPackageInfo](#) (see page 719)
[ScanParameters](#) (see page 719)
[SDAgentProperties](#) (see page 721)
[SecurityAreaProperties](#) (see page 722)
[SecurityAreaPropertiesRequired](#) (see page 722)
[SecurityClassPermissions](#) (see page 722)

[SecurityClassPermissionsForSecurityProfile](#) (see page 723)
[SecurityClassPermissionProperties](#) (see page 723)
[SecurityClassProperties](#) (see page 723)
[SecurityClassPropertiesRequired](#) (see page 723)
[SecurityGroupPermissionsForAllSecurityProfiles](#) (see page 724)
[SecurityGroupProfilePermission](#) (see page 724)
[SecurityObjectLinkageProperties](#) (see page 724)
[SecurityObjectProfilePermission](#) (see page 724)
[SecurityObjectSecurityAreaLinkages](#) (see page 725)
[SecurityObjectOwner](#) (see page 725)
[SecurityObjectPermissions](#) (see page 725)
[SecurityAreaByName](#) (see page 725)
[SecurityAreaConfiguration](#) (see page 726)
[SecurityAreaConfigurationByName](#) (see page 726)
[SecurityObjectPermissionsForAllSecurityProfiles](#) (see page 726)
[SecurityObjectsSecurityAreaProperties](#) (see page 726)
[SecurityProfileProperties](#) (see page 726)
[SecurityProfilePropertiesRequired](#) (see page 727)
[SetSecurityProfileProperties](#) (see page 727)
[SetSecurityGroupPermissions](#) (see page 728)
[SetSecurityObjectPermissions](#) (see page 728)
[SecurityProfileSpecialPermissionFlags](#) (see page 728)
[SetCatalogSoftwarePackageGroupProperties](#) (see page 729)
[SetComputerProperties](#) (see page 730)
[SetDomainGroupProperties](#) (see page 731)
[SetSecurityClassProperties](#) (see page 731)
[SetSoftwareJobContainerProperties2](#) (see page 731)
[SetSoftwarePolicyProperties2](#) (see page 732)
[SetSoftwareGroupProperties](#) (see page 734)
[SetQueryProperties](#) (see page 734)
[SetQueryProperties3](#) (see page 735)
[SetSoftwarePackageProcedureGroupProperties](#) (see page 735)
[SetSoftwarePackageProcedureProperties3](#) (see page 736)
[SoftwarePackageProcedureProperties3](#) (see page 737)
[SetSoftwareJobContainerProperties](#) (see page 739)
[SetSoftwareJobContainerProperties2](#) (see page 739)
[SetSoftwareJobContainerProperties3](#) (see page 740)
[SoftwareJobContainerPropertiesRequired2](#) (see page 741)
[SetSoftwareJobProperties](#) (see page 742)
[SetSoftwareJobProperties2](#) (see page 743)
[SetSoftwareJobProperties3](#) (see page 744)
[Structure CreateUnitGroupProperties2 CreateUnitGroupProperties2](#) (see page 746)
[CreateSoftwareJobLinkage](#) (see page 746)
[DomainPropertyFilter](#) (see page 747)
[DomainPropertiesRequired](#) (see page 748)
[ElementTypeMask](#) (see page 750)
[EvaluationTypeMask](#) (see page 750)
[SecurityArea](#) (see page 750)
[SetDistributionContainerProperties](#) (see page 751)
[SetSecurityAreaProperties](#) (see page 751)
[SetSoftwarePackageGroupProperties](#) (see page 751)

[SetSoftwarePackageProcedureProperties](#) (see page 752)
[SetSoftwarePackageProcedureProperties2](#) (see page 754)
[SetSoftwarePackageProperties](#) (see page 757)
[SetSoftwarePackageVolumeProperties](#) (see page 757)
[SetSoftwarePolicyProperties](#) (see page 758)
[SetSoftwarePolicyProperties3](#) (see page 759)
[SetSoftwarePolicyPropertiesByName](#) (see page 761)
[SetSoftwarePolicyPropertiesByName3](#) (see page 763)
[SetTimeoutProperties](#) (see page 766)
[SetUnitGroupProperties](#) (see page 767)
[SetUserProfileProperties](#) (see page 767)
[SoftwareJobPermissionMask](#) (see page 768)
[SoftwareJobProperties](#) (see page 768)
[SoftwareJobProperties2](#) (see page 771)
[SoftwareJobProperties3](#) (see page 774)
[SoftwareJobPropertiesRequired](#) (see page 777)
[SoftwareJobStatusCounters](#) (see page 778)
[SoftwarePackageGroupPropertiesRequired](#) (see page 778)
[SoftwarePackageMask](#) (see page 779)
[SoftwarePackageMaskType](#) (see page 779)
[CreateSoftwarePolicyProperties](#) (see page 779)
[CreateSoftwarePolicyProperties3](#) (see page 781)
[CreateUnitGroupProperties](#) (see page 783)
[SoftwareJobBootMask \(bootAfterMask\)](#) (see page 784)
[SoftwareJobBootMask3 \(bootAfterMask\)](#) (see page 784)
[SoftwareJobBootMask \(bootBeforeMask\)](#) (see page 785)
[SoftwareJobBootMask3 \(bootBeforeMask\)](#) (see page 786)
[SoftwareJobContainerProperties](#) (see page 786)
[CreateAssetScriptJobResponse](#) (see page 788)
[AssetScriptJobProperties](#) (see page 788)
[SchedulingOptions](#) (see page 789)
[Scheduling](#) (see page 789)
[Conditions](#) (see page 790)
[Dependency](#) (see page 790)
[PreJob](#) (see page 791)
[PostJob](#) (see page 792)
[Miscellaneous](#) (see page 792)
[DateTimeRestrictions](#) (see page 792)
[FileDependency](#) (see page 793)
[JobDependency](#) (see page 793)
[PeriodicDependency](#) (see page 794)
[JobExecProperties](#) (see page 794)
[NetWorkDrive](#) (see page 795)
[PostJobProperties](#) (see page 796)
[AssetScriptJobPropertiesRequired](#) (see page 796)
[Dsm:OpenAssetScriptJobListResponse](#) (see page 796)
[GetAssetScriptJobResponse](#) (see page 797)
[DayOfTheWeekDependency](#) (see page 797)
[GetAssetScriptJobsProperties](#) (see page 797)
[TriggerInfo](#) (see page 798)
[OSIMProperties](#) (see page 798)
[OSIMPropertiesWithDesc](#) (see page 799)

[OSIMparameterDesc](#) (see page 799)
[ScalabilityServerProperties](#) (see page 799)
[ScalabilityServerPropertiesRequired](#) (see page 800)
[OSImageFilter](#) (see page 801)
[OSImageProperties](#) (see page 802)
[OSIMTargetFilter](#) (see page 802)
[OSIMTargetProperties](#) (see page 803)
[ComputerPropertyFilter2](#) (see page 803)
[ComputerPropertiesRequired2](#) (see page 806)
[DateTime2](#) (see page 809)
[ComputerProperties2](#) (see page 809)
[CreateOSIMComputerProperties](#) (see page 812)
[OSImageProperties](#) (see page 815)
[SoftwareJobContainerProperties3](#) (see page 816)
[SoftwareJobContainerPropertiesRequired](#) (see page 818)
[SoftwareJobContainerPropertiesRequired3](#) (see page 819)
[SoftwareJobContainerPropertyMask](#) (see page 820)
[SoftwareJobOperationMask](#) (see page 821)
[SoftwareJobOperationMask2](#) (see page 822)
[SoftwareJobOrderProperties](#) (see page 823)
[SoftwareJobOrderProperties3](#) (see page 825)
[SoftwarePackageGroupProperties](#) (see page 828)
[SoftwarePackageProcedureBootMask \(bootAfterMask\)](#) (see page 829)
[SoftwarePackageProcedureBootMask2 \(bootAfterMask\)](#) (see page 830)
[SoftwarePackageProcedureBootMask](#) (see page 830)
[SoftwarePackageProcedureBootMask2 \(bootBeforeMask\)](#) (see page 831)
[FindSoftwarePackageProcedure](#) (see page 832)
[SoftwarePackageProcedurePropertiesRequired](#) (see page 833)
[SoftwarePackageProcedureTypeMask](#) (see page 834)
[SoftwarePackagePropertiesRequired](#) (see page 835)
[SoftwarePackageVolumeElementPropertiesRequired](#) (see page 836)
[SoftwarePackageVolumeProperties](#) (see page 836)
[SoftwarePackageVolumePropertiesRequired](#) (see page 837)
[SoftwarePackageProcedureDependencyProperties](#) (see page 837)
[SoftwarePackageProcedureDependencyPropertiesRequired](#) (see page 838)
[SoftwarePackageProcedureGroupProperties](#) (see page 838)
[SoftwarePackageProcedureGroupPropertiesRequired](#) (see page 839)
[SoftwarePackageProcedureManagementTypeMask](#) (see page 840)
[SoftwarePackageProcedureOperationMask](#) (see page 841)
[SoftwarePackageProcedureOperationMask2](#) (see page 842)
[SoftwarePackageProcedureProperties](#) (see page 843)
[SoftwarePackageProcedureProperties2](#) (see page 845)
[SoftwarePackageProperties](#) (see page 847)
[SoftwarePackageProperties2](#) (see page 849)
[SoftwarePackageVolumeElementProperties](#) (see page 850)
[SoftwarePolicyExceptionProperties](#) (see page 851)
[SoftwarePolicyProperties](#) (see page 851)
[SoftwarePolicyProperties3](#) (see page 854)
[SoftwarePolicyPropertiesRequired](#) (see page 856)
[SoftwarePolicyPropertiesRequired2](#) (see page 857)
[SoftwarePolicyPropertiesRequired3](#) (see page 858)

[SoftwareJobContainerProperties2](#) (see page 860)
[SoftwareJobContainerPropertiesRequired2](#) (see page 862)
[SoftwarePolicyState](#) (see page 862)
[SoftwareJobContainerOrder](#) (see page 863)
[SoftwarePolicyOrder](#) (see page 863)
[SoftwarePackageProcedureOrder](#) (see page 863)
[SoftwareByName](#) (see page 864)
[SoftwareJobLinkage](#) (see page 864)
[SoftwareJobLinkage](#) (see page 864)
[SoftwarePackage](#) (see page 865)
[SoftwarePackageOrder](#) (see page 865)
[SoftwarePackageProcedure](#) (see page 865)
[SoftwarePolicyProperties2](#) (see page 865)
[SSPackageInfo](#) (see page 867)
[SSPackageInfoRequired](#) (see page 867)
[StagingCredentials](#) (see page 868)
[UnitInventoryComponentProperties](#) (see page 868)
[UnitInventoryComponentPropertiesRequired](#) (see page 868)
[UnitInventoryGroupProperties](#) (see page 869)
[UnitInventoryGroupPropertiesRequired](#) (see page 869)
[UnitInventoryItemProperties](#) (see page 869)
[UnitInventoryItemPropertiesRequired](#) (see page 870)
[UnitGroupProperties](#) (see page 870)
[SetUnitGroupProperties2](#) (see page 874)
[UnitGroupPropertiesForSoftwarePolicy](#) (see page 874)
[UnitGroupPropertiesRequired](#) (see page 876)
[UnitProperties](#) (see page 877)
[UnitPropertiesRequired](#) (see page 877)
[UnitPropertyFilter](#) (see page 877)
[UnitQueryPropertiesRequired](#) (see page 879)
[UnitQueryProperties](#) (see page 880)
[UnitSoftwareJobProperties](#) (see page 880)
[UnitSoftwareJobProperties2](#) (see page 883)
[UnitSoftwareJobPropertiesRequired](#) (see page 886)
[UnitSoftwareJobTaskMask](#) (see page 887)
[UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroupByName](#) (see page 887)
[UsageListMask](#) (see page 889)
[UserAccountProperties](#) (see page 890)
[UserAccountPropertiesRequired](#) (see page 891)
[UserAccountPropertyFilter](#) (see page 891)
[UserProfileProperties](#) (see page 892)
[UserProfilePropertiesRequired](#) (see page 896)
[UserProfilePropertyFilter](#) (see page 897)
[VolumeElementAttributes](#) (see page 899)
[CreateSoftwarePackageProperties3](#) (see page 900)
[SetSoftwarePackageProperties3](#) (see page 901)
[SoftwarePackageProperties3](#) (see page 902)
[SoftwarePackagePropertiesRequired3](#) (see page 906)
[SoftwarePackageProcedureProperties4](#) (see page 907)
[SoftwarePackageProcedurePropertiesRequired2](#) (see page 910)
[UnitSoftwareJobProperties3](#) (see page 911)

[WSChallengeResponse](#) (see page 914)
[WSAuthenticateResponse](#) (see page 914)
[Item](#) (see page 915)
[UnitGroupProperties2](#) (see page 915)
[CreateUnitGroupProperties3](#) (see page 917)
[createAssessmentTemplate](#) (see page 918)
[updateAssessmentTemplate](#) (see page 918)
[deleteAssessmentTemplates](#) (see page 919)
[getAssessmentTemplate](#) (see page 919)
[getAssessmentTemplateList](#) (see page 920)
[getAssessmentTemplateListDetailed](#) (see page 921)
[createAssessmentTask](#) (see page 922)
[executeAssessmentTask](#) (see page 922)
[getAssessmentStatus](#) (see page 923)
[getAssessmentResult](#) (see page 923)
[getAssessmentTask](#) (see page 924)
[getAssessmentTaskList](#) (see page 924)
[deleteAssessmentTasks](#) (see page 925)
[createOSIMTemplate](#) (see page 926)
[updateOSIMTemplate](#) (see page 926)
[deleteOSIMTemplates](#) (see page 926)
[getOSIMTemplate](#) (see page 927)
[getOSIMTemplateList](#) (see page 927)
[getOsimTemplateListDetailed](#) (see page 928)
[createSoftwareDeploymentTemplate](#) (see page 929)
[updateSoftwareDeploymentTemplate](#) (see page 930)
[deleteSoftwareDeploymentTemplates](#) (see page 930)
[getSoftwareDeploymentTemplate](#) (see page 931)
[getSoftwareDeploymentTemplateList](#) (see page 931)
[getSoftwareDeploymentTemplateListDetailed](#) (see page 932)
[createAutomationProcess](#) (see page 933)
[deleteAutomationProcesses](#) (see page 934)
[executeAutomationProcess](#) (see page 934)
[getAutomationProcessList](#) (see page 935)
[getAutomationProcessListDetailed](#) (see page 936)
[getAutomationResult](#) (see page 937)
[getAutomationResult2](#) (see page 937)
[getAutomationProcessTargetStatus](#) (see page 938)
[getAutomationProcessStatus](#) (see page 939)
[runAutomationSoftwareJobContainer](#) (see page 940)
[runAutomationOSInstallJob](#) (see page 941)
[runAutomationGenericJobContainer](#) (see page 941)
[getActivityLog](#) (see page 942)
[getActivityLog2](#) (see page 943)
[setActivationStatus](#) (see page 944)
[getSupportedOSForAssessment](#) (see page 944)
[SDTemplateProperties](#) (see page 945)
[SoftwarePackageFilter](#) (see page 945)
[SoftwarePackageProperties2](#) (see page 946)
[SDProperties](#) (see page 947)
[ManagerProperties](#) (see page 948)
[ManagerVersion](#) (see page 948)

[CreateSoftwareJobContainerProperties4](#) (see page 949)
[BootImageFilter](#) (see page 950)
[BootImageProperties](#) (see page 950)
[BootServerProperties](#) (see page 951)
[OSIMTargetProperties2](#) (see page 952)
[OSIMTargetProperties](#) (see page 952)
[BootServerFilter](#) (see page 953)
[ComputerProperties2](#) (see page 953)
[QueryFilter](#) (see page 956)
[QueryProperties2](#) (see page 956)
[QueryProperty](#) (see page 957)
[QueryFolderProperties](#) (see page 957)
[HWModelFilter](#) (see page 957)
[Health Monitoring Sequences](#) (see page 958)
[Sequences for Automation Web Services](#) (see page 958)

ComputerProperties2

The following sequences of elements specify the properties required when you are retrieving the properties of a computer.

xsd:string computerUUID

Specifies the UUID of the computer.

xsd:string computerLabel

Specifies the display name of the computer.

xsd:string computerDomainUUID

Specifies the UUID identifying the CA ITCM domain of the computer.

xsd:string computerPrimaryNetworkAddress

Specifies the IP address of the computer in 0.0.0.0 format.

xsd:string computerHostUUID

Specifies the agent-generated UUID that identifies the computer.

xsd:string computerAssetSourceUUID

Specifies the UUID identifying the common CA asset to which this computer is linked.

xsd:string computerHostName

Specifies the host name of the computer.

xsd:string computerSerialNumber

Specifies the serial number of the computer.

xsd:string computerDiscSerialNumber

Specifies the serial number of the computer disc.

xsd:string computerPrimaryMACAddress

Specifies the MAC address of the computer. The string must have a maximum length of 64 characters.

xsd:string computerAssetTag

Specifies the string value that identifies a computer asset.

xsd:long computerClassID

Specifies the numeric value indicating the class of the computer object. For example, Operating System.

dsm:UsageListMask computerUsageList

Specifies the computer usage list mask.

xsd:string computerURI

Specifies the fully qualified Uniform Resource Identifier of the computer.

xsd:string computerSystemID

Specifies the user-defined identifier for computer.

xsd:string computerVendorName

Specifies the name of the computer vendor.

xsd:string computerDomainLabel

Specifies the display name of the DSM domain to which the computer belongs.

xsd:string computerCreationUser

Specifies the name of the user account that created the computer record.

dsm:DateTime computerCreationDate

Specifies the computer creation date and time.

xsd:string computerLastUpdatedUser

Specifies the name of the user account that last updated the computer properties.

dsm:DateTime computerLastUpdatedDate

Specifies the date and time when the computer properties were last updated.

xsd:long computerVersionNumber

Specifies the version number of the computer record.

Note: The version number is incremented each time a record is updated.

xsd:string agentName

Specifies the agent name of the computer.

xsd:long agentRestrictions

Specifies the agent restrictions.

xsd:string agentDescription

Specifies the agent description.

xsd:string userdata1

Specifies the user defined data1.

xsd:string userdata2

Specifies the user defined data2.

xsd:string userdata3

Specifies the user defined data3.

xsd:string userdata4

Specifies the user defined data4.

xsd:string scalabilityserverName

Specifies the name of the scalability server that the agent is pointing to.

dsm:DateTime2 agentCreationDateTime

Specifies the agent creation data and time.

dsm:DateTime2 lastRunDateTime

Specifies the agent last run date and time.

xsd:string agentIPAddress

Specifies the IP address of the agent.

xsd:long agentTrustlevel

Specifies the trust level of the agent.

xsd:string agentOrigin

Specifies the origin of the agent.

dsm:AMAgentStatus amStatus

Specifies the AM Agent status.

dsm:SDAgentStatus sdStatus

Specifies the SD Agent status.

dsm:RCAgentStatus rcStatus

Specifies the RC Agent status.

dsm:ProtectionStatus protectionStatus

Specifies the Protection Agent status.

dsm:ASPStatus aspStatus

Specifies the ASP status.

dsm:AVStatus avStatus

Specifies the AV status.

dsm:HIPSStatus hipsStatus

Specifies the HIPS status.

QueryFilter

The QueryFilter sequence specifies a search filter that you can use for searching or listing queries.

dsm: QueryFilterCondition filterCondition

Specifies the filter condition you want to apply.

dsm: QueryProperty queryProperty

Specifies the property you want to filter.

xsd:string searchString

Specifies the string you want to search in the name property.

dsm: QueryTargetType targetType

Specifies the query target type.

QueryProperties2

The QueryProperties2 sequence contains the properties of a query.

xsd:string queryUUID

Specifies the UUID of the query.

Xsd:string queryName

Specifies the name of the query.

dsm: QueryTargetType queryType

Specifies the type of the query.

Xsd:boolean replicated

Specifies whether is query is replicated from enterprise manager. If true, it indicates that the query is replicated from enterprise manager.

QueryProperty

The QueryProperty sequence specifies the query properties on which you can apply the sort and filter conditions.

QUERY-LABEL

Specifies the name of the query.

QUERY-TARGET-TYPE

Specifies the target type of the query.

QueryFolderProperties

The QueryFolderProperties sequence contains the properties of a query folder.

Xsd:string folderName

Specifies the name of the query folder.

Xsd: long folderId

Specifies the ID of the query folder.

HWModelFilter

The HWModelFilter sequence specifies a search filter that you can use for searching or listing unique hardware models.

dsm: HWMODEL-FILTER-CONDITION filterCondition

Specifies the filter condition you want to apply.

xsd:string searchString

Specifies the string you want to search in the hardware model name.

AttributeMask

This sequence of elements specifies a number of attributes of a software package. A combination of the Boolean values may be set to true.

xsd:boolean system

Owned and created by the dsm system. Cannot be removed or manipulated.

xsd:boolean hidden

Not visible; used internally by dsm.

xsd:boolean distributed

A folder present at a Domain Manager that has been distributed from Enterprise Manager.

Remarks

None

AuthenticationData

This sequence of elements specifies the string and size of authentication data that must be sent to the domain controller for authentication.

char *authenticationData

Specifies the authentication data

long size

Specifies the size of the authentication data

Remarks

None

CreateDistributionContainerProperties

xsd:string name (m)**xsd:long priority**

The default value is 5, can be 1(highest) to 10 (lowest).

xsd:boolean prioritySupplied**xsd:string ahdInfo**

Advanced help desk information.

xsd:boolean ahdInfoSupplied

CreateDomainGroupProperties

xsd:string groupLabel (m)

Specifies the domain group name.

xsd:string groupDescription

Specifies the domain group description.

xsd:boolean groupDescriptionSupplied (m)

True or false depending on the groupDescription value.

CreateInstallSoftwareJobOrderPolicyProperties

This sequence of elements specifies which properties are required when creating a job order for a software job within a software policy.

xsd:long timeoutHours

The time in hours after which time, if the job has been unable to complete, the job will cease attempts to complete. If not supplied, the default value is 48.

xsd:boolean timeoutHoursSupplied

SoftwareJobStartupTime startWhen

Indicates when the software job should be started. If not supplied, the default value is EXACT_LOCAL_TIME.

xsd:boolean startWhenSupplied

xsd:string jobName

The name of the software job with which this order is associated. If no name is specified, Unicenter Software Delivery creates a default name.

xsd:boolean jobNameSupplied

SoftwareJobBootMask bootBeforeMask

Indicates the reboot options requested for the software job before the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootBeforeMaskSupplied

SoftwareJobBootMask bootAfterMask

This element indicates the reboot options requested for the software job after the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootAfterMaskSupplied**SoftwareJobOperationMask2 operationMask2**

Properties for the operation of the software job. If not supplied, the default value is calendarControl.

xsd:boolean operationMaskSupplied**xsd:string userParameters**

Contains user-defined parameters.

xsd:boolean userParametersSupplied**xsd:long promptTimeoutHours**

Indicates the number of hours for which a user prompt should be displayed before forced execution of the job. The default value is 1 day. This parameter should be set to a value that does not exceed the timeout value for the job. The default value is 48 hours.

xsd:boolean promptTimeoutHoursSupplied**xsd:string deliveryCalendar**

The name of a Unicenter delivery calendar to be used for scheduling the job. The calendar controls the days, dates, and times when delivery is permitted. The time cells not set in the Unicenter Calendar define the exclusion intervals. If the name of the calendar is unknown to Unicenter, the job is not created.

xsd:boolean deliveryCalendarSupplied

CreateInstallSoftwareJobOrderPolicyProperties3

This sequence of elements specifies which properties are required when creating a job order for a software job within a software policy.

xsd:long timeoutHours

Specifies the time in hours after which, if the job has been unable to complete, the job will cease attempts to complete. If you do not provide any value, the default value 48 is used.

xsd:boolean timeoutHoursSupplied

Specifies true or false depending on whether the timeoutHours parameter is supplied.

SoftwareJobStartupTime startWhen

Specifies when to start the software job. If you do not provide any value, the default value EXACT_LOCAL_TIME is used.

xsd:boolean startWhenSupplied

Specifies true or false depending on whether the startWhen parameter is supplied.

xsd:string jobName

Specifies the name of the software job with which this order is associated. If you do not specify any name, SD creates a default name.

xsd:boolean jobNameSupplied

Specifies true or false depending on whether the jobName parameter is supplied.

SoftwareJobBootMask3 bootBeforeMask

Specifies the restart options requested for the software job before executing the job. If you do not specify any value, the default value noReboot is used.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwareJobBootMask3 bootAfterMask

Specifies the restart options requested for the software job after executing the job. If you do not specify any value, the default value noReboot is used.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

SoftwareJobOperationMask2 operationMask2

Specifies the properties for the operation of the software job. If you do not specify any value, the default value calendarControl is used.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask2 parameter is supplied.

xsd:string userParameters

Specifies the user-defined parameters.

xsd:boolean userParametersSupplied

Specifies true or false depending on whether the userParameters parameter is supplied.

xsd:long promptTimeoutHours

Specifies the number of hours for which to display a user prompt before the job is forced to start. The default value is 24 hours. You must set this parameter to a value that does not exceed the timeout value for the job (default value is 48 hours).

xsd:boolean promptTimeoutHoursSupplied

Specifies true or false depending on whether the promptTimeoutHours parameter is supplied.

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to use for scheduling the job. The calendar controls the days, dates, and times of the scheduled delivery. The time cells that are not set in the calendar define the exclusion intervals. If the name of the calendar is unknown to SD, the job is not created.

xsd:boolean deliveryCalendarSupplied

Specifies true or false depending on whether the deliveryCalendar parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

CreateInstallSoftwareJobOrderProperties

This sequence of elements specifies the software job order properties are to be initialized when a software job order is created.

- `DateTime` `executionDate`
Default: Current date and time of the manager computer.
- `xsd:boolean` `executionDateSupplied`
- `xsd:long` `timeoutHours`
Default: 168 hours.
- `xsd:boolean` `timeoutHoursSupplied`
- `SoftwareJobStartupTime` `startWhen`
Default: EXACT_LOCAL_TIME.
- `xsd:boolean` `startWhenSupplied`
- `xsd:string` `jobName (m)`
- `xsd:boolean` `jobNameSupplied (m)`
- `SoftwareJobBootMask` `bootBeforeMask`
Default: noReboot.
- `xsd:boolean` `bootBeforeMaskSupplied`
- `SoftwareJobBootMask` `bootAfterMask`
Default: noReboot.
- `xsd:boolean` `bootAfterMaskSupplied`
- `SoftwareJobOperationMask2` `operationMask2`
Default: calendarControl.
- `xsd:boolean` `operationMaskSupplied`
- `xsd:string` `userParameters`
- `xsd:boolean` `userParametersSupplied`
- `xsd:long` `promptTimeoutHours`
Default: 24 hours.
- `xsd:boolean` `promptTimeoutHoursSupplied`
- `xsd:string` `deliveryCalendar`
- `xsd:boolean` `deliveryCalendarSupplied`
- `DateTime` `deliveryDateTime`
Default: Current date and time of the manager computer.
- `xsd:boolean` `deliveryDateTimeSupplied`

CreateInstallSoftwareJobOrderProperties3

This sequence of elements specifies the software job order properties that are to be initialized when a software job order is created.

DateTime executionDate

Specifies the execution date. If not supplied, the default value is the current date and time of the manager computer.

xsd:boolean executionDateSupplied

Specifies true or false depending on whether the executionDate parameter is supplied.

xsd:long timeoutHours

Specifies the time in hours after which, if the job has been unable to complete, the job will cease attempts to complete. If not supplied, the default value is set to 168 hours.

xsd:boolean timeoutHoursSupplied

Specifies true or false depending on whether the timeoutHours parameter is supplied.

SoftwareJobStartupTime startWhen

Specifies when to start the software job. If not supplied, the default value is EXACT_LOCAL_TIME.

xsd:boolean startWhenSupplied

Specifies true or false depending on whether the startWhen parameter is supplied.

xsd:string jobName (m)

Specifies the name of the software job with which this order is associated. If no name is specified, SD creates a default name.

xsd:boolean jobNameSupplied

Specifies true or false depending on whether the jobName parameter is supplied.

SoftwareJobBootMask3 bootBeforeMask

Specifies the reboot options requested for the software job before the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwareJobBootMask3 bootAfterMask

Specifies the reboot options requested for the software job after the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

SoftwareJobOperationMask2 operationMask2

Specifies the properties for the operation of the software job. If not supplied, the default value is calendarControl.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask2 parameter is supplied.

xsd:string userParameters

Specifies the user-defined parameters.

xsd:boolean userParametersSupplied

Specifies true or false depending on whether the userParameters parameter is supplied.

xsd:long promptTimeoutHours

Specifies the number of hours for which a user prompt must be displayed before forced execution of the job starts. If not supplied, the default value is 24 hours.

xsd:boolean promptTimeoutHoursSupplied

Specifies true or false depending on whether the promptTimeoutHours parameter is supplied.

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to be used for scheduling the job. The calendar controls the days, dates, and times when delivery is permitted. The time cells that are not set in the calendar define the exclusion intervals. If the name of the calendar is unknown to SD, the job is not created.

xsd:boolean deliveryCalendarSupplied

Specifies true or false depending on whether the deliveryCalendar parameter is supplied.

DateTime deliveryDateTime

Specifies the delivery date and time. If not supplied, the default value is the current date and time of the manager computer.

xsd:boolean deliveryDateTimeSupplied

Specifies true or false depending on whether the deliveryDateTime parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

CreateQueryProperties

This sequence of elements specifies which volume properties are to be initialized when a query is created.

- xsd:string queryName (^m)
- xsd:boolean queryNameSupplied (^m)
Note: Set to true value only.
- xsd:string queryTable
- xsd:boolean queryTableSupplied
- xsd:boolean queryTypeSupplied
Default: AGENT
- xsd:string queryTableLabelField
- xsd:boolean queryTableLabelFieldSupplied
- xsd:string queryTableLabelPrimaryKey
- xsd:boolean queryTableLabelPrimaryKeySupplied
- xsd:string queryTableLabelSecondaryKey
- xsd:boolean queryTableLabelSecondaryKeySupplied
- ArrayOfQueryFragment queryFragments (^m)
- xsd:boolean queryFragmentsSupplied (^m)
Note: Set to true value only.

Remarks

If the value of QueryType is Agent, the following optional fields of CreateQueryProperties are mandatory:

- xsd:string queryTableLabelField
- xsd:boolean queryTableLabelFieldSupplied
- xsd:string queryTableLabelPrimaryKey
- xsd:boolean queryTableLabelPrimaryKeySupplied
- xsd:string queryTable
- xsd:boolean queryTableSupplied

For more details, see [QueryProperties](#) (see page 716).

CreateQueryProperties2

This sequence of elements specifies which query properties are to be initialized when a query is created.

- `xsd:string queryName (m)`
- `xsd:boolean queryNameSupplied (m)`
Note: Set to true value only.
- `xsd:string queryTable`
- `xsd:boolean queryTableSupplied`
- `xsd:boolean queryTypeSupplied`
Default: AGENT
- `xsd:string queryTableLabelField`
- `xsd:boolean queryTableLabelFieldSupplied`
- `xsd:string queryTableLabelPrimaryKey`
- `xsd:boolean queryTableLabelPrimaryKeySupplied`
- `xsd:string queryTableLabelSecondaryKey`
- `xsd:boolean queryTableLabelSecondaryKeySupplied`
- `ArrayOfQueryFragment2 queryFragments2 (m)`
- `xsd:boolean queryFragmentsSupplied (m)`
Note: Set to true value only.

Remarks

If the value of QueryType is Agent, the following optional fields of [CreateQueryProperties](#) (see page 640) are mandatory:

- `xsd:string queryTableLabelField`
- `xsd:boolean queryTableLabelFieldSupplied`
- `xsd:string queryTableLabelPrimaryKey`
- `xsd:boolean queryTableLabelPrimaryKeySupplied`
- `xsd:string queryTable`
- `xsd:boolean queryTableSupplied`

CreateSecurityAreaProperties

`xsd:string securityAreaName (m)`

Name of security Area to be created.

`xsd:string securityAreaDescription`

The description to be set for the security Area.

xsd:bool securityAreaDescriptionSupplied

True or false depending on whether the description is provided or not.

CreateSecurityProfileProperties

xsd:string securityProfileName (m)

Name of the security profile to be created. Name should be provided in URI format. For example, (winnt://<domain name>/<username or groupname>)

xsd:string securityProfileDescription

The description of the security profile.

xsd:bool securityProfileDescriptionSupplied

True or false depending on whether the description is provided or not.

xsd:bool securityAreaEnabled

Set this parameter to 'true' if you want security area support to be enabled for this profile.

xsd:bool securityAreaEnabledSupplied

True or false depending on whether security area enabled supplied or not.

dsm:SecurityProfilePermission securityProfilePermission

The properties required to create default security permissions.

xsd:bool securityProfilePerlmissionSupplied

True or false depending on whether SecurityProfilePermission is provided or not.

**dsm: SecurityProfileSpecialPermissionFlags
securityProfileSpecialPermissionFlags**

The properties required to create special access permission.

User needs to fill this structure if he or she selects 'SPECIALACCESS' from 'SecurityProfilePermission' enum. Default value is 'No Access'.

xsd:bool SecurityPlofileSpecialPermissionFlagsSupplied

True or false depending on whether SecurityProfileSpecialPermissionFlags is provided or not.

CreateSoftwareGroupProperties

Members of sequence: CreateSoftwareGroupProperties

xsd:string softwareGroupName (m)

Specifies the name of software group to be created.

xsd:string comment

Specifies the comment for the software group.

xsd:boolean commentSupplied (m)

True or false depending on the value of comment.

SoftwareGroupType softwareGroupType

The default value is *SW_PACKAGE_GROUP*.

xsd:boolean softwareGroupTypeSupplied (m)

True or false depending on the value of softwareGroupType.

CreateSoftwareJobContainerProperties

This sequence of elements specifies which job container properties are to be initialized when a software job container is created.

- **xsd:string parentSoftwareJobContainerId**
Default: SD_Jobs
- **xsd:boolean parentSoftwareJobContainerIdSupplied**
- **xsd:string name (m)**
- **xsd:boolean nameSupplied (m)**
- **xsd:string comment**
- **xsd:boolean commentSupplied**
- **SoftwareJobContainerPropertyMask**
softwareJobContainerPropertyMask
Default: linkageBatch and Cascade
- **xsd:boolean softwareJobContainerPropertyMaskSupplied**

Remarks

For more details, see [SoftwareJobContainerProperties](#) (see page 786).

CreateSoftwarePackageProcedureDependencyProperties

This sequence of elements specifies which dependency properties are to be initialized when a software package procedure dependency is created.

- xsd:string softwarePackageProcedureId (^m)
- xsd:boolean softwarePackageProcedureIdSupplied (^m)
- xsd:string dependantSoftwarePackageProcedureId (^m)
- xsd:boolean dependantSoftwarePackageProcedureIdSupplied (^m)

Remarks

For more details, see [SoftwarePackageProcedureDependencyProperties](#) (see page 837).

CreateSoftwarePackageProcedureGroupProperties

This sequence of elements specifies which procedure group properties are to be initialized when a software package procedure group is created.

- xsd:string parentSoftwarePackageGroupId
Default: SD_Software Library
- xsd:boolean parentSoftwarePackageGroupIdSupplied
- xsd:string name (^m)
- xsd:boolean nameSupplied (^m)
- xsd:string comment
- xsd:boolean commentSupplied

Remarks

For more details, see [SoftwarePackageProcedureGroupProperties](#) (see page 838).

CreateSoftwarePackageProcedureProperties

This sequence of elements specifies which procedure properties are to be initialized when a software package procedure is created.

- xsd:string softwarePackageId (^m)
- xsd:boolean softwarePackageIdSupplied (^m)
- xsd:string softwarePackageProcedureName (^m)
- xsd:boolean softwarePackageProcedureNameSupplied (^m)

- SoftwarePackageProcedureTask softwarePackageProcedureTask
Default: INSTALL.
- xsd:boolean softwarePackageProcedureTaskSupplied
- SoftwarePackageProcedureType softwarePackageProcedureType
Default: EXEFILE.
- xsd:boolean softwarePackageProcedureTypeSupplied
- xsd:string comment
- xsd:boolean commentSupplied
- xsd:boolean userSelectable
Default: false.
- xsd:boolean userSelectableSupplied
- xsd:boolean externalProcedure
Default: false.
- xsd:boolean externalProcedureSupplied
- xsd:long operatingSystemType
Default: SDAPI_OS_ANY.
- xsd:boolean operatingSystemTypeSupplied
- xsd:string subPath
- xsd:boolean subPathSupplied
- xsd:string procToRun (^m)
- xsd:boolean procToRunSupplied (^m)
- xsd:string parameters
- xsd:boolean parametersSupplied
- xsd:string softwarePackageVolumeld (^m)
- xsd:boolean softwarePackageVolumeldSupplied (^m)
- SoftwarePackageProcedureOperationMask operationMask
- xsd:boolean operationMaskSupplied
- SoftwarePackageProcedureBootMask bootBeforeMask
Default: noReboot.
- xsd:boolean bootBeforeMaskSupplied
- SoftwarePackageProcedureBootMask bootAfterMask
Default: noReboot.
- xsd:boolean bootAfterMaskSupplied
- xsd:string dependancyQueryId

- xsd:boolean dependencyQueryIdSupplied
- SoftwarePackageProcedureManagementTypeMask
managementTypeMask

Default: SD.

- xsd:boolean managementTypeMaskSupplied

Remarks

For more details, see [SoftwarePackageProcedureProperties](#) (see page 843).

CreateSoftwarePackageProcedureProperties2

The CreateSoftwarePackageProcedureProperties3 sequence of elements specifies which procedure properties are to be initialized when a software package procedure is created.

Parameters

xsd:string softwarePackageId (")

Specifies the UUID that identifies the software package.

xsd:boolean softwarePackageIdSupplied (")

Specifies true or false depending on whether the softwarePackageId parameter is supplied.

xsd:string softwarePackageProcedureName (")

Specifies the name of the software package procedure.

xsd:boolean softwarePackageProcedureNameSupplied (")

Specifies true or false depending on whether the softwarePackageProcedureName parameter is supplied.

SoftwarePackageProcedureTask softwarePackageProcedureTask

Specifies the task-type of the software package procedure. If you do not provide any value, the default value INSTALL is used.

xsd:boolean softwarePackageProcedureTaskSupplied

Specifies true or false depending on whether the softwarePackageProcedureTask parameter is supplied.

SoftwarePackageProcedureType softwarePackageProcedureType

Specifies the type of the software package procedure. If you do not provide any value, the default value EXEFILE is used.

xsd:boolean softwarePackageProcedureTypeSupplied

Specifies true or false depending on whether the softwarePackageProcedureTypeSupplied parameter is supplied.

xsd:string comment

Specifies a comment for the software package procedure.

xsd:boolean commentSupplied

Specifies true or false depending on whether the comment parameter is supplied.

xsd:boolean userSelectable

Specifies whether the procedure is allowed to be included in the SD Catalog. If you do not provide the appropriate value, the default value false is used.

xsd:boolean userSelectableSupplied

Specifies true or false depending on whether the userSelectable parameter is supplied.

xsd:boolean externalProcedure

Specifies whether the procedure is external to SD. If you do not provide the appropriate value, the default value false is used.

xsd:boolean externalProcedureSupplied

Specifies true or false depending on whether the externalProcedure parameter is supplied.

xsd:long operatingSystemType

Specifies the type of the operating system. If you do provide any value, the default value SDAPI_OS_ANY is used.

xsd:boolean operatingSystemTypeSupplied

Specifies true or false depending on whether the operatingSystemType parameter is supplied.

xsd:string subPath

Specifies the path of the procedure within the volume.

xsd:boolean subPathSupplied

Specifies true or false depending on whether the subPath parameter is supplied.

xsd:string procToRun (^m)

Specifies the procedure command that is to be run.

xsd:boolean procToRunSupplied (^m)

Specifies true or false depending on whether the procToRun parameter is supplied.

xsd:string parameters

Specifies a string value that contains the input parameters for the procedure.

xsd:boolean parametersSupplied

Specifies true or false depending on whether the parameters parameter is supplied.

xsd:string softwarePackageVolumeld ("")

Specifies the UUID identifying the volume of the software package to which the procedure is associated.

xsd:boolean softwarePackageVolumeldSupplied ("")

Specifies true or false depending on whether the softwarePackageVolumeld parameter is supplied.

SoftwarePackageProcedureOperationMask2 operationMask

Specifies the operation property for the procedure operation.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask parameter is supplied.

SoftwarePackageProcedureBootMask2 bootBeforeMask

Specifies the options requested for the software package procedure before the procedure is executed. If no value is provided, the default value noReboot is used.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwarePackageProcedureBootMask2 bootAfterMask

Specifies the options requested for the software package procedure after the procedure is executed. If no value is provided, the default value noReboot is used.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

xsd:string dependancyQueryId

Specifies the UUID of the query that checks whether the condition is fulfilled, provided a prerequisite is defined for the procedure.

xsd:boolean dependancyQueryIdSupplied

Specifies true or false depending on whether the dependancyQueryId parameter is supplied.

**SoftwarePackageProcedureManagementTypeMask
managementTypeMask**

Specifies whether the requested software package for the procedure is allowed to run on the Software Managed software targets. If no value is provided, the default value SD is used.

xsd:boolean managementTypeMaskSupplied

Specifies true or false depending on whether the managementTypeMask parameter is supplied.

Remarks

None

CreateSoftwarePackageProcedureProperties3

The CreateSoftwarePackageProcedureProperties3 sequence of elements specifies which procedure properties are to be initialized when a software package procedure is created.

Parameters

xsd:string softwarePackageId

Specifies the UUID that identifies the software package.

xsd:boolean softwarePackageIdSupplied

Specifies true or false depending on whether the softwarePackageId parameter is supplied.

dsm:CommonSoftwarePackageProcedureProperties3

commonSoftwarePackageProcedureProperties

Procedure properties

CommonSoftwarePackageProcedureProperties3

Procedure properties used while creating/updating the software package procedure

Parameters

xsd:string softwarePackageName

Specifies the name of the software package procedure.

xsd:boolean softwarePackageNameSupplied

Specifies true or false depending on whether the softwarePackageName parameter is supplied.

SoftwarePackageProcedureTask softwarePackageProcedureTask

Specifies the task-type of the software package procedure. If you do not provide any value, the default value INSTALL is used.

xsd:boolean softwarePackageProcedureTaskSupplied

Specifies true or false depending on whether the softwarePackageProcedureTask parameter is supplied.

SoftwarePackageProcedureType softwarePackageProcedureType

Specifies the type of the software package procedure. If you do not provide any value, the default value EXEFILE is used.

xsd:boolean softwarePackageProcedureTypeSupplied

Specifies true or false depending on whether the softwarePackageProcedureTypeSupplied parameter is supplied.

xsd:string comment

Specifies a comment for the software package procedure.

xsd:boolean commentSupplied

Specifies true or false depending on whether the comment parameter is supplied.

xsd:boolean userSelectable

Specifies whether the procedure is allowed to be included in the SD Catalog. If you do not provide the appropriate value, the default value false is used.

xsd:boolean userSelectableSupplied

Specifies true or false depending on whether the userSelectable parameter is supplied.

xsd:boolean externalProcedure

Specifies whether the procedure is external to SD. If you do not provide the appropriate value, the default value false is used.

xsd:boolean externalProcedureSupplied

Specifies true or false depending on whether the externalProcedure parameter is supplied.

xsd:long operatingSystemType

Specifies the type of the operating system. If you do provide any value, the default value SDAPI_OS_ANY is used.

xsd:boolean operatingSystemTypeSupplied

Specifies true or false depending on whether the operatingSystemType parameter is supplied.

xsd:string subPath

Specifies the path of the procedure within the volume.

xsd:boolean subPathSupplied

Specifies true or false depending on whether the subPath parameter is supplied.

xsd:string procToRun

Specifies the procedure command that is to be run.

xsd:boolean procToRunSupplied

Specifies true or false depending on whether the procToRun parameter is supplied.

xsd:string parameters

Specifies a string value that contains the input parameters for the procedure.

xsd:boolean parametersSupplied

Specifies true or false depending on whether the parameters parameter is supplied.

xsd:string softwarePackageVolumeld

Specifies the UUID identifying the volume of the software package to which the procedure is associated.

xsd:boolean softwarePackageVolumeldSupplied

Specifies true or false depending on whether the softwarePackageVolumeld parameter is supplied.

SoftwarePackageProcedureOperationMask3 operationMask

Specifies the operation property for the procedure operation.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask parameter is supplied.

SoftwarePackageProcedureBootMask2 bootBeforeMask

Specifies the options requested for the software package procedure before the procedure is executed. If no value is provided, the default value noReboot is used.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwarePackageProcedureBootMask2 bootAfterMask

Specifies the options requested for the software package procedure after the procedure is executed. If no value is provided, the default value noReboot is used.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

xsd:string dependancyQueryId

Specifies the UUID of the query that checks whether the condition is fulfilled, provided a prerequisite is defined for the procedure.

xsd:boolean dependancyQueryIdSupplied

Specifies true or false depending on whether the dependancyQueryId parameter is supplied.

**SoftwarePackageProcedureManagementTypeMask
managementTypeMask**

Specifies whether the requested software package for the procedure is allowed to run on the Software Managed software targets. If no value is provided, the default value SD is used.

xsd:boolean managementTypeMaskSupplied

Specifies true or false depending on whether the managementTypeMask parameter is supplied.

SoftwarePackageProcedureOperationMask3

The SoftwarePackageProcedureOperationMask2 sequence includes a series of flags that indicate options for the operation of a software package procedure.

xsd:boolean offline

Executes the procedure in the off-line mode from the scalability server.

xsd:boolean prompt

Displays a message on the target computer before the job starts.

xsd:boolean calendarControl

Controls the execution by the job execution calendar defined for this target agent.

xsd:boolean shield

Prevents the user from being logged-on while the job executes.

xsd:boolean lock

xsd:boolean noPromptCancel

Enables the user to cancel a prompt user job.

xsd:boolean background

xsd:boolean rerun

Runs the Activate or Configure job every time the software policy is evaluated.

xsd:boolean permActArea

xsd:boolean execPromptTimeout

Runs the job if the prompt timeout expires. If the value is not specified, the job is postponed.

xsd:boolean resolveQueryGroups

Resolves the target groups before the job is set up.

xsd:boolean autoDeliverItem

Stores the package in the scalability server library.

xsd:boolean defaultItemProc

Specifies that this is the default procedure for this package with the task defined for the procedure.

xsd:boolean renewFailedOnly

Renews only the failed jobs.

xsd:boolean useDefaultJobMessage

(If enabled) Specifies that the software package procedure comment is used as the default user message for jobs created using this procedure.

Xsd:boolean excludeFromRAC

If set to true, this procedure will be excluded from RAC installation.

xsd:boolean runAtShutdown

This is applicable only for UNIX machines

computerClassID

Note: For the latest list of supported operating systems, visit CA Support.

The possible value for computerClassID is one of the following:

4	Windows Server 2003 Web
5	Windows Server 2003 Ent
6	Windows Server 2003 DC
8	Windows 2000 Professional
9	Windows 2000 Advanced Server
10	Windows 2000 DC Server
11	Windows 2000 Server
13	Windows 95
14	Windows 98
15	Windows 98 SE
16	Windows ME
18	Windows NT Workstation 3.51
19	Windows NT workstation 4.0
20	Windows NT Server 4.0
21	Windows NT Server 4.0 Enterprise
23	Windows XP Home
24	Windows XP Professional
25	Windows XP Media Center Edition
26	Windows XP Tablet PC Edition
27	Windows XP Professional x64 Edition

29	Windows 3.0
30	Windows 3.1
31	Windows 3.11
33	Windows CE 2.00
34	Windows CE 2.01
35	Windows CE 2.11
36	Windows CE 3.00
37	Windows Mobile 2003
38	Palm OS
39	Symbian
41	DG_UX
44	DEC 3.0
45	DEC 3.2
46	DEC 4.0
48	FUJIXp
50	HPUX 9
53	ICLUnix
75	NCR SV
76	NCR SV 2x

77	NCR SV 3x
78	NCR SST
79	NCR SST S4
80	NCR SST S4I
82	AIX 3.2
83	AIX 4.1
84	AIX 4.2
85	AIX 4.3
86	AIX 5.0
87	AIX 5.1
88	AIX 5.2
90	Unixware 2.0
91	Unixware 2.1
93	SCO OpenServer 3.2
94	SCO OpenServer 5.2
96	Sinix 5.42
97	Sinix 5.43
98	Sinix 5.44
99	Sinix 5.45
101	Irix 6.2
102	Irix 6.3
103	Irix 6.4
104	Irix 6.5
106	Dynix 4.2.x
107	Dynix 4.4.x
109	Solaris 2.3
110	Solaris 2.4
111	Solaris 2.5
112	Solaris 2.6
113	Solaris 7
114	Solaris 8
115	Solaris 9

116	SunOS 4.1.4
117	OS/2
118	AS400
119	OS390
120	DOS
122	Macintosh 10.0
123	Macintosh 10.1
124	Macintosh 10.2
125	Macintosh 10.3
126	Macintosh 10.4
129	OpenVMS 5.5
130	OpenVMS 6.0
131	OpenVMS 6.1
132	OpenVMS 6.2
133	OpenVMS 7.0
134	OpenVMS 7.1
135	OpenVMS 7.2
136	Windows NT Server 3.51
137	Windows XP Embedded
138	Windows Server 2003 Standard
139	Windows Small Business Server 2003
140	Windows Server 2003 Enterprise x64
141	Windows Server 2003 Enterprise 64-Bit Itanium

202	Caldera Linux 3.0
203	Caldera Linux 3.1
204	SCO Linux 4.0
206	Connectiva Linux 8
207	Connectiva Linux 9
208	Connectiva Linux 10
210	Debian GNU/Linux 2.0
211	Debian GNU/Linux 2.1
212	Debian GNU/Linux 2.2
213	Debian GNU/Linux 3.0
215	Fedora Core 1
216	Fedora Core 2
218	Miracle Linux 2.1
219	Miracle Linux 3.0
221	Red Flag Linux 3.0
222	Red Flag Linux 3.2
223	Red Flag Linux 4.0
224	Red Flag Linux 4.1
226	Red Hat Linux 7.0
227	Red Hat Linux 7.1
228	Red Hat Linux 7.2
229	Red Hat Linux 7.3
230	Red Hat Linux 8.0
231	Red Hat Enterprise Linux ES 2.1

232	Red Hat Enterprise Linux AS 2.1
233	Red Hat Enterprise Linux WS 2.1
234	Red Hat Enterprise Linux ES 3
235	Red Hat Enterprise Linux AS 3
236	Red Hat Enterprise Linux WS 3
237	Red Hat Enterprise Linux ES 4.0
238	Red Hat Enterprise Linux AS 4.0
239	Red Hat Enterprise Linux WS 4.0
240	Red Hat Desktop
242	Sun Java Desktop System 2003
243	Sun Java Desktop System 2
245	SuSE Linux 7.0
246	SuSE Linux 7.1
247	SuSE Linux 7.3
248	SuSE Linux 8.0
249	SuSE Linux 8.1
250	SuSE Linux 8.2
251	SuSE Linux Professional 9
252	SuSE Linux Professional 9.1
253	SuSE Linux Personal 9
254	SuSE Linux Personal 9.1
255	SuSE Linux Enterprise Server 7
256	SuSE Linux Enterprise Server 8
257	SuSE Linux Enterprise Server 9
258	SuSE Standard Server 8
259	SuSE Desktop 1
260	SuSE Desktop 2
262	TurboLinux Server 8
263	TurboLinux Workstation 8
264	TurboLinux Enterprise 8
265	TurboLinux 10
267	Xandros Desktop 1.0
268	Xandros Desktop 1.1
269	Xandros Desktop 2.0

270	OpenVMS 7.3
271	Unixware 7.1.4
273	SCO OpenServer 5.7
276	Nokia AdminSuite 1.0
277	OpenVMS 8.0
278	OpenVMS 8.2
279	Solaris 10
280	AIX 5.3
281	NCR MP-RAS 3.02
282	HPUX 11.10
283	HPUX 11.22
284	HPUX 11.23
285	HPUX 11.00
286	HP Tru64 5.1
287	HP Tru64 5.0
288	HP Tru64 4.0F
289	HP Tru64 4.0G
290	HPUX 10.00
291	HPUX 10.10
292	HPUX 10.20
293	HPUX 10.30
294	HPUX 11.20
298	Windows Vista Business

299	Windows Vista Enterprise
300	Windows Vista Home Premium
301	Windows Vista Home Basic
302	Windows Vista Ultimate
303	Windows Vista Business x64 Edition
304	Windows Vista Enterprise x64 Edition
305	Windows Vista Home Premium x64 Edition
306	Windows Vista Home Basic x64 Edition
307	Windows Vista Ultimate x64 Edition
309	Windows Mobile 5.0
5029	Macintosh 10.x or later
5082	SuSE Linux Enterprise Server 10
5083	SuSE Linux Enterprise Server 10.1
5084	SuSE Linux Enterprise Desktop 10
5085	SuSE Linux Enterprise Desktop 10.1
5088	Windows Server 2003 Standard x64
5089	Windows Server 2003 Datacenter x64
5090	Unixware 7.1.3
5091	Macintosh 10.5
5092	Red Hat Enterprise Linux Server 5
5093	Red Hat Enterprise Linux Desktop 5
5094	Red Hat Enterprise Linux 5 Advanced Platform
5096	Sun Servers
5097	Sun Fire E25K
5098	Sun Fire E20K
5099	Sun Fire 15K
5100	Sun Fire 12K
5101	Sun Enterprise 10000
5102	Sun Enterprise 6500
5103	Sun Enterprise 5500
5104	Sun Enterprise 4500
5105	Sun Enterprise 3500

5107	Windows Embedded for Point of Service Version 1
5110	Windows Embedded for Point of Service 1.0
5111	HPUX 11.11
5112	HPUX 11.31
5115	Sun RTOS
5116	Red Hat Enterprise Linux Server 5.1
5117	Red Hat Enterprise Linux Desktop 5.1
5118	Red Hat Enterprise Linux 5.1 Advanced Platform
5120	Windows Server 2008 Standard Edition
5121	Windows Server 2008 Standard Edition x64 Edition
5122	Windows Server 2008 Enterprise Edition
5123	Windows Server 2008 Enterprise Edition x64 Edition
5124	Windows Server 2008 Datacenter Edition
5125	Windows Server 2008 Datacenter Edition x64 Edition
5126	Windows Web Server 2008

5127	Windows Web Server 2008 x64 Edition
5128	Windows Storage Server 2008
5129	Windows Storage Server 2008 x64 Edition
5130	Windows Small Business Server 2008
5131	Windows Essential Business Server 2008
5132	Windows Server 2008 IA-64 Edition
5137	Windows Mobile 6.0
5139	AIX 6.1
5148	Windows Server 2008 Hyper-V
5149	VMWare ESX Server 3.5
5150	VMWare ESXi 3.5
5153	Virtual Host Machine
5154	HP Chassis
5155	IBM Server
5157	Windows 7 Enterprise
5158	Windows 7 Enterprise x64 Edition
5159	Windows 7 Home Premium
5160	Windows 7 Home Premium x64 Edition
5161	Windows 7 Professional
5162	Windows 7 Professional x64 Edition
5163	Windows 7 Starter
5164	Windows 7 Home Basic
5165	Windows 7 Home Basic x64 Edition
5166	Windows 7 Ultimate
5167	Windows 7 Ultimate x64 Edition
5168	Macintosh 10.6
5169	SUSE Linux Enterprise Server 11
5189	Windows Server 2008 R2 Datacenter x64 Edition
5190	Windows Server 2008 R2 Enterprise x64 Edition
5191	Windows Server 2008 R2 IA-64 Edition
5192	Windows Server 2008 R2 Standard x64 Edition
5193	Windows Server 2008 Server Core Standard Edition
5194	Windows Server 2008 Server Core Standard x64 Edition
5195	Windows Server 2008 R2 Server Core Standard x64 Edition
5196	Windows Server 2008 Server Core Enterprise Edition

5197	Windows Server 2008 Server Core Enterprise x64 Edition
5198	Windows Server 2008 R2 Server Core Enterprise x64 Edition
5199	Windows Server 2008 Server Core Datacenter Edition
5200	Windows Server 2008 Server Core Datacenter x64 Edition
5201	Windows Server 2008 R2 Server Core Datacenter x64 Edition
5202	Windows Web Server 2008 R2 x64 Edition
5203	Windows Web Server 2008 Server Core Edition
5204	Windows Web Server 2008 Server Core x64 Edition
5205	SuSE Linux Enterprise Desktop 11
5210	Windows Server 2008 Server Core Installation
5211	Windows Server 2008 (x64) Server Core Installation
5221	openSUSE 11.0
5222	openSUSE 11.1
5223	Sun eXtended System Control Facility (XSCF)
5224	Windows Mobile 6.1
5226	VMWare ESX Server 4.0
5229	Windows Embedded POSReady 2009
5230	Windows Storage Server 2008 Standard Edition
5231	Windows Storage Server 2008 Basic Edition
5232	Windows Storage Server 2008 Enterprise Edition
5233	Windows Storage Server 2008 Workgroup Edition
5234	Windows Storage Server 2008 Standard X64 Edition
5235	Windows Storage Server 2008 Basic X64 Edition
5236	Windows Storage Server 2008 Enterprise X64 Edition
5237	Windows Storage Server 2008 Workgroup X64 Edition
5238	VMWare ESX Server 4.1

5239	Windows Mobile 6.5
5240	VMware ESXi 4.0
5241	VMware ESXi 4.1
5243	Citrix XenServer 4.0
5244	Citrix XenServer 4.1
5245	Citrix XenServer 5.0
5246	Citrix XenServer 5.5
5247	Citrix XenServer 5.6
5248	Windows Server 2008 R2 Hyper-V
5249	Hyper-V Server 2008
5250	Hyper-V Server 2008 R2
5254	Windows Embedded Standard 7 Edition
5255	Windows Embedded Standard 7 X64 Edition
5257	AIX 7.1
5258	Red Hat Enterprise Linux Server 6.0 x86
5259	Red Hat Enterprise Linux Server 6.0 x86_64
5260	Debian GNU/Linux 5.0
5261	Debian GNU/Linux 6.0
5262	Debian GNU/Linux 7.0
5264	Ubuntu Linux 10.04
5265	Ubuntu Linux 10.10
5266	Ubuntu Linux 11.04
5267	Ubuntu Linux 11.10
5268	Ubuntu Linux 12.04
5269	Macintosh 10.7
5270	Red Hat Enterprise Linux Server 6.1 x86
5271	Red Hat Enterprise Linux Server 6.1 x86_64
5272	openSUSE 11.2 x86
5273	openSUSE 11.2 x86_64

5274	openSUSE 11.3 x86
5275	openSUSE 11.3 x86_64
5276	openSUSE 11.4 x86
5277	openSUSE 11.4 x86_64
5278	Red Hat Enterprise Linux Server 6.2 x86
5279	Red Hat Enterprise Linux Server 6.2 x86_64
5296	Windows CE 6.00
5297	Windows CE 5.00
5298	Windows CE .NET 4.0
5299	Windows CE .NET 4.1
5300	Windows CE .NET 4.2
5302	Oracle Enterprise Linux Server 5 x86
5303	Oracle Enterprise Linux Server 5 x86_64
5304	Oracle Linux Server 6 x86
5305	Oracle Linux Server 6 x86_64
5309	Red Hat Enterprise Linux Server 6.3 x86
5310	Red Hat Enterprise Linux Server 6.3 x86_64
5311	Windows Embedded POSReady 7
5313	Windows 8 Pro x86
5314	Windows 8 Pro x64
5315	Windows 8 Enterprise x86
5316	Windows 8 Enterprise x64
5317	Windows Embedded Standard 8 x86
5318	Windows Embedded Standard 8 x64
5319	Windows 8 x86
5320	Windows 8 x64
5322	Windows Server 2012 Foundation x64

5323	Windows Server 2012 Essentials x64
5324	Windows Server 2012 Standard x64
5325	Windows Server 2012 Datacenter x64
5332	Windows Storage Server 2012 Workgroup x64
5333	Windows Storage Server 2012 Standard x64
5334	Solaris 11
5337	Macintosh 10.8
5338	VMware ESXi 5.1
5339	Red Hat Enterprise Linux 5.5 Advanced Platform
5340	Red Hat Enterprise Linux 5.6 Advanced Platform
5341	Red Hat Enterprise Linux 5.7 Advanced Platform
5342	Red Hat Enterprise Linux 5.8 Advanced Platform
5343	Red Hat Enterprise Linux 5.9 Advanced Platform
5344	Red Hat Enterprise Linux Desktop 5.5
5345	Red Hat Enterprise Linux Desktop 5.6
5346	Red Hat Enterprise Linux Desktop 5.7
5347	Red Hat Enterprise Linux Desktop 5.8
5348	Red Hat Enterprise Linux Desktop 5.9
5349	Red Hat Enterprise Linux Server 5.5
5350	Red Hat Enterprise Linux Server 5.6
5351	Red Hat Enterprise Linux Server 5.7
5352	Red Hat Enterprise Linux Server 5.8
5353	Red Hat Enterprise Linux Server 5.9
5354	Red Hat Enterprise Linux Server 6.4 x86
5355	Red Hat Enterprise Linux Server 6.4 x86_64
5356	Hyper-V Server 2012
5357	Windows Server 2012 Hyper-V
5402	openSUSE Linux 12
5403	Citrix XenServer 6.0
5405	Citrix XenServer 6.1

ComputerProperties

This sequence of elements specifies which properties are required when retrieving a computer's properties.

xsd:string computerUUID

The UUID identifying the computer.

xsd:string computerLabel

The display name of the computer.

xsd:string computerDomainUUID

The UUID identifying the CA ITCM domain for the given computer.

xsd:string computerPrimaryNetworkAddress

The IP address of the given computer in 0.0.0.0 format.

xsd:string computerHostUUID

An agent-generated UUID that identifies the given computer.

xsd:string computerAssetSourceUUID

The UUID identifying the common CA asset to which this computer is linked.

xsd:string computerHostName

The hostname of the computer.

xsd:string computerSerialNumber

The serial number of the computer.

xsd:string computerDiscSerialNumber

The serial number of the computer disc.

xsd:string computerPrimaryMACAddress

The MAC address of the computer. The string should have a maximum length of 64 characters.

xsd:string computerAssetTag

A string value that identifies a computer asset.

xsd:long computerClassID

A numeric value indicating the class of the computer object; for example, Operating System.

UsageListMask computerUsageList

The computer usage list mask.

string computerURI

The fully-qualified Uniform Resource Identifier of the computer.

string computerSystemID

A user-defined identifier for the computer system.

string computerVendorName

The name of the vendor of this computer unit.

string computerDomainLabel

The display name of the CA ITCM domain to which this computer unit belongs.

string computerCreationUser

The name of the user account that created the computer unit's record.

DateTime computerCreationDate

The date and time on which the computer unit's record was created.

string computerLastUpdatedUser

The name of the user account that last updated the computer's properties.

DateTime computerLastUpdatedDate

The date and time on which the computer's properties were last updated.

long computerVersionNumber

The version number of the computer's record. A version number is incremented each time a record is updated.

Remarks

None

ComputerPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a computer's properties.

- xsd:boolean computerUUIDRequired
- xsd:boolean computerLabelRequired
- xsd:boolean computerDomainUUIDRequired
- xsd:boolean computerPrimaryNetworkAddressRequired

- xsd:boolean computerHostUUIDRequired
- xsd:boolean computerAssetSourceUUIDRequired
- xsd:boolean computerHostNameRequired
- xsd:boolean computerSerialNumberRequired
- xsd:boolean computerDiscSerialNumberRequired
- xsd:boolean computerPrimaryMACAddressRequired
- xsd:boolean computerAssetTagRequired
- xsd:boolean computerClassIDRequired
- xsd:boolean computerUsageListRequired
- xsd:boolean computerURIRequired
- xsd:boolean computerSystemIDRequired
- xsd:boolean computerVendorNameRequired
- xsd:boolean computerDomainLabelRequired
- xsd:boolean computerCreationUserRequired
- xsd:boolean computerCreationDateRequired
- xsd:boolean computerLastUpdatedUserRequired
- xsd:boolean computerLastUpdatedDateRequired
- xsd:boolean computerVersionNumberRequired

Remarks

For more details, see [ComputerProperties](#) (see page 669).

ComputerPropertyFilter

This sequence of elements specifies a search filter when searching for or listing computers.

ComputerProperty computerProperty

The identifier for the computer property on which the search is based.

xsd:string searchString

If the property to search on is a string property, this parameter specifies the value of the search filter to be applied.

This field is mandatory if computerProperty is set to one of the following values:

- COMPUTER_UUID
- COMPUTER_LABEL

- COMPUTER_DOMAIN_UUID
- COMPUTER_PRIMARY_NETWORK_ADDRESS
- COMPUTER_HOST_UUID
- COMPUTER_ASSET_SOURCE_UUID
- COMPUTER_HOST_NAME
- COMPUTER_SERIAL_NUMBER
- COMPUTER_DISC_SERIAL_NUMBER
- COMPUTER_PRIMARY_MAC_ADDRESS
- COMPUTER_ASSET_TAG
- COMPUTER_URI
- COMPUTER_SYSTEM_ID
- COMPUTER_VENDOR_NAME
- COMPUTER_DOMAIN_LABEL
- COMPUTER_CREATION_USER
- COMPUTER_LAST_UPDATED_USER

xsd:long searchValue

If the property to search on is a numeric property, this parameter specifies the value of the search filter to be applied. This field is mandatory if computerProperty is set to one of the following values:

- COMPUTER_CLASSID
- COMPUTER_USAGE_LIST
- COMPUTER_CREATION_DATE
- COMPUTER_LAST_UPDATED_DATE
- COMPUTER_VERSION_NUMBER

Remarks

None

ComputerStatusInfoRequired

xsd:bool bComputerNameRequired

Name of the computer.

xsd: bool bComputerTypeRequired

Type of the computer (OS type)

xsd: bool bComputerDescRequired

Description of the computer

xsd: bool bIPAddressRequired

Ip Address of the computer.

xsd: bool bComputerStatusCodeRequired

Status code returned during scan or deployment.

xsd: bool bComputerStatusCodeExRequired

Extended Status code.

xsd: bool bComputerStatusMsgRequired

Status message if any.

xsd:bool bPrimerExistsRequired

DmPrimer existence.

xsd:bool bCredentialsNecessaryRequired

Credentials necessary flag.

CreateAgentComponents

Members of sequence CreateAgenComponents:

xsd: boolean amAgent

AM agent needs to be created when set to true.

xsd: boolean amSWUsageAgent

AM software usage agent needs to be created when set to true.

xsd:boolean amPerformanceAgent

AM performance agent needs to be created when set to true.

xsd:boolean sdAgent

SD agent needs to be created when set to true.

xsd:boolean dtsAgent

DTS agent needs to be created when set to true.

xsd: boolean rcAgent

RC agent needs to be created when set to true.

CreateAgentComponents

Members of sequence CreateAgentComponentst:

xsd:boolean amAgent

AM agent needs to be created when set to true.

xsd:boolean amSWUsageAgent

AM software usage agent needs to be created when set to true.

xsd:boolean amPerformanceAgent

AM performance agent needs to be created when set to true.

xsd:boolean sdAgent

SD agent needs to be created when set to true.

xsd:boolean dtsAgent

DTS agent needs to be created when set to true.

xsd:boolean rcAgent

RC agent needs to be created when set to true.

CreateCatalogSoftwarePackageGroupProperties

This sequence of elements specifies the properties required for creating a new CatalogSoftwarePackageGroup.

xsd:string parentCatalogSoftwarePackageGroupId

The UUID identifying the parent of this new catalog software package group. The default value is the root Catalog folder.

xsd:boolean parentCatalogSoftwarePackageGroupIdSupplied

A flag indicating if the parent id of this new catalog software package group is supplied.

xsd:string name ("")

The name of the catalog software package group.

xsd:boolean nameSupplied ("")

A flag indicating if the name of the catalog software package group is supplied.

xsd:string unitGroupId

The UUID identifying the unit group to which the software packages in this catalog group is to be made available.

xsd:boolean unitGroupIdSupplied

A flag indicating if the unit group id is supplied.

xsd:string comment

A comment for the catalog software package group.

xsd:boolean commentSupplied

Indicates if a comment is supplied.

Remarks

None

CreateComputerProperties

Members of sequence CreateComputerProperties:

xsd:string computerLabel

Specifies computer label.

xsd:string computerHostName

Specifies computer host name. If not supplied computer label is used as host name.

xsd:boolean computerHostNameSupplied**xsd:string computerPrimaryNetworkAddress (m)**

Specifies the primary network address of the computer.

ComputerType computerType

Specifies type of computer. The default value is MACHINE.

xsd:boolean computerTypeSupplied (m)

True or false depending on the value of computerType.

xsd:long operatingSystemType (m)

Specifies type of the operating system.

xsd:string scalabilityServerName

If not supplied internally system fetches manager name and sets it as scalability server.

xsd:boolean scalabilityServerNameSupplied (m)

True or false depending on the value of scalabilityServerName.

xsd>CreateAgentComponents createAgentComponents

Mandatory parameter. At least one agent component must be Available.

xsd:SDAgentProperties sdAgentProperties.

Properties that is set for SDAgent. Needed only when SD agent needs to be created.

xsd:boolean SDAgentPropertiesSupplied (m)

True or false depending on the value of sdAgentProperties.

xsd:string user

Specifies user name who created the computer. The default value is blank.

xsd:boolean userSupplied (m)

True or false depending on the value of user.

xsd:string phone

Specifies phone number of user who created the computer.

xsd:boolean phoneSupplied (m)

True or false depending on the value of phone.

xsd:string location

Specifies location of user who created the computer. The default value is blank.

xsd:boolean locationSupplied (m)

True or false depending on the value of location.

xsd:string comment

Specifies any comment user who created the computer needs to set.

xsd:boolean commentSupplied (m)

True or false depending on the value comment.

xsd:string computerPrimaryMACAddress

Specifies the computer MAC Address.

xsd:boolean computerPrimaryMACAddressSupplied (m)

True or false depending on the value of computerPrimaryMACAddress.

xsd:string osimImage

Specifies the osim image.

xsd:boolean osimImageSupplied (m)

True or false depending on the value osimImage.

xsd:string osimBootServerName

If not specified, the scalability server name is taken as the OSIM boot server name.

xsd:boolean osimBootServerNameSupplied (m)

True or false depending on the value of osimBootServerName.

CreateComputerProperties2

Members of sequence CreateComputerProperties:

xsd:string computerHostName

Specifies computer host name.

xsd:string computerLabel

Specifies computer label. If its not supplied, hostname will be used for label.

xsd:boolean computerLabelSupplied

xsd:string computerPrimaryNetworkAddress

Specifies the primary network address of the computer.

xsd:long classID

Defines ClassID for specifying the OS type of the computer. See [ComputerClassID](#) (see page 50) for more information.

xsd:string computerPrimaryMACAddress

Specifies the computer MAC Address.

xsd:boolean computerPrimaryMACAddressSupplied

xsd:string serialNumber

The serial number of the computer.

xsd:string serialNumberSupplied

xsd:string assetTag

A string value that identifies a computer asset.

xsd:string assetTagSupplied

ComputerType computerType

Specifies type of computer. The default value is MACHINE.

xsd:boolean computerTypeSupplied

True or false depending on the value of computerType.

xsd:string scalabilityServerName

If not supplied internally system fetches manager name and sets it as scalability server.

xsd:boolean scalabilityServerNameSupplied

Specifies true or false based on the value of scalabilityServerName.

CreateAgentComponents createAgentComponents

Mandatory parameter. At least one agent component must be Available.

xsd:boolean scalabilityServerNameSupplied

Specifies true or false based on the value of scalabilityServerName.

xsd:string description

Defines in brief, information about scalability and agent selection for the registered computer.

xsd:SDAgentProperties sdAgentProperties.

Defines the Properties that are set for SDAgent. Needed only at the time of SD agent creation.

xsd:boolean SDAgentPropertiesSupplied (m)

True or false depending on the value of sdAgentProperties.

xsd:string user

Specifies user name who created the computer. The default value is blank.

xsd:boolean userSupplied (m)

True or false depending on the value of user.

xsd:string phone

Specifies phone number of user who created the computer.

xsd:boolean phoneSupplied (m)

True or false depending on the value of phone.

xsd:string location

Specifies location of user who created the computer. The default value is blank.

xsd:boolean locationSupplied (m)

Specifies true or false based on the value of location.

xsd:string comment

Specifies any comment user who created the computer needs to set.

xsd:boolean commentSupplied (m)

Specifies true or false based on the value comment.

CreateSoftwareJobContainerProperties2

- xsd:string name (m)
- xsd:string comment
- xsd:boolean commentSupplied
- xsd:long priority

Default: 5

Limits: 1 to 10

- `xsd:long prioritySupplied`
- `xsd:string parentSoftwareJobContainerId`
Default: `SD_Jobs`
- `xsd:boolean parentSoftwareJobContainerIdSupplied`
- `SoftwareJobContainerPropertyMask`
`softwareJobContainerPropertyMask`
Default: `linkageBatch` and `Cascade`.
- `xsd:boolean softwareJobContainerPropertyMaskSupplied`

CreateSoftwareJobContainerProperties3

The `CreateSoftwareJobContainerProperties3` sequence of elements specifies which job container properties are to be initialized when a software job container is created.

xsd:string parentSoftwareJobContainerId

Specifies the parent software job container ID. If no value is provided, the root “`__SD_Jobs`” is used as the parent.

xsd:boolean parentSoftwareJobContainerIdSupplied

Specifies true or false depending on whether the `parentSoftwareJobContainerId` parameter is supplied.

xsd:string name (^m)

Specifies the name of the software job container.

xsd:string comment

Specifies the comment for the software job container.

xsd:boolean commentSupplied

Specifies true or false depending on whether the comment parameter is supplied.

xsd:long priority

Specifies the priority that can range from 1 (highest) to 10 (lowest), If no value is provided, the default value, 5, is used.

xsd:long prioritySupplied

Specifies true or false depending on whether the priority parameter is supplied.

SoftwareJobContainerPropertyMask**softwareJobContainerPropertyMask**

Specifies the mask for the software job container property. If no value is provided, the default value, linkageBatch and Cascade, is used.

xsd:boolean softwareJobContainerPropertyMaskSupplied

Specifies true or false depending on whether the softwareJobContainerPropertyMask parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is provided.

Remarks

None

CreateSoftwareJobOrderPolicyProperties

This sequence of elements specifies which properties are required when creating a job order for a software job within a software policy.

xsd:long timeoutHours

The time in hours after which time, if the job has been unable to complete, the job will cease attempts to complete. If not supplied, the default value is 48.

xsd:boolean timeoutHoursSupplied

SoftwareJobStartupTime startWhen

Properties that indicate when the software job should be started. If not supplied, the default value is EXACT_LOCAL_TIME.

xsd:boolean startWhenSupplied**xsd:string jobName**

The name of the software job with which this order is associated. If no name is specified, Unicenter Software Delivery creates a default name.

xsd:boolean jobNameSupplied

SoftwareJobBootMask bootBeforeMask

Indicates the reboot options requested for the software job before the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootBeforeMaskSupplied

SoftwareJobBootMask bootAfterMask

This element indicates the reboot options requested for the software job after the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootAfterMaskSupplied

SoftwareJobOperationMask operationMask

Properties for the operation of the software job. If not supplied, the default value is calendarControl.

xsd:boolean operationMaskSupplied**xsd:string userParameters**

Contains user-defined parameters.

xsd:boolean userParametersSupplied**xsd:long promptTimeoutHours**

This indicates the number of hours for which a user prompt should be displayed before forced execution of the job. The default value is 1 day. This parameter should be set to a value that does not exceed the timeout value for the job.

If not supplied, the default value is 48 hours.

xsd:boolean promptTimeoutHoursSupplied

xsd:string deliveryCalendar

The name of a Unicenter delivery calendar to be used for scheduling the job. The calendar controls the days, dates, and times when delivery is permitted. The time cells not set in the Unicenter Calendar define the exclusion intervals.

If the name of the calendar is unknown to Unicenter, the job is not created.

xsd:boolean deliveryCalendarSupplied

Remarks

None

CreateSoftwareJobOrderPolicyProperties3

The CreateSoftwareJobOrderPolicyProperties3 sequence of elements specifies which properties are required when creating a job order for a software job within a software policy.

xsd:long timeoutHours

Specifies the time in hours after which, if the job has been unable to complete, the job will cease attempts to complete. If you do not specify any value, the default value 48 is used.

xsd:boolean timeoutHoursSupplied

Specifies true or false depending on whether the timeoutHours parameter is supplied.

SoftwareJobStartupTime startWhen

Specifies the properties that specify when to start the software job. If you do not specify any value, the default value EXACT_LOCAL_TIME is used.

xsd:boolean startWhenSupplied

Specifies true or false depending on whether the startWhen parameter is supplied.

xsd:string jobName

Specifies the name of the software job with which this order is associated. If you do not specify any name, SD creates a default name.

xsd:boolean jobNameSupplied

Specifies true or false depending on whether the jobName parameter is supplied.

SoftwareJobBootMask3 bootBeforeMask

Specifies the restart options requested for the software job before executing the job. If you do not supply any value, the default value noReboot is used.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwareJobBootMask3 bootAfterMask

Specifies the restart options requested for the software job after executing the job. If you do not supply any value, the default value noReboot is used.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

SoftwareJobOperationMask operationMask

Specifies the properties for the operation of the software job. If you do not specify any value, the default value calendarControl is used.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask parameter is supplied.

xsd:string userParameters

Specifies the user-defined parameters.

xsd:boolean userParametersSupplied

Specifies true or false depending on whether the userParameters parameter is supplied.

xsd:long promptTimeoutHours

Specifies the number of hours for which to display a user prompt before the job is forced to start. The default value is 24 hours. You must set this parameter to a value that does not exceed the timeout value for the job (default value is 48 hours).

xsd:boolean promptTimeoutHoursSupplied

Specifies true or false depending on whether the promptTimeoutHours parameter is supplied.

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to use for scheduling the job. The calendar controls the days, dates, and times of the scheduled delivery. The time cells that are not set in the calendar define the exclusion intervals. If the name of the calendar is unknown to SD, the job is not created.

xsd:boolean deliveryCalendarSupplied

Specifies true or false depending on whether the deliveryCalendar parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

Remarks

None

CreateSoftwareJobOrderProperties

This sequence of elements specifies which software job order properties is to be initialized when a software job order is created.

- DateTime executionDate
Default: Current date and time of the manager computer.
- xsd:boolean executionDateSupplied
- xsd:long timeoutHours
Default: 168 hours.
- xsd:boolean timeoutHoursSupplied
- SoftwareJobStartupTime startWhen
Default: EXACT_LOCAL_TIME.

- xsd:boolean startWhenSupplied
- xsd:string jobName (^m)
- xsd:boolean jobNameSupplied (^m)
- SoftwareJobBootMask bootBeforeMask
Default: noReboot
- xsd:boolean bootBeforeMaskSupplied
- SoftwareJobBootMask bootAfterMask
Default: noReboot
- xsd:boolean bootAfterMaskSupplied
- SoftwareJobOperationMask operationMask
Default: calendarControl.
- xsd:boolean operationMaskSupplied
- xsd:string userParameters
- xsd:boolean userParametersSupplied
- xsd:long promptTimeoutHours
Default: 24 hours.
- xsd:boolean promptTimeoutHoursSupplied
- xsd:string deliveryCalendar
- xsd:boolean deliveryCalendarSupplied
- DateTime deliveryDateTime
Default: Current date and time of the manager computer.
- xsd:boolean deliveryDateTimeSupplied

Remarks

For more details, see [SoftwareJobOrderProperties](#) (see page 823).

CreateSoftwareJobOrderProperties3

The CreateSoftwareJobOrderProperties3 sequence of elements specifies the software job order properties that are to be initialized when a software job order is created.

DateTime executionDate

Specifies the execution date. If not supplied, the default value is the current date and time of the manager computer.

xsd:boolean executionDateSupplied

Specifies true or false depending on whether the executionDate parameter is supplied.

xsd:long timeoutHours

Specifies the time in hours after which, if the job has been unable to complete, the job will cease attempts to complete. If not supplied, the default value is set to 168 hours.

xsd:boolean timeoutHoursSupplied

Specifies true or false depending on whether the timeoutHours parameter is supplied.

SoftwareJobStartupTime startWhen

Specifies the time to start the software job. If not supplied, the default value is EXACT_LOCAL_TIME.

xsd:boolean startWhenSupplied

Specifies true or false depending on whether the startWhen parameter is supplied.

xsd:string jobName (")

Specifies the name of the software job with which this order is associated. If no name is specified, SD creates a default name

xsd:boolean jobNameSupplied (")

Specifies true or false depending on whether the jobName parameter is supplied.

SoftwareJobBootMask3 bootBeforeMask

Specifies the reboot options requested for the software job before the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwareJobBootMask3 bootAfterMask

Specifies the reboot options requested for the software job after the job is executed. If not supplied, the default value is noReboot.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

SoftwareJobOperationMask operationMask

Specifies the properties for the operation of the software job. If not supplied, the default value is calendarControl.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask parameter is supplied.

xsd:string userParameters

Specifies the user-defined parameters

xsd:boolean userParametersSupplied

Specifies true or false depending on whether the userParameters parameter is supplied.

xsd:long promptTimeoutHours

Specifies the number of hours for which a user prompt must be displayed before forced execution of the job starts. If not supplied, the default value is 24 hours.

xsd:boolean promptTimeoutHoursSupplied

Specifies true or false depending on whether the promptTimeoutHours parameter is supplied.

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to be used for scheduling the job. The calendar controls the days, dates, and times when delivery is permitted. The time cells that are not set in the calendar define the exclusion intervals. If the name of the calendar is unknown to SD, the job is not created.

xsd:boolean deliveryCalendarSupplied

Specifies true or false depending on whether the deliveryCalendar parameter is supplied.

DateTime deliveryDateTime

Specifies the delivery date and time. If not supplied, the default value is the current date and time of the manager computer.

xsd:boolean deliveryDateTimeSupplied

Specifies true or false depending on whether the deliveryDateTime parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

Remarks

None

CreateSoftwarePackageGroupProperties

This sequence of elements specifies which software package group properties are to be initialized when a software package group is created.

- **xsd:string parentSoftwarePackageGroupId**
Default: SD_Software Library
- **xsd:boolean parentSoftwarePackageGroupIdSupplied**
- **xsd:string name** (^m)
- **xsd:boolean nameSupplied** (^m)

- xsd:string comment
- xsd:boolean commentSupplied

Remarks

For more details, see [SoftwarePackageGroupProperties](#) (see page 828).

CreateSoftwarePackageProperties

This sequence of elements specifies which software package properties are to be initialized when a software package is created.

- xsd:string softwarePackageName (^m)
- xsd:boolean softwarePackageNameSupplied (^m)
- xsd:string softwarePackageVersion (^m)
- xsd:boolean softwarePackageVersionSupplied (^m)
- xsd:string supplier
- xsd:boolean supplierSupplied
- xsd:string comment
- xsd:boolean commentSupplied
- xsd:string basedOnSoftwarePackageId
- xsd:boolean basedOnSoftwarePackageIdSupplied
- SoftwarePackageMask softwarePackageMask
Default: checksum.
- xsd:boolean softwarePackageMaskSupplied

Remarks

For more details, see [SoftwarePackageProperties](#) (see page 847).

CreateSoftwarePackageVolumeProperties

This sequence of elements specifies which volume properties are to be initialized when a software package volume is created.

- xsd:string volumeName (^m)
- xsd:boolean volumeNameSupplied (^m)
- SourceMedium volumeType
Default: HardDisk

- `xsd:boolean volumeTypeSupplied`
- `xsd:string softwarePackageId` ^(m)
- `xsd:boolean softwarePackageIdSupplied` ^(m)

Remarks

For more details, see [SoftwarePackageVolumeProperties](#) (see page 836).

DistributionContainerProperties

xsd:string distributionContainerId

The UUID identifying the distribution container.

xsd:string distributionContainerName

The display name of the distribution container.

xsd:long priority

DistributionStatus distributionStatus

The status of the distribution.

DistributionPermissionMask distributionPermissionMask

The mask that tells the operations allowed on a distribution container in current state.

DateTime regDateTime

The Date and Time when the distribution container was created.

DateTime distDateTime

The Date and Time when the distribution is scheduled to send.

DateTime haltDateTime

The Date and Time when the distribution is scheduled to halt.

NoOfDistributionDomains noOfDistributionDomains

Total number of target domains and their state.

xsd:string ahdInfo

Advanced help desk information.

DistributionContainerPropertiesRequired

- xsd:boolean distributionContainerIdRequired
- xsd:boolean distributionContainerNameRequired
- xsd:boolean priorityRequired
- xsd:boolean distributionStatusRequired
- xsd:boolean distributionPermissionMaskRequired
- xsd:boolean regDateTimeRequired
- xsd:boolean distDateTimeRequired
- xsd:boolean haltDateTimeRequired
- xsd:boolean noOfDistributionDomainsRequired
- xsd:boolean ahdInfoRequired

DistributionPermissionMask

- xsd:boolean deleteAllowed;
- xsd:boolean haltAllowed;
- xsd:boolean renewAllowed;
- xsd:boolean resumeAllowed;
- xsd:boolean sendAllowed;

DistributionContainerOrderPropertiesRequired

xsd:boolean distributionContainerOrderIdRequired (m)

True or false depending on whether distribution container order Id is required.

xsd:boolean distributionContainerOrderNameRequired (m)

True or false depending on whether distribution container order name is required.

xsd:boolean distributionContainerOrderTypeRequired (m)

True or false depending on whether distribution container order type is required.

xsd:boolean jobContainerPropertiesRequired (m)

True or false depending on whether job container properties is required.

xsd:boolean fetchItemOrderPropertiesRequired (m)

True or false depending on whether fetch item order properties is required.

CreateSoftwarePolicyProperties2

Members of sequence CreateSoftwarePolicyProperties2:

xsd:string policyName (m)

Name of policy to be created.

xsd:string comment

The comment to be set for the policy.

xsd:boolean commentSupplied (m)

True or false depending on the value of comment.

xsd:long priority

Can be 1(highest) to 10(lowest), If not supplied, the default value is 5.

xsd:boolean prioritySupplied (m)

True or false depending on whether priority value is supplied by user or not.

CreateSoftwareJobLinkage createSoftwareJobLinkage

The sequence *CreateSoftwareJobLinkage* is provided for setting the software job linkage related properties. .

xsd:boolean createSoftwareJobLinkageSupplied (m)

True or false depending on the value of createSoftwareJobLinkage.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

Specifies the type of evaluation that will done when the policy is applied to the group. The default value is SP_ POPULATEEXCEPTIONS

xsd:boolean softwarePolicyEvaluationTypeSupplied (m)

True or false depending on the value of softwarePolicyEvaluationType.

SoftwarePolicyEvaluationPeriodType

softwarePolicyEvaluationPeriodType

Specifies whether the software policy will be evaluated once or periodically. If periodic then the evaluation is done as per the evaluation period set. The default value is once.

xsd:boolean softwarePolicyEvaluationPeriodTypeSupplied (m)

True or false depending on the value for softwarePolicyEvaluationPeriodType.

SoftwarePolicyEvaluationPeriodType
softwarePolicyEvaluationPeriodType

Optional parameter. Specifies whether the software policy will be evaluated once or periodically. If periodic then the evaluation is done as per the evaluation period set. The default value is once.

xsd:boolean softwarePolicyEvaluationPeriodTypeSupplied (m)

True or false depending on the value for softwarePolicyEvaluationPeriodType.

DateTime nextEvaluationDate

Specifies the date and time for next evaluation. The default value is current date and time in manager machine.

xsd:boolean nextEvaluationDateSupplied (m)

True or false depending on the value of nextEvaluationDate.

xsd:long evaluationPeriod

Specifies the period of evaluation, in hours. default value is 1440 minutes or 24 hours.

xsd:boolean evaluationPeriodSupplied (m)

True or false depending on the value of evaluationPeriod.

xsd:boolean enterpriseTime

If set to true, the time specifications are based on the enterprise manager time. By default it is set to false;

xsd:boolean enterpriseTimeSupplied (m)

True or false depending on the value of enterpriseTimeSupplied.

xsd:string evaluationCalendar

Specifies the name of the calendar that controls the time when the evaluation can take place. To activate a evaluationCalendar you must specify "evaluationPeriod". The default value is an empty string.

xsd:boolean evaluationCalendarSupplied (m)

True or false depending on the value of evaluationCalendarSupplied.

AdhocTargetEvaluation adhocTargetEvaluation

Target evaluation is triggered when agents report inventory to the manager. This parameter specifies the type of target evaluation. The default value is NO_ADHOC.

xsd:boolean adhocTargetEvaluationSupplied (m)

True or false depending on the value of adhocTargetEvaluation.

xsd:boolean regenerate

This parameter is considered only if evaluationType is SP_SETUPJOBS or SP_SETUPANDEXECUTEJOBS. The default value is true.

xsd:boolean regenerateSupplied (m)

True or false depending on the value of regenerate.

xsd:boolean evaluateDynamicGroups

(Optional). If set to true, dynamic groups are evaluated during evaluation of the software policy. The default value is false.

xsd:boolean evaluateDynamicGroupsSupplied (m)

True or false depending on the value of evaluateDynamicGroups.

CreateUnitGroupPropertiesByName

Members of sequence CreateUnitGroupPropertiesByName:

xsd:string groupLabel (m)

Specifies the name of the group.

xsd:string groupDescription

Specifies the description of the group.

xsd:boolean groupDescriptionSupplied (m)

True or false depending on the groupDescription value.

xsd:string queryName

Specifies the name of the query to be linked to the newly created group.

xsd:boolean queryNameSupplied (m)

True or false depending on the value of queryName.

xsd:string engineName

Specifies the name of the query evaluation engine to be associated with the unit group; the unit group is associated with All Engines.

xsd:boolean engineNameSupplied (m)

True or false depending on the engineName value.

xsd: long evaluationPeriod

Period in which the query is evaluated. The period is specified in minutes. The default value is 1440 minutes Maximum value allowed is 10080.

xsd:boolean evaluationPeriodSupplied (m)

True or false depending on the value of evaluationPeriod.

CreateUserProfileProperties

Members of sequence CreateUserProfileProperties:

xsd:string userProfileName (m)

Specifies the name of user profile in the format <ComputerName>/<UserName>.

xsd:long hostComputerOperatingSystemType (m)

Specifies the operating system of the host computer.

SDAgentProperties sdAgentProperties

Specifies the properties to be set for SD agent.

xsd:boolean sdAgentPropertiesSupplied (m)

True or false depending on the value of sdAgentProperties.

xsd:string user

Specifies the user name. The default value is blank.

xsd:boolean userSupplied (m)

True or false depending on the value of user.

xsd:string phone

Specifies the user's phone number.

xsd:boolean phoneSupplied (m)

True or false depending on the value of phone.

xsd:string location

Specifies the location of the user. The default value is blank.

xsd:boolean locationSupplied (m)

True or false depending on the value of location.

xsd:string comment

Specifies the comment.

xsd:boolean commentSupplied (m)

True or false depending on the value comment.

DateTime

This sequence of elements represents a date and time structure.

- `xsd:int year`
- `xsd:int month`
- `xsd:int dayOfMonth`
- `xsd:int hour`
- `xsd:int minute`

Remarks

None

DeploymentPackageProperties

`xsd:string productName`

Name of the product

`xsd:string packageName`

Name of the Package

`xsd:string packageVersion`

Package version

`xsd:string productDisplayName`

Product display name.

`xsd:string packageDisplayName`

Package display name

`xsd:string packageVersionExtended`

Extended form of package version.

`xsd:string displaySortOrder`

Display sort order.

`xsd:string platform`

Package platform

`xsd:string language`

Package language

`xsd:string description`

Package description

`xsd:string windowsCommand`

Windows installation command

xsd:string unixCommand

Unix installaton command

xsd:string operatingSystemType

OperatingSystemType

xsd:string msiProductCode

MSI product code.

xsd:string pifProductCode

PIF product code.

dsm:ArrayOfParameterInfo arrayOfParamInfo

Deployment Package parameters list

DeploymentPackagePropertiesRequired

Members of Sequence DeploymentPackagePropertiesRequired

- xsd:bool productNameRequired
- xsd:bool packageNameRequired
- xsd:bool packageVersionRequired
- xsd:bool productDisplayNameRequired
- xsd:bool packageDisplayNameRequired
- xsd:bool packageVersionExtRequired
- xsd:bool displaySortOrderRequired
- xsd:bool platformRequired
- xsd:bool languageRequired
- xsd:bool descriptionRequired
- xsd:bool windowsCommandRequired
- xsd:bool unixCommandRequired
- xsd:bool operatingSystemTypeRequired
- xsd:bool MSIProductCodeRequired
- xsd:bool PIFProductCodeRequired
- xsd:bool parameterInfoRequired

DeploymentInfo

xsd:string targetUserName

Name of the user to use for deployment.

xsd:bool bTargetUserSupplied

Target user supplied flag

xsd:string targetPassword

Password pertaining to the user to be used for deployment.

xsd:bool bTargetPwdSupplied

Target password supplied flag.

xsd:string scalabilityServer

Scalability Server. If specified, then the deployment happens from it.

xsd:bool bScalabilityServerSupplied

Scalability server supplied flag

xsd:string scalabilityServerUserName

Name of the user to connect to the scalability server.

xsd:bool bSSUserNameSupplied

Scalability server user name supplied flag

xsd:string scalabilityServerPassword

Password of the scalability server user.

xsd:bool bSSPasswordSupplied

Scalability server user password supplied flag

dsm:ArrayOfMachineCredentials arrayOfTargetMachineCredentials

Array of MachineCredentials structure which contains credential information for each target machine.

xsd:bool bTargetMachCredentialsSupplied

Array of target machine credentials supplied flag

DeploymentPackageInfo

xsd:string productName

Product name pertaining to the package.

xsd:string packageName

Package name.

xsd:string packageVersion

Version of the package.

xsd:bool bPackageVersionSupplied

Package version supplied flag

xsd:string platform

Package platform.

xsd:bool bPlatformSupplied

Platform supplied flag

xsd:string language

Language of the specified package.

xsd:bool bLanguageSupplied

Language supplied flag

dsm:ArrayOfParameterValues arrayOfParameterValues

Array of ParameterValue structures which contain values of the parameters that are needed for package installation.

xsd:bool bParamValuesSupplied

ParameterValues supplied flag.

DiscoveredSoftwareProperties

This sequence defines the properties that you can define for discovered software.

xsd:string softwareTitle

The title of the discovered software item.

xsd:string softwareVersion

The version of the discovered software item.

DiscoveredSoftwareType softwareType

The type of the discovered software item.

DiscoveredSoftwareSourceType sourceType

The type of the discovery source for the discovered software item

Remarks

None

DiscoveredSoftwarePropertiesRequired

This sequence of elements specifies which properties are required when retrieving a discovered software item's properties.

- xsd:boolean softwareTitleRequired
- xsd:boolean softwareVersionRequired
- xsd:boolean softwareTypeRequired
- xsd:boolean sourceTypeRequired

Remarks

For more details, see [DiscoveredSoftwareProperties](#) (see page 700).

DistributionContainerOrderProperties

xsd:string distributionContainerOrderId

The UUID of the distribution container order

xsd:string distributionContainerOrderName

The name of the distribution container order.

DistributionContainerOrderType distributionContainerOrderType

Type of the distribution container order.

SoftwareJobContainerProperties jobContainerProperties

Job container properties. This field is populated only if distributionContainerOrderType = REGISTER_JOB_CONTAINER.

FetchItemOrderProperties fetchItemOrderProperties

FetchItemOrder properties. This field is populated only if distributionContainerOrderType = FETCH_ITEM.

DistributionContainerOrderProperties3

The DistributionContainerOrderProperties3 sequence of elements provides the distribution container order properties.

xsd:string distributionContainerOrderId

Specifies the UUID of the distribution container order.

xsd:string distributionContainerOrderName

Specifies the name of the distribution container order.

DistributionContainerOrderType distributionContainerOrderType

Specifies the type of the distribution container order.

SoftwareJobContainerProperties3 jobContainerProperties3

Specifies the job container properties. This field is populated only if distributionContainerOrderType = REGISTER_JOB_CONTAINER.

FetchItemOrderProperties fetchItemOrderProperties

Specifies the FetchItemOrder properties. This field is populated only if distributionContainerOrderType = FETCH_ITEM.

DistributionDomainProperties

xsd:string distributionDomainId

The UUID of the distribution domain.

xsd:string domainId

The UUID of the target domain.

xsd:string domainName

The display name of the target domain.

xsd:string domainDescription

Specifies the description of the domain.

xsd:long percentSent

Indicates how many percent of the container file (size) has been transferred to the actual domain.

DistributionStatus distributionStatus

The distribution status.

xsd:string distributionMessage

A status text that explains the current status of the container file transfer. (Can be empty).

xsd:long distributionErrorCode

If distributionStatus is DIST_ERRORED, a detailed error code that is explained by the distributionMessage.

DistributionOrderStatus distributionOrderStatus

An overall order status.

xsd:string distributionOrderMessage

A status text that explains the overall order status. (Can be empty).

xsd:long distributionOrderErrorCode

If distributionOrderStatus is DIST_ORDER_ERROR, a detailed error code that is explained by the distributionOrderMessage.

xsd:long distributionFailedOrderNumber

If distributionOrderStatus is DIST_ORDER_ERROR, this is number of the order in the container that failed. All subsequent orders have not been executed.

DistributionDomainPropertiesRequired

- xsd:boolean distributionDomainIdRequired;
- xsd:boolean domainIdRequired;
- xsd:boolean domainNameRequired;
- xsd:boolean domainDescriptionRequired;
- xsd:boolean percentSentRequired;
- xsd:boolean distributionStatusRequired;
- xsd:boolean distributionMessageRequired;
- xsd:boolean distributionErrorCodeRequired;
- xsd:boolean distributionOrderStatusRequired;
- xsd:boolean distributionOrderMessageRequired;
- xsd:boolean distributionOrderErrorCodeRequired;
- xsd:boolean distributionFailedOrderNumberRequired;

DomainGroupProperties

This sequence of elements specifies the properties of a domain group.

xsd:string groupUUID

The UUID identifying the domain group.

xsd:string groupLabel

The name of the domain group.

xsd:string groupDescription

Describes the domain group.

GroupInheritance groupInheritance

This flag indicates the nature of the inheritance of permissions for members of the group.

xsd:string creationUser

Specifies the domain created user.

DateTime creationdate

The date and time on which the domain group was created.

xsd:string lastUpdatedUser

The name of the user who last updated the group's properties.

DateTime lastUpdatedDate

The date and time on which the domain group's properties were last updated.

DomainGroupPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a domain group's properties.

xsd:boolean groupUUIDRequired

True or false depending on whether group UUID needs to be fetched.

xsd:boolean groupLabelRequired

True or false depending on whether group label needs to be fetched.

xsd:boolean groupInheritanceRequired

True or false depending on whether group inheritance needs to be fetched.

xsd:boolean groupDescriptionRequired

True or false depending on whether group description needs to be fetched.

xsd:boolean creationUserRequired

True or false depending on whether creation user needs to be fetched.

xsd:boolean creationdateRequired

True or false depending on whether creation date needs to be fetched.

xsd:boolean lastUpdatedUserRequired

True or false depending on whether last updated user needs to be fetched.

xsd:boolean lastUpdatedDateRequired

True or false depending on whether last updated date needs to be fetched.

DomainProperties

xsd:string domainUUID

Specifies the domain UUID

domainType

Specifies domain is a domain or enterprise.

xsd:string parentDomainUUID

Specifies the UUID of the parent of the domain.

xsd:string domainLabel

Specifies the label of the domain.

xsd:string domainDescription

Specifies the description of the domain.

xsd:string creationUser

Specifies the domain created user.

Date creationDate

Specifies the date the domain was created.

xsd:string lastUpdatedUser

Specifies the last updated user name.

Date lastUpdatedDate

Represents the date on which the domain was modified.

xsd:string contactInformation

Specifies the contact information for the domain

xsd:string disHWUUID

Specifies the UUID of the discovered hardware.

xsd:string dbHostName

Specifies the database host name.

xsd:string dbServer

Specifies the database server

xsd:string dbType

Specifies the database type.

xsd:string dbName

Specifies the database name.

xsd:string dbUserName

Specifies the database user name.

xsd:string dbInstance

Specifies the database instance.

xsd:string domainNumber

Specifies the domain number; used by AM component.

dsm:UsageListMask usageList

Binary bits represents the components installed on the domain.

MachineCredentials

xsd:string targetMachine

Target machine name or IP.

xsd:string userName

User name with access to the target machine.

xsd:string password

Password of the user with access to the target machine.

NoOfDistributionDomains

xsd:long totalNoOfDomains

Total number of target domains.

xsd:long noOfDomainsWaiting

The number of target domains to those distribution is not yet activated.

xsd:long noOfDomainsActive

The number of target domains to those distribution is activated.

xsd:long noOfDomainsFailed

The number of target domains to those distribution is failed.

xsd:long noOfDomainsOK

The number of target domains to those distribution is successfully done.

ParameterInfo

Members of Sequence PackageParameters

xsd:string paramName

Name of the parameter.

xsd:int paramType

Parameter type.

Valid values are:

- 0 – Alphanumeric
- 1 – String
- 2 – Password type.

xsd:string paramValue

Parameter value. This needs to be set according to the above type when supplied as input for DeployPackage method.

xsd:string paramDescription

Description of the parameter.

xsd:string formatValidation

A pattern matching string used to validate the entered parameter.

ParameterValue

xsd:string paramName

Parameter name whose value needs to be supplied for deployment of the scanned package.

xsd:string value

Corresponding value depending on the type of the parameter

QueryFragment

This sequence of elements represents one fragment of a CA ITCM query definition. It describes a search statement.

QueryFragmentType queryFragmentType

The type of the query fragment, is it a database or directory statement?

xsd:string pseudoStatement

A user-defined description of the nature of the query fragment's statement.

xsd:string statement

The underlying SQL-style statement from which the query fragment is formulated.

Remarks

None

QueryResultProperties

This sequence of elements specifies the properties available when retrieving a query result object's properties.

xsd:long objectClass

The class of the object returned as part of a query result.

xsd:string objectUUID

The UUID identifying the object returned as part of a query result.

Remarks

None

QueryResultPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a query result object's properties.

- `xsd:boolean objectClassRequired`
- `xsd:boolean objectUUIDRequired`

Remarks

For more details, see [QueryResultProperties](#) (see page 708).

ResumeDeliveryDistributionContainerProperties

`xsd:string distributionContainerId (m)`

Specifies the UUID of the distribution container to be resumed.

`DateTime distributionDateTime`

Specifies the time to resume the associated distribution to the addressed domain managers. The default value is the current date and time of the enterprise manager computer.

`xsd:boolean distributionDateTimeSupplied`

True or false depending on whether value of distribution datetime is supplied.

`DateTime haltDateTime`

Specifies the date and time when a distribution order should be halted if still ongoing. If not supplied, the comstore parameter `DistMaxDistTime` is used.

`xsd:boolean haltDateTimeSupplied`

True or false depending on whether value of halt datetime is supplied.

ExportSoftwarePkgProperties

`xsd:string targetPath`

Specifies the target directory path where the software package is exported. The path can be a local or a network path.

`xsd:boolean targetOnServer`

To export the software package to the manager, set the field to 1.

To export the software package to the computer which has WebServices installed, set the field to 0.

`PackageExportType exportAs`

Specifies if the software package is exported as a Software Package or a Library Image.

FetchItemJobOrderProperties

xsd:string fetchItemJobOrderName

FetchItemJobOrder Name

xsd:string fetchItemJobOrderPath

FetchItemJobOrder Path

xsd:string fetchItemJobOrderComment

FetchItemJobOrder Comment

xsd:long orderPosition

This is the position the order will get in the container and the order it is handled at the Domain manager. 0 means last. 1 means first. If the value is greater than the number of orders then it is added last in the container. By default, the position is 0 (that is, at the end).

xsd:boolean orderPositionSupplied

FetchItemOrderProperties

xsd:string fetchItemOrderName

Name of the FetchItemOrder

xsd:string fetchItemOrderPath

Path of the FetchItemOrder

xsd:string fetchItemOrderComment

Comment of the FetchItemOrder

FileSystemElementPath

This sequence of values indicates the location of a file system element given its name and path.

- **xsd:string elementName**
- **xsd:string elementPath**

Remarks

None

GetSecurityAreaConfiguration

ResponseByName

xsd:bool SecurityAreaSupport

True or false depending upon whether global security area is enabled or not.

dsm:ArrayOfSecurityAreasByName securityAreas

Array of SecurityAreaNames and SecurityAreaEnable values.

GetSecurityGroupPermissions

xsd:string securityGroupUUID

UUID of the security group.

Xsd:string securityObjectOwner

Owner of the security object.

dsm:InheritanceLevel inheritLevel

Inheritance level of the security group.

**dsm:ArrayOfSecurityGroupProfilePermission
arrayOfSecurityGroupProfilePermission**

An array of [SecurityGroupProfilePermission](#) (see page 724) – one for each security profile.

GetSecurityObjectsPermissions

xsd:string securityObjectUUID

UUID of the security object.

xsd:string securityObjectOwner

Owner of the security object.

dsm:InheritanceLevel inheritLevel

Inheritance level of the security group.

**dsm:ArrayOfSecurityObjectProfilePermission
arrayOfSecurityObjectProfilePermission**

An array of SecurityObjectProfilePermission – one for each security profile.

ImportSoftwarePackageProperties

This sequence of elements specifies which software package properties are expected when importing a software package.

- **xsd:string regInfoPath** (^m)
- **xsd:boolean regInfoPathSupplied** (^m)
- **xsd:string sourcePath** (^m)

- xsd:boolean sourcePathSupplied (m)
- xsd:string basedOnSoftwarePackageld
- xsd:boolean basedOnSoftwarePackageldSupplied

Remarks

For more details, see [SoftwarePackageProperties](#) (see page 847).

ImportSoftwarePackageProperties2

Members of sequence: ImportSoftwarePackageProperties2

xsd:string regInfoPath

Specifies the directory on the server where the registration information subdirectory is located. By default, the source path is taken as a location of the registration information.

xsd:boolean regInfoPathSupplied (m)

True or false depending on the value of regInfoPath.

xsd:string sourcePath (m)

Directory on the server where an image of the item being registered is located.

xsd:string basedOnSoftwarePackageld

Specifies the original software package Id. The default value is an empty string.

xsd:boolean basedOnSoftwarePackageldSupplied (m)

True or false depending on the value of basedOnSoftwarePackageld.

xsd:boolean sourceOnManager

Specifies whether the source image is on the manager. Default is false.

xsd:boolean sourceOnManagerSupplied (m)

True or false depending on the value of sourceOnManager.

ImportSoftwarePackagePropertiesByName

Members of sequence: ImportSoftwarePackagePropertiesByName

xsd:string regInfoPath

Specifies the directory on the server where the registration information subdirectory is located. By default the source path is taken as a location of the registration information.

xsd:boolean regInfoPathSupplied (m)

True or false depending on the value of regInfoPath.

xsd:string sourcePath (m)

Directory on the server where an image of the item being registered is located.

xsd:string basedOnSoftwarePackageName

Specifies the source software package name. The default value is an empty string.

xsd:boolean basedOnSoftwarePackageNameSupplied (m)

True or false depending on the value of basedOnSoftwarePackageName.

xsd:string basedOnSoftwarePackageVersion

Specifies the version of the source software package. The parameter is mandatory if parameter basedOnSoftwarePackageName is specified.

xsd:boolean basedOnSoftwarePackageVersionSupplied (m)

True or false depending on the value of basedOnSoftwarePackageVersion.

xsd:boolean sourceOnManager

Specifies whether the source library image is on the manager. The default value is false.

xsd:boolean sourceOnManagerSupplied (m)

True or false depending on the value of sourceOnManager.

QueryFragment2

This sequence of elements represents one fragment of a DSM query definition. It describes a search statement.

QueryFragmentType2 queryFragmentType

Type of the query fragment; specifies whether it is a database or directory statement.

xsd:string pseudoStatement

A user-defined description of the nature of the query fragment's statement.

xsd:string statement

The underlying SQL-style statement from which the query fragment is formulated.

xsd:string linkQueryName

Name of the existing query used to link with this new query. Existing query will be added as an argument of new query.

xsd:string attributeFilter

Used to specify the attribute values based on which the query results can be streamlined.

For example: (assetCn=z*) AND (assetDescription=This is a sample Query).

The above attributeFilter helps to find assets with name starting 'z'.

Other Attributes are as follows:

- assetCn
- assetDescription
- assetName
- c
- company
- dc
- department
- directReports
- displayName
- dnsHostName

- email
- givenName
- groupCn
- groupDescription
- groupMembers
- groupName
- jobTitle
- l
- memberOf
- objectClass
- operatingSystem
- operatingSystemServicePack
- operatingSystemVersion
- postalCode
- streetAddress
- telephoneNumber
- uniqueComputerFields
- uniqueUserFields
- userCn
- userDescription
- userID
- userName
- userPassword
- userSn.

xsd:bool attributeFilterSupplied

Set the value of this parameter to true if 'attributeFilter' is supplied otherwise set it to false.

xsd:bool searchRecursive

Set it true if recursive search is desired.

xsd:bool searchRecursiveSupplied

Set it to true if 'searchRecursive' parameter is set to true otherwise set it to false.

Remarks

The value of 'attributeFilterSupplied' can be set to true or false if value of 'queryFragmentType' is DIRECTORY_***. For all other values 'attributeFilterSupplied' should be set to false.

QueryProperties

This sequence of elements describes the properties of a query.

xsd:string queryId

The UUID identifying the query.

xsd:string queryName

The display name of the query.

xsd:string queryTable

The database table in which the target objects of the query are stored.

QueryType queryType

The type of the query.

xsd:string queryTableLabelField

The database table in which the target objects of the query are stored.

xsd:string queryTableLabelPrimaryKey

The primary key of the database table in which the target objects of the query are stored.

xsd:string queryTableLabelSecondaryKey

The secondary key of the database table in which the target objects of the query are stored.

ArrayOfQueryFragment queryFragments

An array of query statement fragments that are combined to create the complete query definition.

Remarks

None

QueryProperties3

This sequence of elements describes the properties of a query.

xsd:string queryId

The UUID identifying the query.

xsd:string queryName

The display name of the query.

xsd:string queryTable

The database table in which the target objects of the query are stored.

QueryType queryType

The type of the query.

xsd:string queryTableLabelField

The database table in which the target objects of the query are stored.

xsd:string queryTableLabelPrimaryKey

The primary key of the database table in which the target objects of the query are stored.

xsd:string queryTableLabelSecondaryKey

The secondary key of the database table in which the target objects of the query are stored.

ArrayOfQueryFragment2 queryFragments

An array of query statement fragments that are combined to create the complete query definition.

Xsd:string sqlStatement

The sql statement of the query

QueryPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a query's properties.

- `xsd:boolean queryIdRequired`
- `xsd:boolean queryNameRequired`
- `xsd:boolean queryTableRequired`
- `xsd:boolean queryTypeRequired`
- `xsd:boolean queryTableLabelFieldRequired`
- `xsd:boolean queryTableLabelPrimaryKeyRequired`
- `xsd:boolean queryTableLabelSecondaryKeyRequired`
- `xsd:boolean queryFragmentsRequired`

Remarks

For more details, see [QueryProperties](#) (see page 716).

QueryPropertiesRequired3

This sequence of elements specifies which properties are required when retrieving a query's properties.

`xsd:boolean queryIdRequired`

`xsd:boolean queryNameRequired`

`xsd:boolean queryTableRequired`

`xsd:boolean queryTypeRequired`

`xsd:boolean queryTableLabelFieldRequired`

`xsd:boolean queryTableLabelPrimaryKeyRequired`

`xsd:boolean queryTableLabelSecondaryKeyRequired`

`xsd:boolean queryFragmentsRequired`

`xsd:boolean sqlStatementRequired`

ScanPackageInfo

xsd:string productName (m)

Name of the product.

xsd:string packageName (m)

Name of the package which needs to be scanned for.

xsd:string packageVersion

Version of the package.

xsd:bool bPackageVersionSupplied

Flag specifying whether version has been supplied

xsd:string platform

Package platform.

xsd:bool bPlatformSupplied

Flag specifying whether platform has been supplied

xsd:string language

Language of the package that needs to be scanned

xsd:bool bLanguageSupplied

Flag specifying whether language has been supplied

ScanParameters

dsm:ArrayOfstring targetMachines (Optional)

Names or IP's of the target machines to be scanned. This cannot be combined with IP address range (start IP and end IP), domain name, query name and URI options.

xsd:bool bTargetMachinesSupplied

Flag specifying whether the target machines have been supplied.

xsd:string domainName (Optional)

Name of the domain. If the domainName and IP address range (start IP and end IP) are supplied then the IP addresses falling with the specified domain will be scanned. If IP address range is not specified, then scan will happen for all the machines in the domain. This parameter is valid only for WinNT domains.

Domain name cannot be combined with targetMachines, query name and URI options, but can be combined only with IPAddressRange.

xsd:bool bDomainNameSupplied

Flag specifying whether the domain name has been supplied

xsd:string startIPAddress(Optional)

Starting IP address of the IP address range.

xsd:bool bStartIPAddressSupplied

Flag specifying whether the start IP address has been supplied

xsd:string endIPAddress(Optional)

End IP address of Range. Mandatory if the startIPAddress is supplied.

If the domainName and IP range are supplied then scan will happen for the IPs that fall in that domain.

xsd:bool bEndIPAddressSupplied

Flag specifying whether the end IP address has been supplied. The IP address range cannot be combined with either targetMachines, queryName or URI. If IP range is supplied (i.e StartIPAddress and EndIPAddress) then they should be IPv4 addresses only.

xsd:string queryName

Name of the query. This when run should refer to a set of computers. This option cannot be combined with targetMachines, domainName, IP address range and URI.

xsd:bool bQueryNameSupplied

Flag specifying whether the query name has been supplied

xsd:string URI

URI to specify the machines on which the scan is to take place.

xsd:bool bURISupplied

Flag specifying whether the URI has been supplied. This option cannot be combined with targetMachines, domainName, IP address range and queryname.

xsd:bool bDeploymentPackageFromSS (Optional)

This flag should be set to true, if the packages are transferred from a scalability server for eventual deployment.

xsd:bool bDeploymentPackageFromSSSupplied

Flag specifying whether bDeploymentPackageFromSS flag has been supplied.

xsd:bool bScanStagedPackages (Optional)

This flag should be set to true, if packages staged on scalability server needs to be scanned instead of the deployed ones. By default, the scan is done against the deployed packages.

xsd:bool bScanStagedPackagesSupplied

Flag specifying whether bScanStagedPackages flag has been supplied

SDAgentProperties

Members of sequence SDAgentProperties:

xsd:string jobCalendar

Specifies the name of calendar.

xsd:boolean jobCalendarSupplied (m)

True or false depending on the value of jobCalendar.

xsd:boolean softwareManagedSystem

The default value is false

xsd:boolean softwareManagedSystemSupplied (m)

True or false depending on the softwareManaged value.

dsm:RACPolicy racPolicy

Specifies RAC Policy. The default value is COMMON.

xsd:boolean RACPolicySupplied (m)

True or false depending on the value of RACPolicy.

DownloadMethod downloadMethod.

The default value is INTERNAL_NOS.

xsd:boolean downloadMethodSupplied (m)

True or false depending on the value of downloadMethod.

SecurityAreaProperties

xsd:string securityAreaUUID

The uuid of the security Area.

xsd:string securityAreaOwner

The owner of the security Area.

xsd:string securityAreaID

The ID of the security Area

xsd:string securityAreaName

The name of the security Area.

xsd:string securityAreaDescription

The description of the security Area.

SecurityAreaPropertiesRequired

- **xsd:bool securityAreaUUIDRequired**
- **xsd:bool securityAreaOwnerRequired**
- **xsd:bool securityAreaIDRequired**
- **xsd:bool securityAreaNameRequired**
- **xsd:bool securityAreaDescriptionRequired**

SecurityClassPermissions

xsd:string securityProfileUUID

UUID of the security profile.

dsm:SecurityProfilePermission securityProfilePermissionMask

The properties required to create default security permissions.

dsm:SecurityProfileSpecialPermissionFlags

securityProfileSpecialPermissionFlags

The properties required to create special access permission.

User needs to fill this structure if he or she selects 'SPECIALACCESS' from '[SecurityProfilePermission](#)' (see page 90) enum. Default value is 'No Access'.

SecurityClassPermissionsForSecurityProfile

dsm:SecurityClassType securityClassType

The security class type.

dsm:SecurityProfilePermission securityProfilePermissionMask

The properties required to create default security permissions.

dsm:SecurityProfileSpecialPermissionFlags

securityProfileSpecialPermissionFlags

The properties required to create special access permission.

User needs to fill this structure if he or she selects 'SPECIALACCESS' from '[SecurityProfilePermission](#)' (see page 90)' enum. Default value is 'No Access'.

SecurityClassPermissionProperties

dsm:SecurityClassType securityClassType

Security class type.

dsm:ArrayOfSecurityClassPermissions arrayOfSecurityClassPermissions

An array of 'SecurityClassPermissions' – one for each security profile.

SecurityClassProperties

dsm:SecurityClassType securityClassType

Security class type.

xsd:string securityClassUUID

UUID of the security class.

xsd:string securityClassOwner

Owner of the security class.

dsm: ArrayOfSecurityClassPermissions arrSecurityClassPermissions

It is an array of security permissions of the security class with respect to the total number of security profiles.

SecurityClassPropertiesRequired

- xsd:bool securityClassTypeRequired
- xsd:bool securityClassUUIDRequired
- xsd:bool securityClassOwnerRequired
- xsd:bool securityClassPermissionsRequired

SecurityGroupPermissionsForAllSecurityProfiles

xsd:string securityProfileUUID

UUID of the security profile. Security permissions of the security group are retrieved with respect to this security profile.

dsm:SecurityObjectPermissions ObjectPermissions

The security permissions each profile has on the selected group.

dsm:SecurityObjectPermissions membersPermissions

The security permissions each security profile has on the sub-groups and members of the selected group.

SecurityGroupProfilePermission

xsd:string securityProfileUUID

UUID of the security profile. Security permissions for the given security group are retrieved w.r.t this security profile.

dsm:SecurityObjectPermissions objectPermissions

The security permissions each profile has on the selected group.

dsm:SecurityObjectPermissions membersPermissions

The security permissions each security profile has on the sub-groups and members of the selected group.

SecurityObjectLinkageProperties

xsd:string securityAreaName

Security Area name.

dsm:SecurityAreaLinkage securityAreaLinkage

It specifies whether user want to link or unlink given security object from the above security area.

SecurityObjectProfilePermission

xsd:string securityProfileUUID

UUID of the security profile.

dsm:SecurityObjectPermissions securityObjectPermissions

Security permissions of the security object with respect to the above security profile UUID.

SecurityObjectSecurityAreaLinkages

xsd:string securityObjectUUID

The security objects UUID.

**dsm:ArrayOfSecurityObjectLinkageProperties
securityObjectLinkagesPropertiesArray**

An array of SecurityObjectLinkageProperties structure.

It contains security areas to which security objects are to be linked or unlinked.

SecurityObjectOwner

xsd:string securityObjectUUID

UUID of the security object for which owner is to be determined.

xsd:string securityObjectOwner

It contains owner of the security object.

SecurityObjectPermissions

dsm:SecurityProfilePermission securityProfilePermission

The properties required to create default security permissions

**dsm: SecurityProfileSpecialPermissionFlags
securityProfileSpecialPermissionFlags**

The properties required to create special access permission.

User needs to fill this structure if he or she selects 'SPECIALACCESS' from 'SecurityProfilePermission' enum. Default value is 'No Access'.

SecurityAreaByName

xsd:string securityAreaName

SecurityArea Name.

bool securityAreaEnabled

True or false depending upon whether security area is enabled or not.

SecurityAreaConfiguration

xsd:bool securityAreaSupport

True or false depending upon whether security area settings to be enabled or disabled.

dsm:ArrayOfSecurityAreas arrayOfSecurityAreas

Array of SecurityArea and SecurityAreaEnable values.

SecurityAreaConfigurationByName

xsd:bool securityAreaSupport

True or false depending upon whether security area settings to be enabled or disabled.

dsm:ArrayOfSecurityAreasByName arrayOfSecurityAreas

Array of SecurityAreaNames and SecurityAreaEnable values.

SecurityObjectPermissionsForAllSecurityProfiles

xsd:string securityProfileUUID

UUID of the security profile.

dsm:SecurityObjectPermissions securityObjectPermissions

Security permissions of the security object with respect to the above security profile.

SecurityObjectsSecurityAreaProperties

xsd:string securityObjectUUID

UUID of the security object.

dsm:ArrayOfString securityAreasLinked

Array of security areas linked to the object.

SecurityProfileProperties

xsd:string securityProfileUUID

The uuid of the security profile.

xsd:string securityProfileOwner

The owner of the security profile.

xsd:string securityProfileName

The name of the security profile.

xsd:string securityProfileDescription

The description of the security profile.

dsm:SecurityProfileType securityProfileType

The type of the security profile.

xsd:bool builtInProfile

This property indicates whether the security profile is built-in or not. Built-in profiles are created at the installation time.

xsd:bool securityAreaEnabled

This property indicates whether security area support for security profile is enabled or not.

SecurityProfilePropertiesRequired

This sequence of elements specifies which properties are required when retrieving a security profile properties.

- **xsd:bool securityProfileUUIDRequired**
- **xsd:bool securityProfileOwnerRequired**
- **xsd:bool securityProfileNameRequired**
- **xsd:bool securityProfileDescriptionRequired**
- **xsd:bool securityProfileTypeRequired**
- **xsd:bool builtInProfileRequired**
- **xsd:bool securityAreaEnabledRequired**

SetSecurityProfileProperties

xsd:string securityProfileNewName

The new name of the security profile. Name should be provided in URI format. For example, (winnt://<domain name>/<username or groupname>)

xsd:bool securityProfileNameSupplied

True or false depending upon whether new security profile name is provided or not.

xsd:string securityProfileDescription

The description of the security profile.

xsd:bool securityProfileDescriptionSupplied

True or false depending upon whether security profile description is provided or not.

xsd:bool securityAreaEnabled

Pass true if you want this security profile to be security area enabled.

xsd:bool securityAreaEnabledSupplied

True or false depending on whether security area enabled supplied or not.

SetSecurityGroupPermissions

xsd:string securityProfileUUID

UUID of the security profile.

dsm:SecurityObjectPermissions objectPermissions

The security permissions each profile has on the selected group.

xsd:bool objectPermissionsSupplied

True or false depending on whether SecurityObjectPermissions is required or not.

dsm: SecurityObjectPermissions membersPermissions

The security permissions each security profile has on the sub-groups and members of the selected group.

xsd:bool memberPermissionsSupplied

True or false depending on whether SecurityObjectPermissions is required or not.

SetSecurityObjectPermissions

xsd:string securityProfileUUID

UUID of the security profile.

dsm:SecurityObjectPermissions securityObjectPermissions

The security permissions of the security object with respect to above security profile.

SecurityProfileSpecialPermissionFlags

xsd:bool createPermission

Allows user to create objects.

xsd:bool viewPermission

Allows user to view objects.

xsd:bool readPermission

Allows user to read sub-objects of an object.

xsd:bool writePermission

Allows user to change an object.

xsd:bool executePermission

Allows execution, depends on object type.

xsd:bool deletePermission

Allows user to delete object.

xsd:bool changePermission

Allows user to change the ACE itself.

xsd:bool takeOwnershipPermission

Allows user to take ownership of an object.

SetCatalogSoftwarePackageGroupProperties

This sequence of elements specifies the properties required for updating a CatalogSoftwarePackageGroup.

xsd:string softwarePackageGroupId ^(m)

The UUID identifying the catalog software package group. ^(m)

xsd:boolean softwarePackageGroupIdSupplied ^(m)**xsd:string unitGroupId**

The UUID identifying the unit group to which the software packages in this catalog group are to be made available.

xsd:boolean unitGroupIdSupplied**xsd:string name**

The display name of the catalog software package group.

xsd:boolean "nameSupplied" type**xsd:string comment**

A comment for the catalog software package group.

xsd:boolean commentSupplied**Remarks**

None

SetComputerProperties

Members of sequence SetComputerProperties:

xsd:string computerPrimaryNetworkAddress

Denotes the primary network address of the computer.

xsd:boolean computerPrimaryNetworkAddressSupplied (m)

True or false depending on the value of
computerPrimaryNetworkAddress.

xsd:SDAgentProperties sdAgentProperties

Specifies SD agent properties; ignored if SDAgent is not configured.

xsd:boolean sdAgentPropertiesSupplied (m)

True or false depending on the value of sdAgentProperties.

xsd:string user

Specifies user name.

xsd:boolean userSupplied (m)

True or false depending on the value of user.

xsd:string phone

Specifies the user's phone number.

xsd:boolean phoneSupplied (m)

True or false depending on the value of phone.

xsd:string location

Specifies the location of the user.

xsd:boolean locationSupplied (m)

True or false depending on the value of location.

xsd:string comment

Specifies the comment.

xsd:boolean commentSupplied (m)

True or false depending on the value comment.

SetDomainGroupProperties

xsd:string groupLabel

Specifies the domain group label.

xsd:boolean groupLabelSupplied (m)

Specifies if the domain group label is supplied.

xsd:string groupDescription

Specifies the domain group description.

xsd:boolean groupDescriptionSupplied (m)

Specifies if domain group description is supplied.

SetSecurityClassProperties

It contains security profile UUID and security permissions for the security class with respect to the security profile.

xsd:string securityProfileUUID

The security profile UUID.

dsm:SecurityClassPermissions securityClassPermissions

The required security permissions for the array of security classes with respect to above security profile.

SetSoftwareJobContainerProperties2

This sequence of elements specifies which software job container properties are to be set.

- **xsd:string name**
- **xsd:boolean nameSupplied**
- **xsd:string comment**
- **xsd:boolean commentSupplied**
- **xsd:long priority**
Limits: 1 (highest) to 10 (lowest)
- **xsd:long prioritySupplied**
Default: linkage batch and cascade.
- **SoftwareJobContainerPropertyMask**
softwareJobContainerPropertyMask
- **Xsd:boolean softwareJobContainerPropertyMaskSupplied**

SetSoftwarePolicyProperties2

Members of sequence SetSoftwarePolicyProperties2:

xsd:string policyName

The new name of the software policy that needs to be updated

xsd:boolean policyNameSupplied (m)

True or false depending on the value of policyName.

xsd:string comment

New comment string for software policy.

xsd:boolean commentSupplied (m)

True or false depending on the value of comment.

xsd:long priority

The priority of the container, can be 1(highest) to 10 (lowest).

xsd:boolean prioritySupplied

xsd:string unitGroupId

UUID of group that is associated with the policy. You can provide the uuid of an existing unit group or a NULL string. If you provide a NULL string, the software policy if already associated with an unit group will be disassociated from the unit group.

xsd:boolean unitGroupIdSupplied (m)

True or false depending on the value of unitGroupId.

CreateSoftwareJobLinkage createSoftwareJobLinkage

Sequence for setting the software job linkage related properties.

xsd:boolean createSoftwareJobLinkageSupplied (m)

True or false depending on the value of createSoftwareJobLinkage.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

Specifies the type of evaluation that will be done when the policy is applied to the group.

xsd:boolean softwarePolicyEvaluationTypeSupplied (m)

True or false depending on the value of softwarePolicyEvaluationType.

DateTime nextEvaluationDate

Specifies the date and time for next evaluation. The default value is actual time.

xsd:boolean nextEvaluationDateSupplied (m)

True or false depending on the value of nextEvaluationDate.

xsd:long evaluationPeriod

Specifies the period of evaluation in hours.

xsd:boolean evaluationPeriodSupplied

Mandatory parameter. True or false depending on the value of evaluationPeriod.

xsd:boolean enterpriseTime

If set to true, the time specifications are based on the enterprise manager time.

xsd:boolean enterpriseTimeSupplied

Mandatory parameter. True or false depending on the value of enterpriseTimeSupplied.

xsd:string evaluationCalendar

Specifies the name of the calendar that controls the time when the evaluation can take place. To activate a evaluationCalendar user also has to specify "evaluationPeriod".

xsd:boolean evaluationCalendarSupplied (m)

True or false depending on the value of evaluationCalendarSupplied.

AdhocTargetEvaluation adhocTargetEvaluation

Target evaluation is triggered when agents report inventory to the manager. This parameter specifies the type of target evaluation.

xsd:boolean adhocTargetEvaluationSupplied (m)

True or false depending on the value of adhocTargetEvaluation.

xsd:boolean regenerate

This parameter is considered only if evaluationType is SP_SETUPJOBS or SP_SETUPANDEXECUTEJOBS.

xsd:boolean regenerateSupplied (m)

True or false depending on the value of regenerate.

SoftwarePolicyEvaluationPeriodType**softwarePolicyEvaluationPeriodType**

(Optional). Specifies whether the software policy will be evaluated once or periodically. If periodic then the evaluation is done as per the evaluation period set. The default value is once.

xsd:boolean softwarePolicyEvaluationPeriodTypeSupplied (m)

True or false depending on the value for softwarePolicyEvaluationPeriodType.

SetSoftwareGroupProperties

Members of sequence: SetSoftwareGroupProperties:

xsd:string softwareGroupName

Specifies new name for software group.

xsd:boolean softwareGroupNameSupplied (m)

True or false depending on the value of softwareGroupName.

xsd:string comment

Specifies new comment string for software group.

xsd:boolean commentSupplied (m)

True or false depending on the value of comment.

SetQueryProperties

This sequence of elements specifies which query properties are to be set.

- **xsd:string queryId (m)**
- **xsd:boolean queryIdSupplied (m)**
Note: Set to true.
- **xsd:string queryName**
- **xsd:boolean queryNameSupplied**
- **xsd:string queryTable**
- **xsd:boolean queryTableSupplied**
- **xsd:string queryTableLabelField**
- **xsd:boolean queryTableLabelFieldSupplied**
- **xsd:string queryTableLabelPrimaryKey**
- **xsd:boolean queryTableLabelPrimaryKeySupplied**
- **xsd:string queryTableLabelSecondaryKey**
- **xsd:boolean queryTableLabelSecondaryKeySupplied**
- **ArrayOfQueryFragment queryFragments**
- **xsd:boolean queryFragmentsSupplied**

Remarks

For more details, see [QueryProperties](#) (see page 716).

SetQueryProperties3

This sequence of elements specifies which query properties are to be updated.

xsd:string queryName

The name of the query

xsd:boolean queryNameSupplied

Specify true or false depending on whether the queryName parameter is supplied.

xsd:string queryTable

Query table to be updated

xsd:boolean queryTableSupplied

Specify true or false depending on whether the queryTable parameter is supplied.

xsd:string queryTableLabelField

xsd:boolean queryTableLabelFieldSupplied

xsd:string queryTableLabelPrimaryKey

xsd:boolean queryTableLabelPrimaryKeySupplied

xsd:string queryTableLabelSecondaryKey

xsd:boolean queryTableLabelSecondaryKeySupplied

ArrayOfQueryFragment2 queryFragments

xsd:boolean queryFragmentsSupplied

SetSoftwarePackageProcedureGroupProperties

This sequence of elements specifies which software package procedure group properties are to be set.

- **xsd:string softwarePackageProcedureGroupId** (^m)
- **xsd:boolean softwarePackageProcedureGroupIdSupplied** (^m)
- **xsd:string name**
- **xsd:boolean nameSupplied**
- **xsd:string comment**
- **xsd:boolean commentSupplied**

Remarks

For more details, see [SoftwarePackageProcedureGroupProperties](#) (see page 838).

SetSoftwarePackageProcedureProperties3

The software package procedure properties to be updated.

Parameters

xsd:string softwarePackageId

Specifies the UUID that identifies the software package.

xsd:boolean softwarePackageIdSupplied

Specifies true or false depending on whether the softwarePackageId parameter is supplied.

dsm:CommonSoftwarePackageProcedureProperties3

commonSoftwarePackageProcedureProperties

Procedure properties to be updated

SoftwarePackageProcedureProperties3

The software package procedure properties returned from the server

xsd:string softwarePackageProcedureId

Specifies the UUID identifying the software package procedure.

xsd:string softwarePackageProcedureName

Specifies the name of the software package procedure.

SoftwarePackageProcedureTask softwarePackageProcedureTask

Specifies the task type of the software package procedure. If no value is specified, the default value, INSTALL, is used.

SoftwarePackageProcedureType softwarePackageProcedureType

Specifies the type of the software package procedure. If no value is specified, the default value, EXEFILE, is used.

xsd:string comment

Specifies a comment for the software package procedure.

xsd:boolean userSelectable

Specifies whether to allow the procedure to be included in the SD Catalog. If no value is provided, the default value, false, is used.

xsd:boolean externalProcedure

Specifies whether the procedure is external to SD. If no value is provided, the default value, false, is used.

SoftwarePackageProcedureFiledBy filedBy

Specifies the entity by which the software package procedure is filed.

xsd:long operatingSystemType

Specifies the type of the operating system. If no value is provided, the default value, SDAPI_OS_ANY, is used.

xsd:string subPath

Specifies the path of the procedure within the volume.

xsd:string procToRun

Runs the procedure command.

xsd:string parameters

Specifies a string value containing the input parameters for the procedure.

SoftwarePackageProcedureRelation procedureRelation

Specifies the path of the procedure relative to the path of the software package to which the procedure belongs.

SoftwarePackageProcedureFiledBy activatedBy

Specifies the entity by which the software package procedure was activated.

DateTime activatedDateTime

Specifies the time or date when the procedure is activated.

xsd:string softwarePackageId

Specifies the UUID identifying the software package to which the procedure is associated.

xsd:string softwarePackageName

Specifies the name of the software package to which the procedure is associated.

xsd:string softwarePackageVersion

Specifies the version identifier of the software package to which the procedure is associated.

SoftwarePackageType softwarePackageType

Specifies the type of the software package to which the procedure is associated.

xsd:string softwarePackageVolumeld

Specifies the UUID identifying the volume of the software package to which the procedure is associated.

xsd:string softwarePackageVolumeName

Specifies the name of the volume of the software package to which the procedure is associated.

SoftwarePackageProcedureOperationMask3 operationMask

Specifies the operation property for the procedure operation.

SoftwarePackageProcedureBootMask2 bootBeforeMask

Specifies the restart options requested for the software package procedure before executing the procedure.

SoftwarePackageProcedureBootMask2 bootAfterMask

Specifies the restart options requested for the software package procedure after executing the procedure.

xsd:string dependancyQueryId

Specifies UUID of the query that checks whether the condition is fulfilled, provided a prerequisite is defined for the procedure.

**SoftwarePackageProcedureManagementTypeMask
managementTypeMask**

Specifies whether to allow the requested software package for the procedure to run on the Software Managed software targets. If no value is provided, the default value, SD, is used.

SetSoftwareJobContainerProperties

This sequence of elements specifies which software job container properties are to be set.

- xsd:string softwareJobContainerId (^m)
- xsd:boolean softwareJobContainerIdSupplied (^m)
- xsd:string name
- xsd:boolean nameSupplied
- xsd:string comment
- xsd:boolean commentSupplied
- SoftwareJobContainerPropertyMask
softwareJobContainerPropertyMask
- xsd:boolean softwareJobContainerPropertyMaskSupplied

Remarks

For more details, see [SoftwareJobContainerProperties](#) (see page 786).

SetSoftwareJobContainerProperties2

This sequence of elements specifies which software job container properties are to be set.

- xsd:string name
- xsd:boolean nameSupplied
- xsd:string comment
- xsd:boolean commentSupplied
- xsd:long priority

Limits: 1 (highest) to 10 (lowest)
- xsd:long prioritySupplied

Default: linkage batch and cascade.
- SoftwareJobContainerPropertyMask
softwareJobContainerPropertyMask
- Xsd:boolean softwareJobContainerPropertyMaskSupplied

SetSoftwareJobContainerProperties3

The SetSoftwareJobContainerProperties3 sequence of elements specifies which software job container properties are to be set.

xsd:string name

Specifies the name of the software job container.

xsd:boolean nameSupplied

Specifies true or false depending on whether the name parameter is supplied.

xsd:string comment

Specifies the comment for the software job container.

xsd:boolean commentSupplied

Specifies true or false depending on whether the comment parameter is supplied.

xsd:long priority

Specifies the priority of the software job container. The range of the priority is from 1 (highest) to 10 (lowest). If no value is provided, the default value, linkageBatch and Cascade, is used.

xsd:long prioritySupplied

Specifies true or false depending on whether the priority parameter is supplied.

SoftwareJobContainerPropertyMask**softwareJobContainerPropertyMask**

Specifies which job container properties are to be initialized when a software job container is created.

xsd:boolean softwareJobContainerPropertyMaskSupplied

Specifies true or false depending on whether the softwareJobContainerPropertyMask parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

Xsd:int userMessageSupplied

Checks whether or not the custom administrator message is provided.

Remarks

None

SoftwareJobContainerPropertiesRequired2

This sequence of elements specifies which properties are required when retrieving a software job container's properties.

- xsd:boolean softwareJobContainerIdRequired
- xsd:boolean stateRequired
- xsd:boolean sealRequired
- xsd:boolean noOfParentSoftwareJobContainersRequired
- xsd:boolean noOfChildSoftwareJobContainersRequired
- xsd:boolean noOfSoftwareJobsRequired
- xsd:boolean nameRequired
- xsd:boolean commentRequired
- xsd:boolean priorityRequired
- xsd:boolean softwareJobContainerPropertyMaskRequired
- xsd:boolean creationDateRequired
- xsd:boolean changeDateRequired
- xsd:boolean attribMaskRequired
- xsd:boolean waitCountRequired
- xsd:boolean activeCountRequired
- xsd:boolean errorCountRequired
- xsd:boolean okCountRequired
- xsd:boolean renewStateRequired

SetSoftwareJobProperties

This sequence of elements specifies which software job properties are to be set.

- `xsd:string softwareJobId` (^m)
- `xsd:boolean softwareJobIdSupplied` (^m)
- `xsd:string softwareJobName`
- `xsd:boolean softwareJobNameSupplied`
- `DateTime executionDate`
- `xsd:boolean executionDateSupplied`
- `xsd:long timeoutHours`
- `xsd:boolean timeoutHoursSupplied`
- `SoftwareJobStartupTime startWhen`
- `xsd:boolean startWhenSupplied`
- `SoftwareJobBootMask bootBeforeMask`
- `xsd:boolean bootBeforeMaskSupplied`
- `SoftwareJobBootMask bootAfterMask`
- `xsd:boolean bootAfterMaskSupplied`
- `SoftwareJobOperationMask operationMask`
- `xsd:boolean operationMaskSupplied`
- `xsd:string userParameters`
- `xsd:boolean userParametersSupplied`
- `xsd:long promptTimeoutHours`
- `xsd:boolean promptTimeoutHoursSupplied`
- `xsd:string deliveryCalendar`
- `xsd:boolean deliveryCalendarSupplied`
- `DateTime deliveryDate`
- `xsd:boolean deliveryDateSupplied`

Remarks

For more details, see [SoftwareJobProperties](#) (see page 768).

SetSoftwareJobProperties2

This sequence of elements specifies which software job properties are to be set.

- `xsd:string softwareJobName`
- `xsd:boolean softwareJobNameSupplied`
- `DateTime executionDate`
- `xsd:boolean executionDateSupplied`
- `xsd:long timeoutHours`
- `xsd:boolean timeoutHoursSupplied`
- `SoftwareJobStartupTime startWhen`
- `xsd:boolean startWhenSupplied`
- `SoftwareJobBootMask bootBeforeMask`
- `xsd:boolean bootBeforeMaskSupplied`
- `SoftwareJobBootMask bootAfterMask`
- `xsd:boolean bootAfterMaskSupplied`
- `SoftwareJobOperationMask2 operationMask2`
- `xsd:boolean operationMaskSupplied`
- `xsd:string userParameters`
- `xsd:boolean userParametersSupplied`
- `xsd:long promptTimeoutHours`
- `xsd:boolean promptTimeoutHoursSupplied`
- `xsd:string deliveryCalendar`
- `xsd:boolean deliveryCalendarSupplied`
- `DateTime deliveryDate`
- `xsd:boolean deliveryDateSupplied`

SetSoftwareJobProperties3

The SetSoftwareJobProperties3 sequence of elements specifies which software job properties are to be set.

xsd:string softwareJobName

Specifies the display name of the software job.

xsd:boolean softwareJobNameSupplied

Specifies true or false depending on whether the softwareJobName parameter is supplied.

DateTime executionDate

Specifies the Date and Time when to execute the job.

xsd:boolean executionDateSupplied

Specifies true or false depending on whether the executionDate parameter is supplied.

xsd:long timeoutHours

Specifies the time in hours after which, if the job has been unable to complete, the job will cease attempts to complete. If you do not provide any value, the default value, 48, is used.

xsd:boolean timeoutHoursSupplied

Specifies true or false depending on whether the timeoutHours parameter is supplied.

SoftwareJobStartupTime startWhen

Specifies the time when to start the software job. If you do not provide any value, the default value, EXACT_LOCAL_TIME, is used.

xsd:boolean startWhenSupplied

Specifies true or false depending on whether the startWhen parameter is supplied.

SoftwareJobBootMask3 bootBeforeMask

Specifies the restart options requested for the software job before executing the job. If you do not specify any value, the default value, noReboot, is used.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwareJobBootMask3 bootAfterMask

Specifies the restart options requested for the software job after executing the job. If you do not specify any value, the default value noReboot is used.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the `bootAfterMask` parameter is supplied.

SoftwareJobOperationMask2 operationMask2

Specifies the properties for the operation of the software job. If you do not specify any value, the default value `calendarControl` is used

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the `operationMask2` parameter is supplied.

xsd:string userParameters

Specifies the user-defined parameters

xsd:boolean userParametersSupplied

Specifies true or false depending on whether the `userParameters` parameter is supplied.

xsd:long promptTimeoutHours

Specifies the number of hours to display a user prompt before forced execution of the job. If no value is provided, the default value, 24, is used.

xsd:boolean promptTimeoutHoursSupplied

Specifies true or false depending on whether the `promptTimeoutHours` parameter is supplied.

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to use for scheduling the job. The calendar controls the days, dates, and times of the scheduled delivery. The time cells that are not set in the calendar define the exclusion intervals. If the name of the calendar is unknown to SD, the job is not created

xsd:boolean deliveryCalendarSupplied

Specifies true or false depending on whether the `deliveryCalendar` parameter is supplied.

DateTime deliveryDate

Specifies the delivery date and time. If not supplied, the default value is the current date and time of the manager computer.

xsd:boolean deliveryDateSupplied

Specifies true or false depending on whether the `deliveryDate` parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

Structure CreateUnitGroupProperties2 CreateUnitGroupProperties2

Members of sequence CreateUnitGroupProperties2:

xsd:string groupLabel (m)

Specifies the name of the group.

xsd:string groupDescription

Specifies the description of the group.

xsd:boolean groupDescriptionSupplied (m)

True or false depending on the groupDescription value.

xsd:string queryUUID

Specifies the uuid of the query to be linked to the newly created group. If not specified, no query is linked to the group.

xsd:boolean queryUUIDSUPPLIED (m)

True or false depending on the value of queryUUID.

xsd:string engineName

Specifies the name of the query evaluation engine to be associated with the unit group.

xsd:boolean engineNameSupplied (m)

True or false depending on the engineName value.

xsd:long evaluationPeriod

Period in which the query is evaluated, specified in minutes. The default value is 1440 minutes. The maximum value is 10080

xsd:boolean evaluationPeriodSupplied (m)

True or false depending on the value of evaluationPeriod.

CreateSoftwareJobLinkage

Members of sequence CreateSoftwareJobLinkage

SoftwareJobLinkageType softwareJobLinkageType

Enumerator which specifies the type of software job linkage. The default value is NOLINKAGE

xsd:boolean softwareJobLinkageTypeSupplied (m)

True or false depending on the value of softwareJobLinkageType.

xsd:boolean rollback

This parameter will be considered only if softwareLinkageType is TRANSACTION. The default value is true.

xsd:boolean rollBackSupplied (m)

True or false depending on the value of rollback.

xsd:boolean cascade

This parameter will be considered only if softwareLinkageType is TRANSACTION or SYNCHRONIZED. The default value is false.

xsd:boolean cascadeSupplied (m)

True or false depending on the value of cascade.

DomainPropertyFilter

DomainProperty domainProperty

The identifier for the domain property on which the search is based.

xsd:string searchString

If the property to search on is a string property, this parameter specifies the value of the search filter to be applied.

This field is mandatory if domainProperty is set to one of the following values:

- DP_DOMAIN_UUID,
- DP_PARENT_DOMAIN_UUID
- DP_DOMAIN_LABEL
- DP_DOMAIN_DESCR
- DP_CREATION_USER
- DP_LAST_UPDATED_USER
- DP_CONTACT_INFO
- DP_DIS_HW_UUID
- DP_DB_HOST_NAME
- DP_DB_SERVER
- DP_DB_TYPE
- DP_DB_NAME
- DP_DB_USER_NAME
- DP_DB_INSTANCE

xsd:long searchValue

If the property to search on is a numeric property, this parameter specifies the value of the search filter to be applied. This field is mandatory if domainProperty is set to one of the following values:

- DP_USAGE_LIST_MASK
- DP_DOMAIN_NUMBER
- DP_LAST_UPDATED_DATE
- DP_DOMAIN_TYPE
- DP_CREATION_DATE

DomainPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a domain's properties.

xsd:boolean domainUUIDRequired (m)

True or false depending on whether the domain UUID needs to be fetched.

xsd:boolean domainTypeRequired (m)

True or false depending on whether the domain UUID needs to be fetched.

xsd:boolean parentDomainUUIDRequired (m)

True or false depending on whether the domain type needs to be fetched.

xsd:boolean domainLabelRequired (m)

True or false depending on whether the domain label needs to be fetched.

xsd:boolean domainDescriptionRequired (m)

True or false depending on whether the domain description needs to be fetched.

xsd:boolean creationUserRequired (m)

True or false depending on whether the creation user needs to be fetched.

xsd:boolean creationDateRequired (m)

True or false depending on whether the domain creation date needs to be fetched.

xsd:boolean lastUpdatedUserRequired (m)

True or false depending on whether the domain last updated user needs to be fetched.

xsd:boolean lastUpdatedDateRequired (m)

True or false depending on whether the domain last updated date needs to be fetched.

xsd:boolean contactInformationRequired (m)

True or false depending on whether the domain contact information needs to be fetched.

xsd:boolean disHWUUIDRequired (m)

True or false depending on whether the domain discovered Hardware UUID needs to be fetched.

xsd:boolean dbHostNameRequired (m)

True or false depending on whether the domain database host name needs to be fetched.

xsd:boolean dbServerRequired (m)

True or false depending on whether the domain database server needs to be fetched.

xsd:boolean dbTypeRequired (m)

True or false depending on whether the domain database type needs to be fetched.

xsd:boolean dbNameRequired (m)

True or false depending on whether the domain database name needs to be fetched.

xsd:boolean dbUserNameRequired (m)

True or false depending on whether the domain database user name needs to be fetched.

xsd:boolean dbInstanceRequired (m)

True or false depending on whether the domain database instance needs to be fetched.

xsd:boolean domainNumberRequired (m)

True or false depending on whether the domain ID needs to be fetched.

xsd:boolean usageListRequired (m)

True or false depending on whether the domain usage list needs to be fetched.

ElementTypeMask

This sequence of elements specifies which types of element are to be returned in a request for elements.

xsd:boolean file

This flag is to be set to true if elements of type 'file' are required.

xsd:boolean dir

This flag is to be set to true if elements of type 'directory' are required.

Remarks

None

EvaluationTypeMask

This sequence of elements defines some of the properties that govern the evaluation of a software policy.

xsd:boolean once

Do the evaluation once. Cannot be combined with periodic.

xsd:boolean periodic

Do the evaluation periodically. Cannot be combined with once.

xsd:boolean enterpriseTime

The evaluation start time is the Enterprise Manager time

xsd:boolean adhocCreate

Do evaluation when new or re-installed computers register

xsd:boolean adhocUpdate

Do evaluation when computers report inventory

Remarks

None

SecurityArea

xsd:string securityAreaUUID

SecurityArea UUID

bool securityAreaEnabled

True or false depending upon whether SecurityArea is to be enabled or not.

SetDistributionContainerProperties

- `xsd:long priority`
Default: 5
Limits: 1 to 10
- `xsd:boolean prioritySupplied`
- `xsd:string ahdInfo`
- `xsd:boolean ahdInfoSupplied`

SetSecurityAreaProperties

`xsd:string securityAreaName`

The name of the security Area.

`xsd:bool securityAreaNameSupplied`

True or False whether area name is supplied or not.

`xsd:string securityAreaDescription`

The description of the security Area.

`xsd:bool securityAreaDescriptionSupplied`

True or False whether area description is supplied or not.

SetSoftwarePackageGroupProperties

This sequence of elements specifies which software package group properties are to be set.

- `xsd:string softwarePackageGroupId` (^m)
- `xsd:boolean softwarePackageGroupIdSupplied` (^m)
- `xsd:string name`
- `xsd:boolean nameSupplied`
- `xsd:string comment`
- `xsd:boolean commentSupplied`

Remarks

For more details, see [SoftwarePackageGroupProperties](#) (see page 828).

SetSoftwarePackageProcedureProperties

This sequence of elements specifies which software package procedure properties are to be set.

- `xsd:string softwarePackageProcedureId` (^m)
- `xsd:boolean softwarePackageProcedureIdSupplied` (^m)
- `xsd:string softwarePackageProcedureName`
- `xsd:boolean softwarePackageProcedureNameSupplied`
- `SoftwarePackageProcedureTask softwarePackageProcedureTask`
- `xsd:boolean softwarePackageProcedureTaskSupplied`
- `SoftwarePackageProcedureType softwarePackageProcedureType`
- `xsd:boolean softwarePackageProcedureTypeSupplied`
- `xsd:string comment`
- `xsd:boolean commentSupplied`
- `xsd:boolean userSelectable`
- `xsd:boolean userSelectableSupplied`
- `xsd:boolean externalProcedure`
- `xsd:boolean externalProcedureSupplied`
- `xsd:long operatingSystemType`
- `xsd:boolean operatingSystemTypeSupplied`
- `xsd:string subPath`
- `xsd:boolean subPathSupplied`
- `xsd:string procToRun`
- `xsd:boolean procToRunSupplied`
- `xsd:string parameters`
- `xsd:boolean parametersSupplied`
- `xsd:string softwarePackageVolumeld`
- `xsd:boolean softwarePackageVolumeldSupplied`
- `SoftwarePackageProcedureOperationMask operationMask`
- `xsd:boolean operationMaskSupplied`
- `SoftwarePackageProcedureBootMask bootBeforeMask`
- `xsd:boolean bootBeforeMaskSupplied`
- `SoftwarePackageProcedureBootMask bootAfterMask`
- `xsd:boolean bootAfterMaskSupplied`
- `xsd:string dependancyQueryId`
- `xsd:boolean dependancyQueryIdSupplied`

- SoftwarePackageProcedureManagementTypeMask
managementTypeMask
- xsd:boolean managementTypeMaskSupplied

Remarks

For more details, see [SoftwarePackageProcedureProperties](#) (see page 843).

SetSoftwarePackageProcedureProperties2

The SetSoftwarePackageProcedureProperties2 sequence of elements specifies which software package procedure properties that you can set.

xsd:string softwarePackageProcedureId

Specifies the UUID that identifies the software package procedure.

xsd:boolean softwarePackageProcedureIdSupplied

Specifies true or false depending on whether the softwarePackageProcedureId parameter is supplied.

xsd:string softwarePackageName

Specifies the name of the software package procedure.

xsd:boolean softwarePackageNameSupplied

Specifies true or false depending on whether the softwarePackageName parameter is supplied.

SoftwarePackageProcedureTask softwarePackageProcedureTask

Specifies the type of the task of the software package procedure. If no value is provided, the default value INSTALL is used.

xsd:boolean softwarePackageProcedureTaskSupplied

Specifies true or false depending on whether the softwarePackageProcedureTask parameter is supplied.

SoftwarePackageProcedureType softwarePackageProcedureType

Specifies the type of the software package procedure. If no value is provided, the default value EXEFILE is used.

xsd:boolean softwarePackageProcedureTypeSupplied

Specifies true or false depending on whether the softwarePackageProcedureType parameter is supplied.

xsd:string comment

Specifies a comment for the software package procedure.

xsd:boolean commentSupplied

Specifies true or false depending on whether the comment parameter is supplied.

xsd:boolean userSelectable

Specifies whether the procedure is allowed to be included in the SD Catalog. If no value is provided, the default value false is used

xsd:boolean userSelectableSupplied

Specifies true or false depending on whether the userSelectable parameter is supplied.

xsd:boolean externalProcedure

Specifies whether the procedure is external to SD. If no value is provided, the default value false is used.

xsd:boolean externalProcedureSupplied

Specifies true or false depending on whether the externalProcedure parameter is supplied.

xsd:long operatingSystemType

Specifies the type of the operating system. If no value is provided, the default value SDAPI_OS_ANY is used.

xsd:boolean operatingSystemTypeSupplied

Specifies true or false depending on whether the operatingSystemType parameter is supplied.

xsd:string subPath

Specifies the path of the procedure within the volume.

xsd:boolean subPathSupplied

Specifies true or false depending on whether the subPath parameter is supplied.

xsd:string procToRun

Runs the procedure command.

xsd:boolean procToRunSupplied

Specifies true or false depending on whether the procToRun parameter is supplied.

xsd:string parameters

Specifies a string value that contains the input parameters for the procedure.

xsd:boolean parametersSupplied

Specifies true or false depending on whether the parameters parameter is supplied.

xsd:string softwarePackageVolumeId

Specifies the UUID that identifies the volume of the software package to which the procedure is associated.

xsd:boolean softwarePackageVolumeIdSupplied

Specifies true or false depending on whether the softwarePackageVolumeId parameter is supplied.

SoftwarePackageProcedureOperationMask2 operationMask

Specifies the operation property for the procedure operation.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask parameter is supplied.

SoftwarePackageProcedureBootMask2 bootBeforeMask

Specifies the options requested for the software package procedure before executing the procedure. If no value is provided, the default value noReboot is used.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwarePackageProcedureBootMask2 bootAfterMask

Specifies the options requested for the software package procedure after executing the procedure. If no value is provided, the default value noReboot is used.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

xsd:string dependancyQueryId

Specifies the UUID of the query that checks whether the condition is fulfilled, provided a prerequisite is defined for the procedure.

xsd:boolean dependancyQueryIdSupplied

Specifies true or false depending on whether the dependancyQueryId parameter is supplied.

**SoftwarePackageProcedureManagementTypeMask
managementTypeMask**

Specifies whether to allow the requested software package for the procedure to run on the *Software Managed* software targets. If no value is provided, the default value SD is used.

xsd:boolean managementTypeMaskSupplied

Specifies true or false depending on whether the managementTypeMask parameter is supplied.

Remarks

For more details, see [SoftwarePackageProcedureProperties](#) (see page 843).

SetSoftwarePackageProperties

This sequence of elements specifies which software package properties are to be set.

- `xsd:string softwarePackageId` (^m)
- `xsd:boolean softwarePackageIdSupplied` (^m)
- `xsd:string softwarePackageName`
- `xsd:boolean softwarePackageNameSupplied`
- `xsd:string softwarePackageVersion`
- `xsd:boolean softwarePackageVersionSupplied`
- `xsd:string supplier`
- `xsd:boolean supplierSupplied`
- `xsd:string comment`
- `xsd:boolean commentSupplied`
- `xsd:string archiveComment`
- `xsd:boolean archiveCommentSupplied`
- `SoftwarePackageMask softwarePackageMask`
- `xsd:boolean softwarePackageMaskSupplied`

Remarks

For more details, see [SoftwarePackageProperties](#) (see page 847).

SetSoftwarePackageVolumeProperties

This sequence of elements specifies which software package volume properties are to be set.

- `xsd:string volumeld` (^m)
- `xsd:boolean volumeldSupplied` (^m)
- `xsd:string volumeName` (^m)
- `xsd:boolean volumeNameSupplied` (^m)
- `xsd:long volumeNumber`
- `xsd:boolean volumeNumberSupplied`
- `xsd:string softwarePackageId`
- `xsd:boolean softwarePackageIdSupplied`

Remarks

For more details, see [SoftwarePackageVolumeProperties](#) (see page 836).

SetSoftwarePolicyProperties

This sequence of elements specifies which software policy properties are to be set.

- xsd:string softwarePolicyId (^m)
- xsd:boolean softwarePolicyIdSupplied (^m)
- xsd:string unitGroupId
- xsd:boolean unitGroupIdSupplied
- xsd:string name
- xsd:boolean nameSupplied
- xsd:string comment
- xsd:boolean commentSupplied
- SoftwarePolicyEvaluationType softwarePolicyEvaluationType
- xsd:boolean SoftwarePolicyEvaluationTypeSupplied
- SoftwareJobContainerPropertyMask softwareJobContainerPropertyMask
- xsd:boolean softwareJobContainerPropertyMaskSupplied
- xsd:string evaluationCalendar
- xsd:boolean evaluationCalendarSupplied
- xsd:long evaluationPeriod
- xsd:boolean evaluationPeriodSupplied
- DateTime nextEvaluationDate
- xsd:boolean nextEvaluationDateSupplied
- EvaluationTypeMask evaluationTypeMask
- xsd:boolean evaluationTypeMaskSupplied

Remarks

For more details, see [SoftwarePolicyProperties](#) (see page 851).

SetSoftwarePolicyProperties3

The SetSoftwarePolicyProperties3 sequence of elements specifies the software policy properties that you need to set.

xsd:string policyName

Specifies the new name of the software policy that you must update.

xsd:boolean policyNameSupplied (m)

Specifies true or false depending on whether the policyName parameter is supplied.

xsd:string comment

Specifies a new comment string for the software policy.

xsd:boolean commentSupplied (m)

Specifies true or false depending on whether the comment parameter is supplied.

xsd:long priority

Specifies the priority of the container that can range from 1 (highest) to 10 (lowest).

xsd:boolean prioritySupplied

Specifies true or false depending on whether the priority parameter is supplied.

xsd:string unitGroupId

Specifies the UUID of the group that is associated with the policy. You can provide the UUID of an existing unit group or a NULL string. If you provide a NULL string, the software policy, if already associated with a unit group, is disassociated from the unit group.

xsd:boolean unitGroupIdSupplied (m)

Specifies true or false depending on whether the unitGroupId parameter is supplied.

CreateSoftwareJobLinkage createSoftwareJobLinkage

Sets the software job linkage related properties.

xsd:boolean createSoftwareJobLinkageSupplied (m)

Specifies true or false depending on whether the createSoftwareJobLinkage parameter is supplied.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

Specifies the type of evaluation that is done when the policy is applied to the group.

xsd:boolean softwarePolicyEvaluationTypeSupplied (m)

Specifies true or false depending on whether the softwarePolicyEvaluationType parameter is supplied.

DateTime nextEvaluationDate

Specifies the date and time for the next evaluation. The default value is actual time.

xsd:boolean nextEvaluationDateSupplied (m)

Specifies true or false depending on whether the nextEvaluationDate parameter is supplied.

xsd:long evaluationPeriod

Specifies the period of evaluation in hours.

xsd:boolean evaluationPeriodSupplied (m)

Specifies true or false depending on whether the evaluationPeriod parameter is supplied.

xsd:boolean enterpriseTime

(If set to true) Specifies that the time specifications are based on the enterprise manager time.

xsd:boolean enterpriseTimeSupplied (m)

Specifies true or false depending on whether the enterpriseTime parameter is supplied.

xsd:string evaluationCalendar

Specifies the name of the calendar that controls the time when the evaluation can take place. To activate evaluationCalendar, you also need to specify evaluationPeriod.

xsd:boolean evaluationCalendarSupplied (m)

Specifies true or false depending on whether the evaluationCalendar parameter is supplied.

AdhocTargetEvaluation adhocTargetEvaluation

Specifies the type of target evaluation. The target evaluation is triggered when agents report inventory to the manager.

xsd:boolean adhocTargetEvaluationSupplied (m)

Specifies true or false depending on whether the adhocTargetEvaluation parameter is supplied.

xsd:boolean regenerate

Regenerates the evaluation type. This parameter is considered only if evaluationType is SP_SETUPJOBS or SP_SETUPANDEXECUTEJOBS.

xsd:boolean regenerateSupplied (m)

Specifies true or false depending on whether the regenerate parameter is supplied.

**SoftwarePolicyEvaluationPeriodType
softwarePolicyEvaluationPeriodType**

(Optional) Specifies whether to evaluate the software policy once or periodically. If the policy is evaluated periodically, the evaluation is done based on the evaluation period set. The default value is once.

xsd:boolean softwarePolicyEvaluationPeriodTypeSupplied (m)

Specifies true or false depending on whether the softwarePolicyEvaluationPeriodType parameter is supplied.

xsd:boolean evaluateDynamicGroups

Specifies that the dynamic groups are evaluated during evaluation of the software policy if the value is set to true. The default value is false.

xsd:boolean evaluateDynamicGroupsSupplied

Specifies true or false depending on whether the evaluateDynamicGroups parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

Xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

Remarks

None

SetSoftwarePolicyPropertiesByName

Members of sequence SetSoftwarePolicyPropertiesByName:

xsd:string policyName

Specifies the new name for software Policy.

xsd:boolean policyNameSupplied (m)

True or false depending on the value of policyName.

xsd:string comment

Comment string for software policy.

xsd:boolean commentSupplied (m)

True or false depending on the value of comment.

xsd:long priority

The priority of the container, can be 1(highest) to 10 (lowest).

xsd:boolean prioritySupplied**xsd:string unitGroupName**

Name of group that is associated with the policy. The user can either provide the name of an existing unit group or a NULL string. If you provide a NULL string, the software policy, if already associated with an unit group will be disassociated from the unit group.

xsd:boolean unitGroupNameSupplied (m)

True or false depending on the value of unitGroupName.

CreateSoftwareJobLinkage createSoftwareJobLinkage

The sequence *CreateSoftwareJobLinkage* is provided for setting the software job linkage related properties.

xsd:boolean createSoftwareJobLinkageSupplied (m)

True or false depending on the value of createSoftwareJobLinkage.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

Specifies the type of evaluation that will done when the policy is applied to the group.

xsd:boolean softwarePolicyEvaluationTypeSupplied

Mandatory parameter. True or false depending on the value of softwarePolicyEvaluationType.

DateTime nextEvaluationDate

Specifies the date and time for next evaluation.

xsd:boolean nextEvaluationDateSupplied (m)

True or false depending on the value of nextEvaluationDate.

xsd:long evaluationPeriod

Specifies the period of evaluation, in hours.

xsd:boolean softwarePolicyEvaluationPeriodTypeSupplied (m)

True or false depending on the value for softwarePolicyEvaluationPeriodType.

xsd:boolean evaluationPeriodSupplied (m)

True or false depending on the value of evaluationPeriod.

xsd:boolean enterpriseTime

If set to true, the time specifications are based on the enterprise manager time.

xsd:boolean enterpriseTimeSupplied (m)

True or false depending on the value of enterpriseTimeSupplied.

xsd:string evaluationCalendar

Specifies the name of the calendar that controls the time when the evaluation can take place. To activate a evaluationCalendar user also has to specify "evaluationPeriod".

xsd:boolean evaluationCalendarSupplied (m)

True or false depending on the value of evaluationCalendarSupplied.

AdhocTargetEvaluation adhocTargetEvaluation

Target evaluation is triggered when agents report inventory to the manager. This parameter specifies the type of target evaluation.

xsd:boolean adhocTargetEvaluationSupplied (m)

True or false depending on the value of adhocTargetEvaluation.

xsd:boolean regenerate

This parameter is considered only if evaluationType is SP_SETUPJOBS or SP_SETUPANDEXECUTEJOBS.

xsd:boolean regenerateSupplied (m)

True or false depending on the value of regenerate.

SoftwarePolicyEvaluationPeriodType**softwarePolicyEvaluationPeriodType**

Optional parameter. Specifies whether the software policy will be evaluated once or periodically. If periodic then the evaluation is done as per the evaluation period set. The default value is once.

xsd:boolean evaluateDynamicGroups

Optional parameter. If set to true, dynamic groups are evaluated during evaluation of the software policy. The

default value is false.

xsd:boolean evaluateDynamicGroupsSupplied (m)

True or false depending on the value of evaluateDynamicGroups.

SetSoftwarePolicyPropertiesByName3

TheSetSoftwarePolicyPropertiesByName3 sequence of elements specifies the software policy properties by name.

xsd:string policyName

Specifies the new name for software Policy.

xsd:boolean policyNameSupplied (m)

Specifies true or false depending on whether the policyName parameter is supplied.

xsd:string comment

Specifies the comment string for the software policy.

xsd:boolean commentSupplied (m)

Specifies true or false depending on whether the comment parameter is supplied.

xsd:long priority

Specifies the priority of the container that can range from 1 (highest) to 10 (lowest).

xsd:boolean prioritySupplied

Specifies true or false depending on whether the priority parameter is supplied.

xsd:string unitGroupName

Specifies the name of the group that is associated with the policy. You can either provide the name of an existing unit group or a NULL string. If you provide a NULL string, the software policy, if already associated with an unit group, is disassociated from the unit group.

xsd:boolean unitGroupNameSupplied (m)

Specifies true or false depending on whether the unitGroupName parameter is supplied.

CreateSoftwareJobLinkage createSoftwareJobLinkage

Specifies that the sequence *CreateSoftwareJobLinkage* is provided for setting the software job linkage related properties.

xsd:boolean createSoftwareJobLinkageSupplied (m)

Specifies true or false depending on whether the createSoftwareJobLinkage parameter is supplied.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

Specifies the type of evaluation that is performed when the policy is applied to the group.

xsd:boolean softwarePolicyEvaluationTypeSupplied (m)

Specifies true or false depending on whether the softwarePolicyEvaluationType parameter is supplied.

DateTime nextEvaluationDate

Specifies the date and time for the next evaluation.

xsd:boolean nextEvaluationDateSupplied (m)

Specifies true or false depending on whether the nextEvaluationDate parameter is supplied.

xsd:long evaluationPeriod

Specifies the period of evaluation in hours.

xsd:boolean softwarePolicyEvaluationPeriodTypeSupplied (m)

Specifies true or false depending on whether the softwarePolicyEvaluationPeriodType parameter is supplied.

xsd:boolean evaluationPeriodSupplied (m)

Specifies true or false depending on whether the evaluationPeriod parameter is supplied.

xsd:boolean enterpriseTime

(If set to true) Specifies that the time specifications are based on the enterprise manager time.

xsd:boolean enterpriseTimeSupplied (m)

Specifies true or false depending on whether the enterpriseTime parameter is supplied.

xsd:string evaluationCalendar

Specifies the name of the calendar that controls the time when the evaluation can take place. To activate evaluationCalendar, you also need to specify evaluationPeriod.

xsd:boolean evaluationCalendarSupplied (m)

Specifies true or false depending on whether the evaluationCalendar parameter is supplied.

AdhocTargetEvaluation adhocTargetEvaluation

Specifies the type of target evaluation. The target evaluation is triggered when agents report inventory to the manager.

xsd:boolean adhocTargetEvaluationSupplied (m)

Specifies true or false depending on whether the adhocTargetEvaluation parameter is supplied.

xsd:boolean regenerate

Regenerates the type of the evaluation. This parameter is considered only if evaluationType is SP_SETUPJOBS or SP_SETUPANDEXECUTEJOBS.

xsd:boolean regenerateSupplied (m)

Specifies true or false depending on whether the regenerate parameter is supplied.

SoftwarePolicyEvaluationPeriodType**softwarePolicyEvaluationPeriodType**

(Optional) Specifies whether to evaluate the software policy once or periodically. If the policy is evaluated periodically, the evaluation is done based on the evaluation period set. The default value is once.

xsd:boolean evaluateDynamicGroups

(Optional) Specifies that the dynamic groups are evaluated during evaluation of the software policy if the value is set to true. The default value is false.

xsd:boolean evaluateDynamicGroupsSupplied (m)

Specifies true or false depending on whether the evaluateDynamicGroups parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

SetTimeoutProperties

This sequence has an element that indicates the web service timeout.

xsd:long setTimeoutValue (m)

The number of seconds after which a web service session will timeout if unused. The range is 60 to 7200 seconds

Remarks

None

SetUnitGroupProperties

This sequence of elements specifies which unit group properties are to be set. `xsd:string groupUUID`

- `xsd:boolean groupUUIDSUPPLIED (m)`
- `xsd:string groupLabel (m)`
- `xsd:boolean groupLabelSupplied`
- `xsd:long evaluationFrequency`
- `xsd:boolean evaluationFrequencySupplied`
- `DateTime lastEvaluationDateTime`
- `xsd:boolean lastEvaluationDateTimeSupplied`
- `xsd:string queryUUID`
- `xsd:boolean queryUUIDSUPPLIED`
- `xsd:string groupDescription`
- `xsd:boolean groupDescriptionSupplied`

Remarks

For more details, see [UnitGroupProperties](#) (see page 870).

SetUserProfileProperties

Members of sequence `SetUserProfileProperties`:

`xsd:SDAgentProperties sdAgentProperties`

Specifies the SD agent properties. Ignored if SD agent is not created.

`xsd:boolean sdAgentPropertiesSupplied (m)`

True or false depending on the value of `sdAgentProperties`.

`xsd:string user`

Specifies the user name. The default value is blank.

`xsd:boolean userSupplied (m)`

True or false depending on the value of `user`.

`xsd:string phone`

Specifies the user's phone number. The default value is blank.

`xsd:boolean phoneSupplied (m)`

True or false depending on the value of `phone`.

xsd:string location

Specifies the location of the user. The default value is blank.

xsd:boolean locationSupplied (m)

True or false depending on the value of location.

xsd:string comment

Specifies a comment.

xsd:boolean commentSupplied (m)

True or false depending on the value of comment.

SoftwareJobPermissionMask

This sequence of elements defines the permissions of a software job. Any combination of values can be set.

- **xsd:boolean renewAllowed**
- **xsd:boolean cancelAllowed**
- **xsd:boolean recoveryAllowed**
- **xsd:boolean rescheduleAllowed**
- **xsd:boolean suspendAllowed**
- **xsd:boolean resumeAllowed**
- **xsd:boolean userParametersAllowed**

SoftwareJobProperties

This sequence defines the properties that can be set for a software job.

xsd:string softwareJobId

The UUID identifying the software job.

xsd:string softwareJobName

The display name of the software job.

xsd:string softwarePackageId

The UUID identifying the software package with which this job is associated.

xsd:string softwarePackageName

The name of the software package with which this job is associated.

xsd:string softwarePackageVersion

The version of the software package with which this job is associated.

SoftwarePackageType softwarePackageType

The type of the software package with which this job is associated.

xsd:string softwarePackageProcedureId

The UUID identifying the software package procedure with which this job is associated.

xsd:string softwarePackageProcedureName

The name of the software package procedure with which this job is associated.

DateTime executionDate

The Date and Time when the job are to be executed.

DateTime createdDate

The date and time at which the job was created.

xsd:long timeoutHours

The time in hours after which time, if the job has been unable to complete, the job will cease attempts to complete.

SoftwareJobType softwareJobType

The type of the software job.

SoftwareJobState softwareJobState

The current state of the software job.

SoftwareJobStartupTime startWhen

The range for this parameter is by default 3-168 hours. If, for example, the value 1 hour is selected, it will immediately be replaced by 3 hours by the system. The default value is 7 days (168 hours).

SoftwareJobPermissionMask permissionMask

The permissions applicable to this software job.

SoftwareJobBootMask bootBeforeMask

This element indicates the reboot options requested for the software job before the job is executed.

SoftwareJobBootMask bootAfterMask

This element indicates the reboot options requested for the software job after the job is executed.

SoftwareJobOperationMask operationMask

Properties for the operation of the software job.

xsd:string userParameters

This parameter contains user-defined parameters for the job.

SoftwareJobStatusCounters jobStatusCounters

A summary of the status counts for the targets of this software job.

xsd:long internalMask

Internal mask for software job properties.

xsd:long orderNumber

If the software job is part of a job container, this value represents the order number of the job within its container.

xsd:long promptTimeoutHours

This indicates the number of hours for which a user prompt should be displayed before forced execution of the job. The default value is 1 day. This parameter should be set to a value that does not exceed the timeout value for the job.

SoftwareJobTask softwareJobTask

The task-type that the software job has been asked to complete.

xsd:string deliveryCalendar

The name of a Unicenter delivery calendar to be used for scheduling the job. The calendar works as ordinary Unicenter calendars, to control the days, dates and times when delivery is permitted. The time cells not set on in the Unicenter Calendar thus define the exclusion intervals.

If you try to define a calendar, the name of which is unknown to Unicenter, the job will not be created.

DateTime deliveryDate

The Date and time at which the job was delivered.

xsd:long cause

The job error or status cause. This indicates a return code or identifier.

xsd:string causeParam

The message ID of the message that explains the error (if any) associated with this software job.

SoftwareJobRenewState renewState

Indicates the current state of the job, if it has been renewed.

Remarks

None

SoftwareJobProperties2

This sequence defines the properties that can be set for a software job.

xsd:string softwareJobId

The UUID identifying the software job.

xsd:string softwareJobName

The display name of the software job.

xsd:string softwarePackageId

The UUID identifying the software package associated with this job.

xsd:string softwarePackageName

The name of the software package with which this job is associated.

xsd:string softwarePackageVersion

The version of the software package with which this job is associated.

SoftwarePackageType softwarePackageType

The type of the software package with which this job is associated.

xsd:string softwarePackageProcedureId

The UUID identifying the software package procedure with which this job is associated.

xsd:string softwarePackageProcedureName

The name of the software package procedure with which this job is associated.

DateTime executionDate

The Date and Time when the job are to be executed.

DateTime createdDate

The date and time when the job was created.

xsd:long timeoutHours

The time in hours after which time, if the job has been unable to complete, the job will cease attempts to complete.

SoftwareJobType softwareJobType

The type of the software job.

SoftwareJobState softwareJobState

The current state of the software job.

SoftwareJobStartupTime startWhen

The range for this parameter is by default 3-168 hours. If, for example, the value 1 hour is selected, it will immediately be replaced by 3 hours by the system. The default value is 7 days (168 hours).

SoftwareJobPermissionMask permissionMask

The permissions applicable to this software job.

SoftwareJobBootMask bootBeforeMask

This element indicates the reboot options requested for the software job before the job is executed.

SoftwareJobBootMask bootAfterMask

Indicates the reboot options requested for the software job after the job is executed.

SoftwareJobOperationMask2 operationMask2

Properties for the operation of the software job.

xsd:string userParameters

This parameter contains user-defined parameters for the job.

SoftwareJobStatusCounters jobStatusCounters

A summary of the status counts for the targets of this software job.

xsd:long internalMask

Internal mask for software job properties.

xsd:long orderNumber

If the software job is part of a job container, this value represents the order number of the job within its container.

xsd:long promptTimeoutHours

This indicates the number of hours for which a user prompt should be displayed before forced execution of the job. The default value is 1 day. This parameter should be set to a value that does not exceed the timeout value for the job.

SoftwareJobTask softwareJobTask

The task-type that the software job has been asked to complete.

xsd:string deliveryCalendar

The name of a Unicenter delivery calendar to be used for scheduling the job. The calendar works as ordinary Unicenter calendars, to control the days, dates and times when delivery is permitted. The time cells not set on in the Unicenter calendar thus define the exclusion intervals.

If you try to define a calendar, the name of which is unknown to Unicenter, the job will not be created.

DateTime deliveryDate

The Date and time at which the job was delivered.

xsd:long cause

The job error or status cause. This indicates a return code or identifier.

xsd:string causeParam

The message ID of the message that explains the error (if any) associated with this software job.

SoftwareJobRenewState renewState

Indicates the current state of the job, if it has been renewed.

SoftwareJobProperties3

The SoftwareJobProperties3 sequence defines the properties that you can set for a software job.

xsd:string softwareJobId

Specifies the UUID identifying the software job.

xsd:string softwareJobName

Specifies the display name of the software job.

xsd:string softwarePackageId

Specifies the UUID identifying the software package with which this job is associated.

xsd:string softwarePackageName

Specifies the name of the software package with which this job is associated.

xsd:string softwarePackageVersion

Specifies the version of the software package with which this job is associated.

SoftwarePackageType softwarePackageType

Specifies the type of the software package with which this job is associated.

xsd:string softwarePackageProcedureId

Specifies the UUID identifying the software package procedure with which this job is associated.

xsd:string softwarePackageProcedureName

Specifies the name of the software package procedure with which this job is associated.

DateTime executionDate

Specifies the date and time when you want to run the job.

DateTime createdDate

Specifies the date and time at which the job was created.

xsd:long timeoutHours

Specifies the time in hours after which, if the job has been unable to complete, the job will cease attempts to complete.

SoftwareJobType softwareJobType

Specifies the type of the software job.

SoftwareJobState softwareJobState

Specifies the current state of the software job.

SoftwareJobStartupTime startWhen

Specifies the time to start the software job. The range for this parameter is by default 3-168 hours. For example, if you select the value as 1 hour, the system immediately replaces the value with 3 hours. The default value is 7 days (168 hours).

SoftwareJobPermissionMask permissionMask

Specifies the permissions applicable to this software job.

SoftwareJobBootMask3 bootBeforeMask

Specifies the restart options requested for the software job before executing the job.

SoftwareJobBootMask3 bootAfterMask

Specifies the restart options requested for the software job after executing the job.

SoftwareJobOperationMask2 operationMask2

Specifies the properties for the operation of the software job.

xsd:string userParameters

Specifies the user-defined parameters for the job.

SoftwareJobStatusCounters jobStatusCounters

Specifies the summary of the status counts for the targets of this software job.

xsd:long internalMask

Specifies the internal mask for the software job properties.

xsd:long orderNumber

Specifies the order number of a software job within its job container if the software job is part of that job container.

xsd:long promptTimeoutHours

Specifies the number of hours for which you want to display a user prompt before a forced execution of the job starts. The default value is 1 day. You must set this parameter to a value that does not exceed the timeout value for the job.

SoftwareJobTask softwareJobTask

Specifies the type of the task that the software job is supposed to complete.

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to use for scheduling the job. The calendar works as an ordinary calendar to control the days, dates, and times when the delivery is permitted. The time cells that are not set in the calendar define the exclusion intervals. You cannot create the job if you try to define a calendar with the name unknown to SD.

DateTime deliveryDate

Specifies the date and time at which the job was delivered.

xsd:long cause

Indicates the job error or status cause, that is, indicates a return code or identifier.

xsd:string causeParam

Specifies the message ID of the message that explains the error (if any) associated with this software job.

SoftwareJobRenewState renewState

Specifies the current state of the job, if it has been renewed.

xsd:string userMessage

Specifies the custom administrator message.

Remarks

None

SoftwareJobPropertiesRequired

This sequence of elements specifies which software job properties are required when retrieving a software package group's properties.

- xsd:boolean softwareJobIdRequired
- xsd:boolean softwareJobNameRequired
- xsd:boolean softwarePackageIdRequired
- xsd:boolean softwarePackageNameRequired
- xsd:boolean softwarePackageVersionRequired
- xsd:boolean softwarePackageTypeRequired
- xsd:boolean softwarePackageProcedureIdRequired
- xsd:boolean softwarePackageProcedureNameRequired
- xsd:boolean executionDateRequired
- xsd:boolean createdDateRequired
- xsd:boolean timeoutHoursRequired
- xsd:boolean softwareJobTypeRequired
- xsd:boolean softwareJobStateRequired
- xsd:boolean startWhenRequired
- xsd:boolean permissionMaskRequired
- xsd:boolean bootBeforeMaskRequired
- xsd:boolean bootAfterMaskRequired
- xsd:boolean operationMaskRequired
- xsd:boolean userParametersRequired
- xsd:boolean jobStatusCountersRequired
- xsd:boolean internalMaskRequired
- xsd:boolean orderNumberRequired
- xsd:boolean promptTimeoutHoursRequired
- xsd:boolean softwareJobTaskRequired
- xsd:boolean deliveryCalendarRequired
- xsd:boolean deliveryDateTimeRequired
- xsd:boolean causeRequired
- xsd:boolean causeParamRequired
- xsd:boolean renewStateRequired

Remarks

For more details, see [SoftwareJobProperties](#) (see page 768).

SoftwareJobStatusCounters

This sequence of elements indicates the counts for the number of jobs or job targets that are currently in various different states.

- xsd:long waiting
- xsd:long active
- xsd:long ok
- xsd:long failed

Remarks

None

SoftwarePackageGroupPropertiesRequired

This sequence of elements specifies which software package group properties are required when retrieving a software package group's properties.

- xsd:boolean softwarePackageGroupIdRequired
- xsd:boolean noOfParentSoftwarePackageGroupsRequired
- xsd:boolean noOfChildSoftwarePackageGroupsRequired
- xsd:boolean noOfSoftwarePackagesRequired
- xsd:boolean nameRequired
- xsd:boolean commentRequired
- xsd:boolean creationDateRequired
- xsd:boolean changeDateRequired
- xsd:boolean attribMaskRequired
- xsd:boolean softwarePackageGroupPropertyRequired

Remarks

For more details, see [SoftwarePackageGroupProperties](#) (see page 828).

SoftwarePackageMask

This sequence specifies the software package attributes.

xsd:boolean checksum

A flag that can be set to indicate whether a checksum is performed to ensure package consistency

Remarks

None

SoftwarePackageMaskType

This mask is a sequence of variables that can be marked to indicate which types of software package are to be applied to the current request.

xsd:boolean all

All types of software.

xsd:boolean pgm

Ordinary software.

xsd:boolean swd

Detected software.

xsd:boolean userSelectable

Can be ordered from CA ITCM Catalog.

xsd:boolean sdsoftware

CA ITCM software.

Remarks

None

CreateSoftwarePolicyProperties

This sequence of elements specifies which software policy properties are to be initialized when a software policy is created.

- **xsd:string unitGroupId** (^m)
- **xsd:boolean unitGroupIdSupplied** (^m)
- **xsd:string name** (^m)
- **xsd:boolean nameSupplied** (^m)
- **xsd:string comment**

- xsd:boolean commentSupplied
- SoftwarePolicyEvaluationType softwarePolicyEvaluationType
Default: ET_SETUPANDEXECUTEJOBS
- xsd:boolean softwarePolicyEvaluationTypeSupplied
- SoftwareJobContainerPropertyMask softwareJobContainerPropertyMask
Default: linkageBatch and Cascade)
- xsd:boolean softwareJobContainerPropertyMaskSupplied
- EvaluationTypeMask evaluationTypeMask
Default: Once
- xsd:boolean evaluationTypeMaskSupplied
- xsd:string evaluationCalendar
- xsd:boolean evaluationCalendarSupplied
- xsd:long evaluationPeriod
Default: 24
- xsd:boolean evaluationPeriodSupplied
- DateTime nextEvaluationDate
Default: Current date and time of the manager computer
- xsd:boolean nextEvaluationDateSupplied

Remarks

For more details, see [SoftwarePolicyProperties](#) (see page 851).

CreateSoftwarePolicyProperties3

The CreateSoftwarePolicyProperties3 sequence of elements specifies which software policy properties are to be initialized when a software policy is created.

xsd:string policyName (m)

Specifies the name of policy to be created.

xsd:string comment

Specifies the comment to be set for the policy.

xsd:boolean commentSupplied (m)

Specifies true or false depending on whether the comment parameter is supplied.

xsd:long priority

Specifies the priority that can range from 1 (highest) to 10 (lowest). If no value is provided, the default value, 5, is used.

xsd:long prioritySupplied (m)

Specifies true or false depending on whether the priority parameter is supplied.

CreateSoftwareJobLinkage createSoftwareJobLinkage

Sets the software job linkage related properties.

xsd:boolean createSoftwareJobLinkageSupplied (m)

Specifies true or false depending on whether the createSoftwareJobLinkage parameter is supplied.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

Specifies the type of the evaluation that is performed when the policy is applied to the group. The default value is SP_POPULATEEXCEPTIONS.

xsd:boolean softwarePolicyEvaluationTypeSupplied (m)

Specifies true or false depending on whether the softwarePolicyEvaluationType parameter is supplied.

SoftwarePolicyEvaluationPeriodType softwarePolicyEvaluationPeriodType

Specifies whether to evaluate the software policy once or periodically. If you decide to evaluate the policy periodically, the evaluation is done based on the evaluation period set. The default value is once.

xsd:boolean softwarePolicyEvaluationPeriodTypeSupplied (m)

Specifies true or false depending on whether the softwarePolicyEvaluationPeriodType parameter is supplied.

DateTime nextEvaluationDate

Specifies the date and time for the next evaluation. The default value is the current date and time on the manager computer.

xsd:boolean nextEvaluationDateSupplied (m)

Specifies true or false depending on whether the nextEvaluationDate parameter is supplied.

xsd:long evaluationPeriod

Specifies the period of the evaluation in hours. The default value is 1440 minutes or 24 hours.

xsd:boolean evaluationPeriodSupplied (m)

Specifies true or false depending on whether the evaluationPeriod parameter is supplied.

xsd:boolean enterpriseTime

(If set to true) Specifies that the time specifications are based on the enterprise manager time. By default, the parameter is set to false.

xsd:boolean enterpriseTimeSupplied (m)

Specifies true or false depending on whether the enterpriseTime parameter is supplied.

xsd:string evaluationCalendar

Specifies the name of the calendar that controls the time when the evaluation can take place. To activate a evaluationCalendar you must specify "evaluationPeriod". The default value is an empty string.

xsd:boolean evaluationCalendarSupplied (m)

Specifies true or false depending on whether the evaluationCalendar parameter is supplied.

AdhocTargetEvaluation adhocTargetEvaluation

Specifies the type of the target evaluation. The target evaluation is triggered when agents report the inventory to the manager. The default value is NO_ADHOC.

xsd:boolean adhocTargetEvaluationSupplied (m)

Specifies true or false depending on whether the adhocTargetEvaluation parameter is supplied.

xsd:boolean regenerate

Regenerates the evaluation type. This parameter is applicable only if evaluationType is SP_SETUPJOBS or SP_SETUPANDEXECUTEJOBS. The default value is true.

xsd:boolean regenerateSupplied (m)

Specifies true or false depending on whether the regenerate parameter is supplied.

xsd:boolean evaluateDynamicGroups

(Optional) Evaluates the dynamic groups while evaluating the software policy provided the value is set to true. The default value is false.

xsd:boolean evaluateDynamicGroupsSupplied (m)

Specifies true or false depending on whether the evaluateDynamicGroups parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

userMessageSupplied

Checks whether or not the custom administrator message is supplied.

Remarks

None

CreateUnitGroupProperties

This sequence of elements specifies which volume properties are to be initialized when a unit group is created.

- xsd:string groupLabel (^m)
- xsd:boolean groupLabelSupplied

Note: Set this value to true because groupLabel is mandatory.
- xsd:string groupDescription
- xsd:boolean groupDescriptionSupplied

Note: Set the value to true or false depending on the groupDescription value.

Remarks

For more details, see [UnitGroupProperties](#) (see page 870).

SoftwareJobBootMask (bootAfterMask)

A series of flags that indicate the reboot options for a software job.

xsd:boolean noReboot

Do not reboot the computer after the job has ended.

xsd:boolean reboot

Reboot the computer after the job has ended.

xsd:boolean logoff

If logoff is set to true, the user agent logs off the user, and the machine agent logs off all users before the job starts.

xsd:boolean rebootAfterLast

Reboot the computer when last job in the container has ended.

xsd:boolean logoffAfterLast

if logoffAfterLast is set to true, the user agent logs off the user and the machine agent logs off all users when the last job in the container has ended.

Remarks

Only one flag may be set to true at any one time.

SoftwareJobBootMask3 (bootAfterMask)

This sequence includes a series of flags that indicate the reboot options for a software job.

xsd:boolean noReboot

Specifies that the computer must not be restarted after completion of the job.

xsd:boolean reboot

Restarts the computer after completion of the job.

xsd:boolean logoff

(If set to true) Specifies that the user agent logs off the user and the computer agent logs off all the users before the job starts.

xsd:boolean rebootAfterLast

Restarts the computer after completion of the last job in the container.

xsd:boolean logoffAfterLast

(If set to true) Specifies that the user agent logs off the user and the computer agent logs off all the users after completion of the last job in the container.

xsd:boolean ShutdownAfterLast

(If set to true) Specifies that the target computer shuts down after completion of the last SD job.

Remarks

Only one flag may be set to true at a time.

SoftwareJobBootMask (bootBeforeMask)

A series of flags that indicate the reboot options for a software job.

xsd:boolean noReboot

Do not reboot the computer before starting the job.

xsd:boolean reboot

Reboot the computer before starting the job.

xsd:boolean logoff

If logoff is set to true, the user agent logs off the user and the machine agent logs off all users before the job starts.

xsd:boolean rebootAfterLast

Not applicable.

xsd:boolean logoffAfterLast

Not applicable.

Remarks

Only one flag may be set to true at any one time.

SoftwareJobBootMask3 (bootBeforeMask)

This sequence includes a series of flags that indicate the restart options for a software job.

xsd:boolean noReboot

Specifies that the computer must not be restarted before starting the job.

xsd:boolean reboot

Restarts the computer before starting the job.

xsd:boolean logoff

(If set to true) Specifies that the user agent logs off the user and the computer agent logs off all the users before the job starts.

xsd:boolean rebootAfterLast

Not applicable.

xsd:boolean logoffAfterLast

Not applicable.

xsd:boolean ShutdownAfterLast

Not applicable.

Remarks

Only one flag may be set to true at a time.

SoftwareJobContainerProperties

This sequence defines the properties for the software job container.

xsd:string softwareJobContainerId

The UUID identifying the software job container.

SoftwareJobContainerState state

The current state of the software job container

xsd:boolean seal

This element indicates whether the container is currently sealed.

xsd:long noOfParentSoftwareJobContainers

The number of software job containers by which this container is contained.

xsd:long noOfChildSoftwareJobContainers

The number of software job containers that are contained by this container.

xsd:long noOfSoftwareJobs

The number of software jobs contained in this software job container.

xsd:string name

The display name of the job container.

xsd:string comment

A comment for the job container.

SoftwareJobContainerPropertyMask**softwareJobContainerPropertyMask**

Mask for the software job container property

DateTime creationDate

The Date and Time at which the job container was created.

DateTime changeDate

The Date and Time at which the job container was last updated.

AttributeMask attribMask

A series of attributes that describe the software job container.

xsd:long waitCount

The number of jobs in the container that are not yet activated.

xsd:long activeCount

The number of jobs in the container that have been activated.

xsd:long errorCount

The number of jobs in the container that have failed.

xsd:long okCount

The number of jobs in the container that have successfully completed.

xsd:long renewState

If the job container has been renewed, this element indicates the current state of that renewal.

Remarks

None

CreateAssetScriptJobResponse

The CreateAssetScriptJobResponse sequence includes the following parameters related to the creation of the asset script job:

xsd:string assetJobUid

Contains the UUID of the new asset job.

xsd:long assetJobID

Contains the ID of the new asset job.

Xsd:long domainId

Contains the domain ID of the new asset job

AssetScriptJobProperties

The AssetScriptJobProperties sequence includes the job properties for the asset script job.

xsd:string jobName

Specifies the job name.

xsd:boolean bJobDescriptionSupplied

Specifies whether the job description is supplied for the job.

xsd:string jobDescription

Specifies the description for the job.

xsd:boolean bJobScriptSupplied

Specifies whether the script is supplied for the job.

xsd:string jobScript

Specifies the script to be executed on the agent for this script job.

xsd:boolean bSchedulingOptionsSupplied

Specifies whether the scheduling options for this job are supplied.

dsm:schedulingOptions scheduleProperties

Specifies the scheduling options for the asset job.

More information:

[SchedulingOptions](#) (see page 789)

SchedulingOptions

The SchedulingOptions sequence is used to schedule a script job to run at specified occasions.

dsm:scheduling scheduling

Specifies the frequently of job execution.

dsm:conditions conditions

Specifies various conditions for executing the job such as the date time range, days of the week, and periodic dependency.

dsm:dependency dependency

Specifies the factors to consider for executing the job at the scheduled time.

dsm:preJob prejob

Specifies the actions to be performed before executing the job.

dsm:postJob postJob

Specifies the actions to be performed after executing the job.

dsm:miscellaneous miscellaneous

Specifies the parameters for unattended execution of the job.

More information:

[Scheduling](#) (see page 789)

[Conditions](#) (see page 790)

[Dependency](#) (see page 790)

[PreJob](#) (see page 791)

[PostJob](#) (see page 792)

[Miscellaneous](#) (see page 792)

Scheduling

The Scheduling sequence is used to specify how frequently you want to execute the job.

xsd:int schedule

This field can have following values.

RUN_ONLY_ONCE

Runs the job only the first time when the asset management agent runs on the computers that are linked with this job.

ALWAYS_RUN_JOB

Runs the job whenever the asset management agent runs on the computers that are linked with this job.

OTHER_SCHEDULING_RUN

Runs the job for specified number of times (at most 99) on the computers linked with this job.

Default: ALWAYS_RUN_JOB

Conditions

The Conditions sequence is a scheduling option used to specify the conditions for executing the job.

dsm:DateTimeRestrictions datetimerestrictions

Specifies date and time range for executing the job.

xsd:boolean bRunOnlyOnTheseDays

Specifies whether the job must be run only on the selected days of a week.

dsm:ArrayOfDayOfTheWeekDependency dayOfTheWeekDependency

Specifies the days in a week when the job must be run. If you set `bRunOnlyOnTheseDays` parameter to true, you must specify the days of the week in this parameter.

xsd:boolean blsPeriodDepen

Specifies whether there is a periodic dependency.

dsm:PeriodicDependency periodicDependency

Specifies the options for periodic dependency.

Dependency

The Dependency sequence is a scheduling options used to specify the dependency factors for executing a job.

xsd:boolean bJobDependency

Specifies whether the execution of the job is dependent on the successful completion of another job.

dsm:JobDependency jobDependency

Specifies the job dependency options.

xsd:boolean bFileDependency

Specifies whether the execution of the job is dependent on the existence of a file.

dsm:FileDependency fileDependency

Specifies file dependency options.

xsd:boolean bJobConLimit

Specifies whether there is any limit to the maximum number of simultaneous executions of an action per scalability server.

xsd:int allowOnlyNumExecutionsAtATime

Specifies the number of simultaneous executions allowed.

More information:

[FileDependency](#) (see page 793)

[JobDependency](#) (see page 793)

PreJob

The PreJob sequence is a scheduling option used to specify the actions to be performed before executing the job.

xsd:boolean bJobExecProperties

Specifies whether you want to provide the job execution properties.

dsm:JobExecProperties jobExecProperties

Specifies the job execution properties.

xsd:boolean bNetWorkDriveInfoSupplied

Specifies whether you want to provide the network drive information.

dsm:NetWorkDrive netWorkDrive

Provides the required information to map network drive before job execution.

More information:

[JobExecProperties](#) (see page 794)

[NetWorkDrive](#) (see page 795)

PostJob

The PostJob sequence is a scheduling option used to specify the actions to be performed after executing the job.

xsd:boolean bpostJobProperties

Specifies whether you want to provide post job execution properties.

dsm:PostJobProperties postJobProperties

Specifies the post job execution options.

xsd:boolean bDeleteJobWhenSucceeded

Specifies whether you want the job to be deleted after the successful execution.

Miscellaneous

The Miscellaneous sequence is a scheduling option used to run the job unattended.

xsd:boolean bJobisallowedtorununattended

Specifies whether you want to run the job unattended, that is, without any user interaction.

DateTimeRestrictions

The DateTimeRestrictions sequence is a scheduling option used to specify the date and time restrictions for executing the job.

xsd:boolean bRunOnlywhenwithindatarange

Specifies whether you want to provide the date range for executing the job.

xsd:boolean bRunOnlywhenwithintimerange

Specifies whether you want to provide the time range for executing the job.

dsm:DateTime StartDatetime

Defines the starting date and time for executing the job.

dsm:DateTime EndDatetime

Defines the ending date and time for executing the job.

FileDependency

The FileDependency sequence is a scheduling option used to specify the file dependency options for executing a job.

xsd:string filePath

Specifies the path to the file that you want to check.

xsd:boolean bFileExists

Specifies whether you want to perform an action depending on the existence of the selected file. Set this parameter to false to execute an action if the file does not exist.

JobDependency

The JobDependency sequence is a scheduling option used to specify the job dependency options for executing a job.

xsd:long dependentJobDomainId

Defines the domain ID of the dependent job.

xsd:long dependentJobId

Defines the job ID of the dependent job.

Xsd:string dependentJobUid

Defines the UUID of the dependent job.

xsd:boolean bCurrentJobAttachedToRelatedAsset

Specifies whether you want to run the scheduled job only on completion of a dependent job on the related assets. If the scheduled job is linked to a computer, the dependent job must have been executed on the computer before the scheduled job runs. If the scheduled job is linked to a user, the dependent job must have been executed in the user login before the scheduled job runs.

PeriodicDependency

The PeriodicDependency is a scheduling option used to specify the periodic dependency options for executing the job.

dsm:PERIODTYPE periodType

Specifies that you want to Executes an action based on this period type.

xsd:int period

Specifies the period after which the job must be executed. Depending on the period type, this parameter takes the following values:

Day: number of hours

For example, specify 21 to execute the job at 9PM every day.

Week: Day in a week

For example, specify 3 to execute the job on every Wednesday.

Month: Day in a month

For example, specify 12 to execute the job on the day 12 of every month.

JobExecProperties

The JobExecProperties sequence is a scheduling option used to specify the other properties for job execution.

xsd:string PromptMessage

Defines the message text to be shown on the computer before executing the job.

xsd:string messageOK

Displays the Yes button in the message. You can customize this to match your message.

Default: &YES

xsd:string messageCancel

Displays the No button in the message box. You can customize this to match your message.

Default: &NO

xsd:boolean bDefaultMessage

Specifies whether messageOK must be applied by default. If this parameter is set to false, messageCancel is applied.

xsd:boolean bAutoSelect

Specifies whether you want to enable the auto select option.

xsd:boolean bAutoSelectValue

Specifies whether you want to specify the auto select value.

xsd:long autoSelectAfterSecond

Specifies the time after which the auto select value is applied.

xsd:boolean bForceExecutionAfterNumCancellations

Specifies whether you want to force the execution after certain number of cancellations.

xsd:int forceExecutionAfterNumCancellations

Specifies the number of cancellations after which you want to force the execution of the job. This ensures that the user can cancel an action only a limited number of times.

xsd:boolean bForceExecutionAfter

Specifies whether you want to force the execution of the job after a particular date. This ensures that the user can cancel an action only until a particular date.

xsd:dateTime ForceExecutionAfter

Specifies the date when you want to force the execution of the job.

NetWorkDrive

The NetWorkDrive sequence is a scheduling option used to provide the network drive details for executing a job,

xsd:string drive

Specifies the drive letter to which you want to map the share.

xsd:string sharePath

Specifies the name of the share.

Default: \\SERVER\SHARE

xsd:boolean bDisconnectWhenDone

Specifies whether you want to disconnect the temporary drive connection after executing the job.

PostJobProperties

The PostJobProperties sequence is a scheduling option used to specify the post job properties.

xsd:boolean bMessageAndTimeOut

Specifies whether you want to reboot the windows agent computer after executing the job.

xsd:string messageBeforeReboot

Specifies the message text to be displayed before the computer restarts.

xsd:int messageTimeout

Defines the number of seconds the message must be visible. Specify 0 to give infinite timeout.

xsd:boolean bRestartOS

Restarts the agent computer.

AssetScriptJobPropertiesRequired

The AssetScriptJobPropertiesRequired sequence is used to retrieve the asset job properties.

xsd:boolean assetJobNameRequired

Specifies whether you want to retrieve the asset script job name.

xsd:boolean assetJobDescriptionRequired

Specifies whether you want to retrieve the asset script job description.

xsd:boolean assetJobScriptRequired

Specifies whether you want to retrieve the asset job script.

xsd:boolean assetJobSchedulingOptionsRequired

Specifies whether you want to retrieve the scheduling options of the asset job.

Dsm:OpenAssetScriptJobListResponse

The OpenAssetScriptJobListResponse sequence includes the following parameters related to asset script job list:

Xsd:long OpenListHandle

Contains the ListHandle to the list opened.

Xsd:long totalNoOfScriptJobs

Contains the total number of script jobs in the list.

GetAssetScriptJobResponse

The GetAssetScriptJobResponse sequence contains the response for the GetAssetScriptJob method.

xsd:long assetJobID

Contains the ID of the asset job.

Xsd:long domainId

Contains the domain ID of the new asset job

dsm:AssetScriptJobProperties assetJobProperties

Contains the properties retrieved for a given asset script job.

DayOfTheWeekDependency

The DayOfTheWeekDependency sequence is a scheduling option used to specify a week day.

dsm:WeekDay weekdays

Specifies a day in the week.

More information:

[WeekDay](#) (see page 109)

GetAssetScriptJobsProperties

The GetAssetScriptJobsProperties sequence contains properties of all the asset script jobs.

xsd:string assetJobUid

Contains the UUID of the asset job.

Xsd:long domainId

Contains the domain ID of the asset job.

xsd:string jobName

Contains asset job script name.

xsd:boolean bJobDescriptionSupplied

Specifies whether the job description is supplied for the job.

xsd:string jobDescription

Contains the description about the job.

xsd:boolean bJobScriptSupplied

Specifies whether the script is supplied for the job.

xsd:string jobScript

Contains the script to be executed on the agent for the script job.

xsd:boolean bSchedulingOptionsSupplied

Specifies whether the scheduling options of this job are supplied

dsm:schedulingOptions scheduleProperties

Contains the scheduling options for the job.

TriggerInfo

The TriggerInfo sequence is used to specify the information to trigger the asset management and software delivery job check.

int nCFComponentCode

Specifies the [CF component code](#) (see page 110).

CO_TRIGGER_UNIT_TYPE nUnitType

Specifies the unit type.

xsd:string objDbUuid;

Specifies the UUID of the object

BOOL wakeOnLAN

Specifies the whether you want to send the wake-on-LAN request.

More information:

[CO_TRIGGER_UNIT_TYPE](#) (see page 110)

OSIMProperties

Specifies the OSIM properties.

xsd:string propertyName

Specifies the OSIM property name.

xsd:string propertyValue

Specifies the OSIM property value.

OSIMPropertiesWithDesc

The OSIMPropertiesWithDesc includes the OSIM property name, property value, and an array of description of the property name and property value.

xsd:string propertyName

Specifies the OSIM property name.

xsd:string propertyValue

Specifies the OSIM property value.

dsm:ArrayOfOSIMparameterDesc osiParamDescArray

Specifies the Array description for an OSIM property.

OSIMparameterDesc

The OSIMparameterDesc includes the OSIM parameter description name and description value.

xsd:string propertyName

Specifies the OSIM property description name.

sd:string propertyValue

Specifies the OSIM property description value.

ScalabilityServerProperties

Specifies the scalability server properties.

xsd:string serverLabel

Specifies the name of the scalability server.

xsd:string serverHostName

Specifies the host name of the scalability server.

xsd:string serverUUID

Specifies the UUID of the scalability server.

xsd:string domainUUID

Specifies the domain UUID of the scalability server.

xsd:string serverDescription

Includes brief information about the server.

xsd:long serverversion

Specifies the version of the scalability server.

dsm:DateTime lastEvaluationDateTime

Identifies the last evaluated registration date and time of the scalability server.

dsm:DateTime creationDateTime

Identifies the date and time of the scalability server registration.

xsd:string creationUser

Specifies the name of the user account that created the scalability server.

dsm:DateTime lastUpdateDateTime

Specifies the last updated date and time of the scalability server.

xsd:string lastUpdateUser

Specifies the name of the user account that last updated the scalability server.

xsd:string sharePath

Specifies the UNC share path of the server.

ScalabilityServerPropertiesRequired

The following sequences of elements specify the properties required to retrieve the scalability server properties.

xsd:boolean serverLabelRequired

Specifies whether the scalability server name is required.

xsd:boolean serverHostNameRequired

Specifies whether the scalability server host name is required.

xsd:boolean serverUUIDRequired

Specifies whether the scalability server UUID is required.

xsd:boolean domainUUIDRequired

Specifies whether the scalability server domain UUID is required.

xsd:boolean serverDescriptionRequired

Specifies whether the scalability server description is required.

xsd:boolean serverVersionRequired

Specifies whether the scalability server version is required.

xsd:boolean lastEvaluationDateTimeRequired

Specifies whether the last date and time of evaluation of the scalability server name is required.

xsd:boolean creationUserRequired

Specifies whether the details about the user who created the scalability server is required.

xsd:boolean creationDateTimeRequired

Specifies whether the date and time of creation of the scalability server is required.

xsd:boolean lastUpdateDateTimeRequired

Specifies whether the date and time of last update of the scalability server is required.

xsd:boolean lastUpdateUserRequired

Specifies whether the details about the user who last updated the scalability server is required.

xsd:boolean sharePathRequired

Specifies whether the scalability server UNC share path is required.

OSImageFilter

The following sequences of elements specify a search filter when you are searching or listing OS images.

dsm:OSIMAGEPROPERTY osImageProperty

Specifies the search property.

xsd:string searchString

Specifies the value of the search filter to be applied when the search is performed on a string property.

dsm:DateTime2 dateTime

Specifies the changed or activated date and time of the target computer.

dsm:FILTERCONDITION condition

Specifies the filtercondition to be applied.

OSImageProperties

Specifies the OS image properties.

xsd:string imageName

Specifies the name of the OS image.

xsd:string typeIndicator

Specifies OS image type.

xsd:string operatingSystem

Specifies the operating environment of OS image.

dsm:DateTime2 registeredAt

Specifies the date and time of the OS image registration with the domain manager.

xsd:string sdPackageName

Specifies the name of the related SD package.

xsd:string sdPackageVersion

Specifies the version of the related SD package.

xsd:string sdPackageComment

Specifies the comment related with the SD package.

xsd:string sdOSCode

Specifies the SD code of the OS associated with the OS image.

xsd:string sdOSLocale

Specifies the locale of the OS image.

xsd:string string osImageFlags

Specifies the image flags associated with the OS image.

OSIMTargetFilter

The following sequences of elements specify a search filter when you are searching or listing OS images.

dsm:OSIMTARGETPROPERTY osimTargetProperty

Specifies the search property.

xsd:string searchString

Specifies the value of the search filter to be applied when the search is performed on a string property.

dsm:DateTime2 stateChangedDateTime

Specifies the changed or activated date and time of the target computer.

dsm:FILTERCONDITION condition

Specifies the filtercondition to be applied.

dsm:WS-OSIM-CONFIG-STATE configState

Specifies the configuration state of the OSIM target computer.

OSIMTargetProperties

The following are the OSIMTargetProperties:

xsd:string targetName

Specifies the name of the target computer.

xsd:string targetMACAddress

Specifies MAC Address of the target computer.

dsm:WS-OSIM-CONFIG-STATE configType

Specifies the configuration state of the target computer.

dsm:DateTime2 stateChangedDateTime

Specifies the changed or activated date and time of the target computer.

xsd:string imageName

Specifies the name of the OS image.

ComputerPropertyFilter2

The following sequences of elements specify a search filter when you are searching or listing computers.

dsm:ComputerProperty2 computerProperty

Specifies the computer property to be searched.

dsm:ComputerFilterCondition filterCondition

Specifies the filter condition.

xsd:string searchString

Specifies the value of the search filter to be applied when the search is performed on a string property.

This field is mandatory if computerProperty is set to one of the following values:

- COMPUTER_UUID
- COMPUTER_LABEL
- COMPUTER_DOMAIN_UUID

- COMPUTER_PRIMARY_NETWORK_ADDRESS
- COMPUTER_HOST_UUID
- COMPUTER_ASSET_SOURCE_UUID
- COMPUTER_HOST_NAME
- COMPUTER_SERIAL_NUMBER
- COMPUTER_DISC_SERIAL_NUMBER
- COMPUTER_PRIMARY_MAC_ADDRESS
- COMPUTER_ASSET_TAG
- COMPUTER_URI
- COMPUTER_SYSTEM_ID
- COMPUTER_VENDOR_NAME
- COMPUTER_DOMAIN_LABEL
- COMPUTER_CREATION_USER
- COMPUTER_LAST_UPDATED_USER
- COMPUTER_AGENT_NAME
- COMPUTER_AGENT_DESCRIPTION
- COMPUTER-USER-DATA1
- COMPUTER-USER-DATA2
- COMPUTER-USER-DATA3
- COMPUTER-USER-DATA4
- COMPUTER-AGENT-SS
- COMPUTER-AGENT-IPADDRESS
- COMPUTER-AGENT-ORIGIN
- COMPUTER-ASSET-TAG

xsd:long searchValue

Specifies the value of the search filter to be applied when the search is performed on a numeric property.

This field is mandatory if computerProperty is set to one of the following values:

- COMPUTER_CLASSID
- COMPUTER_USAGE_LIST
- COMPUTER_VERSION_NUMBER
- COMPUTER-AGENT-TRUSTLEVEL

xsd:dateTime2 dateTime

Specifies the value of the search filter to be applied when the search is performed on date and time.

This field is mandatory if computerProperty is set to one of the following values:

- COMPUTER_CREATION_DATE
- COMPUTER_LAST_UPDATED_DATE
- COMPUTER-AGENT-CREATION-DATETIME
- COMPUTER-AGENT-LASTRUN-DATETIME

xsd:AMAgentStatus amstatus

Specifies the value of the search filter to be applied when the search is performed on AMAgent status.

COMPUTER_AGENT_AM_STATUS

xsd:SDAgentStatus sdstatus

Specifies the value of the search filter to be applied when the search is performed on SDAgent status.

COMPUTER_AGENT_SD_STATUS

xsd:RCAgentStatus amstatus

Specifies the value of the search filter to be applied when the search is performed on RCAgent status.

COMPUTER_AGENT_RC_STATUS

xsd:ProtectionStatus protectionstatus

Specifies the value of the search filter to be applied when the search is performed on Protection status.

COMPUTER_AGENT_PT_STATUS

xsd:ASPStatus amstatus

Specifies the value of the search filter to be applied when the search is performed on Anti-Spyware agent status.

COMPUTER_AGENT_ASP_STATUS

xsd:AVStatus amstatus

Specifies the value of the search filter to be applied when the search is performed on Anti-Virus agent status.

COMPUTER_AGENT_AV_STATUS

xsd:HIPSStatus amstatus

Specifies the value of the search filter to be applied when the search is performed on HIPS status.

COMPUTER_AGENT_HIPS_STATUS

dsm:AgentRestrictions restrictions

Specifies the value of the search filter to be applied when the search is performed on agent restrictions.

COMPUTER-AGENT-RESTRICTIONS

ComputerPropertiesRequired2

The following sequences of elements specify the properties required when you are retrieving properties of a computer:

xsd:boolean computerUUIDRequired

Specifies whether computer UUID is required

xsd:boolean computerLabelRequired

Specifies whether computer name or label is required

xsd:boolean computerDomainUUIDRequired

Specifies whether computer domain UUID is required.

xsd:boolean computerPrimaryNetworkAddressRequired

Specifies whether computer primary network address is required.

xsd:boolean computerHostUUIDRequired

Specifies whether computer host UUID is required.

xsd:boolean computerAssetSourceUUIDRequired

Specifies whether computer Asset source UUID is required.

xsd:boolean computerHostNameRequired

Specifies whether computer Hostname is required.

xsd:boolean computerSerialNumberRequired

set to true, if computer serial number is required.

xsd:boolean computerDiscSerialNumberRequired

Specifies whether computer Disc serial number is required.

xsd:boolean computerPrimaryMACAddressRequired

Specifies whether computer MAC Address is required.

xsd:boolean computerAssetTagRequired

Specifies whether computer Asset tag is required.

xsd:boolean computerClassIDRequired

Specifies whether computer OS is required.

xsd:boolean computerUsageListRequired

Specifies whether computer usage list is required.

xsd:boolean computerURIRequired

Specifies whether computer URI is required.

xsd:boolean computerSystemIDRequired

Specifies whether computer system ID is required.

xsd:boolean computerVendorNameRequired

Specifies whether computer vendor name is required.

xsd:boolean computerDomainLabelRequired

Specifies whether computer Domain Label is required.

xsd:boolean computerCreationUserRequired

Specifies whether the details about the user who created the computer is required.

xsd:boolean computerCreationDateRequired

Specifies whether computer creation date is required.

xsd:boolean computerLastUpdatedUserRequired

Specifies whether the details about the user who last updated the computer is required.

xsd:boolean computerLastUpdatedDateRequired

Specifies whether computer last updated date time is required.

xsd:boolean computerVersionNumberRequired

Specifies whether computer version number is required.

xsd:boolean agentNameRequired

Specifies whether computer agent name is required.

xsd:boolean agentRestrictionsRequired

Specifies whether agent restrictions is required.

xsd:boolean agentDescriptionRequired

Specifies whether computer Asset description is required.

xsd:boolean userData1Required

Specifies whether computer user defined data1 is required.

xsd:boolean userData2Required

Specifies whether computer user defined data2 is required.

xsd:boolean userData3Required

Specifies whether computer user defined data3 is required.

xsd:boolean userData4Required

Specifies whether computer user defined data4 is required.

xsd:boolean scalabilityServerNameRequired

Specifies whether Agent Scalability server name is required.

xsd:boolean agentCreationDateTimeRequired

Specifies whether agent creation date and time is required.

xsd:boolean lastRunDateTimeRequired

Specifies whether agent last run date and time is required.

xsd:boolean agentIPAddressRequired

Specifies whether agent IP Address is required.

xsd:boolean agentTrustlevelRequired

Specifies whether agent trust level is required.

xsd:boolean agentOriginRequired

Specifies whether agent origin is required.

xsd:boolean amStatusRequired

Specifies whether AM agent status is required.

xsd:boolean sdStatusRequired

Specifies whether SD agent status is required.

xsd:boolean rcStatusRequired

Specifies whether RC agent status is required.

xsd:boolean protectionStatusRequired

Specifies whether Protection agent status is required.

xsd:boolean aspStatusRequired

Specifies whether Anti-Spyware agent status is required

xsd:boolean avStatusRequired

Specifies whether Anti-Virus agent status is required

xsd:boolean hipsStatusRequired

Specifies whether HIPS agent status is required.

DateTime2

Specifies the date and time structure.

The following sequence of elements represents the date and time structure:

- `xsd:int year`
- `xsd:int month`
- `xsd:int dayOfMonth`
- `xsd:int hour`
- `xsd:int minute`
- `xsd:int seconds`

ComputerProperties2

The following sequences of elements specify the properties required when you are retrieving the properties of a computer.

`xsd:string computerUUID`

Specifies the UUID of the computer.

`xsd:string computerLabel`

Specifies the display name of the computer.

`xsd:string computerDomainUUID`

Specifies the UUID identifying the CA ITCM domain of the computer.

`xsd:string computerPrimaryNetworkAddress`

Specifies the IP address of the computer in 0.0.0.0 format.

`xsd:string computerHostUUID`

Specifies the agent-generated UUID that identifies the computer.

`xsd:string computerAssetSourceUUID`

Specifies the UUID identifying the common CA asset to which this computer is linked.

`xsd:string computerHostName`

Specifies the host name of the computer.

`xsd:string computerSerialNumber`

Specifies the serial number of the computer.

`xsd:string computerDiscSerialNumber`

Specifies the serial number of the computer disc.

xsd:string computerPrimaryMACAddress

Specifies the MAC address of the computer. The string must have a maximum length of 64 characters.

xsd:string computerAssetTag

Specifies the string value that identifies a computer asset.

xsd:long computerClassID

Specifies the numeric value indicating the class of the computer object. For example, Operating System.

dsm:UsageListMask computerUsageList

Specifies the computer usage list mask.

xsd:string computerURI

Specifies the fully qualified Uniform Resource Identifier of the computer.

xsd:string computerSystemID

Specifies the user-defined identifier for computer.

xsd:string computerVendorName

Specifies the name of the computer vendor.

xsd:string computerDomainLabel

Specifies the display name of the DSM domain to which the computer belongs.

xsd:string computerCreationUser

Specifies the name of the user account that created the computer record.

dsm:DateTime computerCreationDate

Specifies the computer creation date and time.

xsd:string computerLastUpdatedUser

Specifies the name of the user account that last updated the computer properties.

dsm:DateTime computerLastUpdatedDate

Specifies the date and time when the computer properties were last updated.

xsd:long computerVersionNumber

Specifies the version number of the computer record.

Note: The version number is incremented each time a record is updated.

xsd:string agentName

Specifies the agent name of the computer.

xsd:long agentRestrictions

Specifies the agent restrictions.

xsd:string agentDescription

Specifies the agent description.

xsd:string userdata1

Specifies the user defined data1.

xsd:string userdata2

Specifies the user defined data2.

xsd:string userdata3

Specifies the user defined data3.

xsd:string userdata4

Specifies the user defined data4.

xsd:string scalabilityserverName

Specifies the name of the scalability server that the agent is pointing to.

dsm:DateTime2 agentCreationDateTime

Specifies the agent creation data and time.

dsm:DateTime2 lastRunDateTime

Specifies the agent last run date and time.

xsd:string agentIPAddress

Specifies the IP address of the agent.

xsd:long agentTrustlevel

Specifies the trust level of the agent.

xsd:string agentOrigin

Specifies the origin of the agent.

dsm:AMAgentStatus amStatus

Specifies the AM Agent status.

dsm:SDAgentStatus sdStatus

Specifies the SD Agent status.

dsm:RCAgentStatus rcStatus

Specifies the RC Agent status.

dsm:ProtectionStatus protectionStatus

Specifies the Protection Agent status.

dsm:ASPStatus aspStatus

Specifies the ASP status.

dsm:AVStatus avStatus

Specifies the AV status.

dsm:HIPSStatus hipsStatus

Specifies the HIPS status.

CreateOSIMComputerProperties

Specifies the properties required to create an OSIM computer.

The following are the properties:

xsd:string computerLabel

Specifies the name of the computer label to be created.

xsd:string computerHostname

Specifies the name of the host name of the computer to be created.

xsd:boolean computerHostNameSupplied

Specifies if the host name of the computer is specified.

xsd:string computerPrimaryNetworkAddress

Specifies the IP address of the computer to be created.

xsd:string serialNumber

Specifies the serial number of the computer to be created.

xsd:boolean serialNumberSupplied

Specifies if the computer serial number is specified.

xsd:string assetTag

Specifies the asset tag of the computer to be created.

xsd:boolean assetTagSupplied

Specifies if the computer asset tag is specified.

dsm:ComputerType computerType

Specifies the computer type.

xsd:boolean computerTypeSupplied

Specifies if the computer type is specified.

xsd:long operatingSystemType

Specifies the operation system type.

xsd:string scalabilityServerName

Specifies the name of the scalability server of the agent

xsd:boolean scalabilityServerNameSupplied

Specifies if the scalabilityserver name is specified.

dsm:CreateAgentComponents createAgentComponents

Specifies the Agent properties for the computer to be created.

xsd:string description

Specifies the agent description.

xsd:boolean descriptionSupplied

Specifies if the description is specified.

dsm:SDAgentProperties sdAgentProperties

Specifies the SD Agent properties for the computer to be created.

xsd:boolean sdAgentPropertiesSupplied

Specifies if the sdAgentProperties is specified.

xsd:string user

Specifies the user who created the computer.

Default: Blank

xsd:boolean userSupplied

Specifies if the username is specified. The possible values are True or False.

xsd:string phone

Specifies the phone number of user who created the computer.

xsd:boolean phoneSupplied

Specifies if the phone number is specified. The possible values are True or False.

xsd:string location

Specifies the location of user who created the computer.

Default: Blank.

xsd:boolean locationSupplied

Specifies if the location of the user is specified. The possible values are True or False.

xsd:string comment

Specifies the user comments.

xsd:boolean commentSupplied

Specifies if the user comments are available. The possible values are True or False.

xsd:string computerPrimaryMACAddress

Specifies the computer MAC Address.

xsd:boolean computerPrimaryMACAddressSupplied

Specifies if the computer MAC address is specified. The possible values are True or False.

xsd:string osimImage

Specifies the OSIM image.

xsd:string osimBootServerName

Specifies if the boot server name

If the name is not specified, the scalability server name is taken as the OSIM boot server name.

xsd:boolean osimBootServerNameSupplied

Specifies if the boot server name is supplied. The possible values are True or False.

OSImageProperties

Specifies the OS image properties.

xsd:string imageName

Specifies the name of the OS image.

xsd:string typeIndicator

Specifies OS image type.

xsd:string operatingSystem

Specifies the operating environment of OS image.

dsm:DateTime2 registeredAt

Specifies the date and time of the OS image registration with the domain manager.

xsd:string sdPackageName

Specifies the name of the related SD package.

xsd:string sdPackageVersion

Specifies the version of the related SD package.

xsd:string sdPackageComment

Specifies the comment related with the SD package.

xsd:string sdOSCode

Specifies the SD code of the OS associated with the OS image.

xsd:string sdOSLocale

Specifies the locale of the OS image.

xsd:string string osImageFlags

Specifies the image flags associated with the OS image.

SoftwareJobContainerProperties3

The SoftwareJobContainerProperties3 sequence defines the properties for the software job container.

xsd:string softwareJobContainerId

Specifies the UUID identifying the software job container.

SoftwareJobContainerState state

Specifies the current state of the software job container.

xsd:boolean seal

Specifies whether the container is currently sealed.

xsd:long noOfParentSoftwareJobContainers

Specifies the number of software job containers including this container.

xsd:long noOfChildSoftwareJobContainers

Specifies the number of software job containers that this container contains.

xsd:long noOfSoftwareJobs

Specifies the number of software jobs contained in this software job container.

xsd:string name

Specifies the display name of the job container.

xsd:string comment

Specifies the comment for the job container.

xsd:long priority

Specifies the priority of the container that ranges from 1 (highest) to 10 (lowest). If no value is provided, the default value, 5, is used.

SoftwareJobContainerPropertyMask

softwareJobContainerPropertyMask

Specifies the mask for the software job container property. If no value is specified, the default value, linkageBatch and Cascade, is used.

DateTime creationDate

Specifies the date and time when the job container was created.

DateTime changeDate

Specifies the date and time when the job container was last updated.

AttributeMask attribMask

Specifies a series of attributes that describe the software job container.

xsd:long waitCount

Specifies the number of jobs in the container that are not yet activated

xsd:long activeCount

Specifies the number of activated jobs in the container.

xsd:long errorCount

Specifies the number of failed jobs in the container.

xsd:long okCount

Specifies the number of successfully completed jobs in the container.

xsd:long renewState

Specifies the current state of the renewal if the job container is renewed.

xsd:string userMessage

Specifies the custom administrator message.

Remarks

None

SoftwareJobContainerPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a software job container's properties.

- xsd:boolean softwareJobContainerIdRequired
- xsd:boolean stateRequired
- xsd:boolean sealRequired
- xsd:boolean noOfParentSoftwareJobContainersRequired
- xsd:boolean noOfChildSoftwareJobContainersRequired
- xsd:boolean noOfSoftwareJobsRequired
- xsd:boolean nameRequired
- xsd:boolean commentRequired
- xsd:boolean softwareJobContainerPropertyMaskRequired
- xsd:boolean creationDateRequired
- xsd:boolean changeDateRequired
- xsd:boolean attribMaskRequired
- xsd:boolean waitCountRequired
- xsd:boolean activeCountRequired
- xsd:boolean errorCountRequired
- xsd:boolean okCountRequired
- xsd:boolean renewStateRequired

Remarks

For more details, see [SoftwareJobContainerProperties](#) (see page 786)

SoftwareJobContainerPropertiesRequired3

This sequence of elements specifies which properties are required when retrieving a software job container's properties.

xsd:boolean softwareJobContainerIdRequired

Specifies true or false depending on whether the softwareJobContainerId parameter is required.

xsd:boolean stateRequired

Specifies true or false depending on whether the state parameter is required.

xsd:boolean sealRequired

Specifies true or false depending on whether the seal parameter is required.

xsd:boolean noOfParentSoftwareJobContainersRequired

Specifies true or false depending on whether the noOfParentSoftwareJobContainers parameter is required.

xsd:boolean noOfChildSoftwareJobContainersRequired

Specifies true or false depending on whether the noOfChildSoftwareJobContainers parameter is required.

xsd:boolean noOfSoftwareJobsRequired

Specifies true or false depending on whether the noOfSoftwareJobs parameter is required.

xsd:boolean nameRequired

Specifies true or false depending on whether the nameR parameter is required.

xsd:boolean commentRequired

Specifies true or false depending on whether the comment parameter is required.

xsd:boolean priorityRequired

Specifies true or false depending on whether the priority parameter is required.

xsd:boolean softwareJobContainerPropertyMaskRequired

Specifies true or false depending on whether the softwareJobContainerPropertyMask parameter is required.

xsd:boolean creationDateRequired

Specifies true or false depending on whether the creationDate parameter is required.

xsd:boolean changeDateRequired

Specifies true or false depending on whether the changeDate parameter is required.

xsd:boolean attribMaskRequired

Specifies true or false depending on whether the attribMask parameter is required.

xsd:boolean waitCountRequired

Specifies true or false depending on whether the waitCount parameter is required.

xsd:boolean activeCountRequired

Specifies true or false depending on whether the activeCount parameter is required.

xsd:boolean errorCountRequired

Specifies true or false depending on whether the errorCount parameter is required.

xsd:boolean okCountRequired

Specifies true or false depending on whether the okCount parameter is required.

xsd:boolean renewStateRequired

Specifies true or false depending on whether the renewState parameter is required.

xsd:int userMessageRequired

Checks whether or not the custom administrator message is required.

Remarks

None

SoftwareJobContainerPropertyMask

This sequence of elements specifies a number of properties related to the creation and setup of a software job container and the rules that should be applied to the jobs within the container.

xsd:boolean linkageNone

The Jobs in the container will be run independently from one another

xsd:boolean linkageSequence

The jobs in the container are only activated, when the previous job in the container has completed successfully for all of its targets

xsd:boolean linkageBatch

For each target, all jobs in the container will be run as a single piece of work. If a job fails for any target, all remaining jobs in the container for that target will be aborted.

xsd:boolean rollback

xsd:boolean cascade

This value can only be set to true if linkageNone is set to false. If cascading is used, any procedure that is to execute as part of the job container, is for its own execution dependent on the successful execution of all procedures in its Procedure Dependencies folder

xsd:boolean buildOnly

xsd:boolean renewed

Remarks

None

SoftwareJobOperationMask

A series of flags that indicate options for the operation of a software job.

xsd:boolean offline

Execute procedure offline from the Scalability Server

xsd:boolean prompt

Display a message on the target computer before the job starts

xsd:boolean calendarControl

Control the execution by the job execution calendar defined for this target agent.

xsd:boolean shield

Prevent user from being logged-on while the job executes.

xsd:boolean noPromptCancel

Allows user to cancel a "prompt user" job

xsd:boolean rerun

Run the activate/configure job everytime the software policy is Evaluated

xsd:boolean execPromptTimeout

Job will execute if prompt timeout expires (not set - job is postponed)

xsd:boolean resolveQueryGroups

Resolve the target groups before the job is set up

xsd:boolean autoDeliverItem

Store the package in the Scalability Server Library

xsd:boolean defaultItemProc

This is the default procedure for this package with the task defined for this procedure.

xsd:boolean renewFailedOnly

Only failed jobs can be renewed.

xsd:boolean runAtShutdown

To Run while the system is allowed to shutdown

Remarks

More than one flag may be set to true at any one time.

SoftwareJobOperationMask2

A series of flags that indicate options for the operation of a software job.

xsd:boolean offline

Execute procedure offline from the Scalability Server

xsd:boolean prompt

Display a message on the target computer before the job starts

xsd:boolean calendarControl

Control the execution by the job execution calendar defined for this target agent.

xsd:boolean shield

Prevent user from being logged-on while the job executes

xsd:boolean noPromptCancel

Allows user to cancel a "prompt user" job

xsd:boolean rerun

Run the activate/configure job everytime the software policy is Evaluated

xsd:boolean execPromptTimeout

Job will execute if prompt timeout expires (not set - job is postponed)

xsd:boolean resolveQueryGroups

Resolve the target groups before the job is set up.

xsd:boolean autoDeliverItem

Store the package in the Scalability Server Library

xsd:boolean defaultItemProc

This is the default procedure for this package with the task defined for this procedure.

xsd:boolean renewFailedOnly

Only failed jobs can be renewed.

xsd:boolean runAtShutdown

To run while the system is allowed to shutdown.

xsd:boolean reInstall

To remove existing installation records and rerun the install procedure.

Remarks

More than one flag may be set to true at any one time.

SoftwareJobOrderProperties

This sequence of elements describes the properties of a software job order.

DateTime executionDate

The date and time at which the job will be activated. If not supplied, the default value is the current date and time of the manager computer.

xsd:boolean executionDateSupplied**xsd:long timeOutHours**

The time in hours after which time, if the job has been unable to complete, the job will cease attempts to complete. If not set, the default is 48.

xsd:boolean timeOutHoursSupplied**SoftwareJobStartupTime startWhen**

Properties that indicate when the software job should be started. If not set, the default value is EXACT_LOCAL_TIME.

xsd:boolean startWhenSupplied

xsd:string jobName (")

The name of the software job with which this order is associated.

xsd:boolean jobNameSupplied (")

SoftwareJobBootMask bootBeforeMask

Indicates the reboot options requested for the software job before the job is executed. If not set, the default value is noReboot.

xsd:boolean bootBeforeMaskSupplied

SoftwareJobBootMask bootAfterMask

This element indicates the reboot options requested for the software job after the job is executed. If not set, the default value is noReboot.

xsd:boolean bootAfterMaskSupplied

SoftwareJobOperationMask operationMask

Properties for the operation of the software job. If not set, the default value is calendarControl

xsd:boolean operationMaskSupplied

xsd:string userParameters

Contains user-defined parameters.

xsd:boolean userParametersSupplied

xsd:long orderNumber

The order number of the job within its job container.

xsd:boolean orderNumberSupplied

xsd:long promptTimeoutHours

This indicates the number of hours for which a user prompt should be displayed before forced execution of the job. The default value is 1 day. Set this parameter to a value that does not exceed the timeout value for the job. If not set, the default value is 48.

xsd:boolean promptTimeoutHoursSupplied

xsd:string deliveryCalendar

The name of a Unicenter delivery calendar to be used for scheduling the job. The calendar works as ordinary Unicenter calendars, to control the days, dates and times when delivery is permitted. The time cells not set in the Unicenter Calendar define the exclusion intervals.

If you try to define a calendar that is unknown to Unicenter, the job is not created.

xsd:boolean deliveryCalendarSupplied

DateTime deliveryDateTime

The date and time at which the job was delivered. If not set, the default value is the current date and time of the manager computer.

xsd:boolean deliveryDateTimeSupplied

Remarks

None

SoftwareJobOrderProperties3

The SoftwareJobOrderProperties3 sequence of elements describes the properties of a software job order.

DateTime executionDate

Specifies the date and time to activate the job. If you do not provide any value, the default value is the current date and time of the manager computer.

xsd:boolean executionDateSupplied

Specifies true or false depending on whether the executionDate parameter is supplied.

xsd:long timeOutHours

Specifies the time (in hours) after which, if the job is unable to complete, the job will cease attempts to complete. If you do not provide any value, the default value is set to 48 hours.

xsd:boolean timeOutHoursSupplied

Specifies true or false depending on whether the timeOutHours parameter is supplied.

SoftwareJobStartupTime startWhen

Specifies the properties that specify when to start the software job. If you do not provide any value, the default value is EXACT_LOCAL_TIME.

xsd:boolean startWhenSupplied

Specifies true or false depending on whether the startWhen parameter is supplied.

xsd:string jobName (")

Specifies the name of the software job with which this order is associated. If you do not provide any value, SD creates a default name.

xsd:boolean jobNameSupplied (")

Specifies true or false depending on whether the jobName parameter is supplied.

SoftwareJobBootMask3 bootBeforeMask

Specifies the restart options requested for the software job before the job is executed. If you do not provide any value, the default value is noReboot.

xsd:boolean bootBeforeMaskSupplied

Specifies true or false depending on whether the bootBeforeMask parameter is supplied.

SoftwareJobBootMask3 bootAfterMask

Specifies the restart options requested for the software job after the job is executed. If you do not provide any value, the default value is noReboot.

xsd:boolean bootAfterMaskSupplied

Specifies true or false depending on whether the bootAfterMask parameter is supplied.

SoftwareJobOperationMask operationMask

Specifies the properties for the operation of the software job. If you do not provide any value, the default value is calendarControl.

xsd:boolean operationMaskSupplied

Specifies true or false depending on whether the operationMask parameter is supplied.

xsd:string userParameters

Specifies the user-defined parameters.

xsd:boolean userParametersSupplied

Specifies true or false depending on whether the userParameters parameter is supplied.

xsd:long orderNumber

Specifies the order number of a software job within its job container if the software job is part of that job container.

xsd:boolean orderNumberSupplied

Specifies true or false depending on whether the orderNumber parameter is supplied.

xsd:long promptTimeoutHours

Specifies the number of hours for which you want to display a user prompt before forced execution of the job starts. If you do not provide any value, the default value is 24 hours.

xsd:boolean promptTimeoutHoursSupplied

Specifies true or false depending on whether the promptTimeoutHours parameter is supplied.

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to use for scheduling the job. The calendar controls the days, dates, and times when delivery is permitted. The time cells that are not set in the calendar define the exclusion intervals. If the name of the calendar is unknown to SD, the job is not created.

xsd:boolean deliveryCalendarSupplied

Specifies true or false depending on whether the deliveryCalendar parameter is supplied.

DateTime deliveryDateTime

Specifies the date and time to deliver the job. If you do not provide any value, the default value is the current date and time of the manager computer.

xsd:boolean deliveryDateTimeSupplied

Specifies true or false depending on whether the deliveryDateTime parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is supplied.

Remarks

None

SoftwarePackageGroupProperties

xsd:string softwarePackageGroupId

The UUID identifying the software package group.

xsd:long noOfParentSoftwarePackageGroups

The number of parent software package groups to which this group is linked.

xsd:long noOfChildSoftwarePackageGroups

The number of software package groups linked to this group.

xsd:long noOfSoftwarePackages

The number of software packages contained in the software package group.

xsd:string name

The display name of the software package group.

xsd:string comment

A comment for the software package group.

DateTime creationDate

The date and time at which the software package group was created.

DateTime changeDate

The date and time at which the software package group was last updated.

AttributeMask attributeMask

A series of attributes that describe the software package group.

SoftwarePackageGroupProperty softwarePackageGroupProperty

This property indicates the catalog status of the software package group.

Remarks

None

SoftwarePackageProcedureBootMask (bootAfterMask)

A series of flags that indicate the reboot options for a software package procedure.

xsd:boolean noReboot

Do not reboot the computer after the job has ended.

xsd:boolean reboot

Reboot the computer after the job has ended.

xsd:boolean logoff

If logoff is set to true the user agent Logs off the user and the machine agent logs off all users before the job starts.

xsd:boolean rebootAfterLast

Reboots the computer when last job in the container has ended.

xsd:boolean logoffAfterLast

If logoffAfterLast is set to true the user agent logs off the user and the machine agent logs off all users when the last job in the container has ended.

Remarks

Only one flag may be set to true at any one time.

SoftwarePackageProcedureBootMask2 (bootAfterMask)

TheSoftwarePackageProcedureBootMask2 (bootAfterMask) sequence includes a series of flags that indicate the restart options for a software package procedure. This sequence also includes the functionality that helps you shut down a target computer after completion of the jobs.

xsd:boolean noReboot

Specifies not to restart the computer after completion of the job.

xsd:boolean reboot

Restarts the computer after completion of the job.

xsd:boolean logoff

(If set to true) Specifies that the user agent Logs off the user and the machine agent logs off all the users before the job starts.

xsd:boolean rebootAfterLast

Restarts the computer after completion of the last job in the container.

xsd:boolean logoffAfterLast

(If set to true) Specifies that the user agent logs off the user and the machine agent logs off all the users when the last job in the container completes.

xsd:boolean ShutdownAfterLast

(If set to true) Specifies that the target computer shuts down after completion of the last SD job.

Remarks

Only one flag may be set to true at a time.

SoftwarePackageProcedureBootMask

A series of flags that indicate the reboot options for a software package procedure.

xsd:boolean noReboot

Do not reboot the computer before starting the job

xsd:boolean reboot

Reboot the computer before starting the job.

xsd:boolean logoff

If logoff is set to true the user agent logs off the user and the machine agent logs off all users before the job starts.

xsd:boolean rebootAfterLast

Not applicable.

xsd:boolean logoffAfterLast

Not applicable.

Remarks

Only one flag may be set to true at any one time.

SoftwarePackageProcedureBootMask2 (bootBeforeMask)

The SoftwarePackageProcedureBootMask2 (bootBeforeMask) sequence includes a series of flags that indicate the restart options for a software package procedure. This sequence also includes the functionality that helps you shut down a target computer after completion of the jobs.

Parameters**xsd:boolean noReboot**

Specifies not to restart the computer before starting the job.

xsd:boolean reboot

Restarts the computer before starting the job.

xsd:boolean logoff

(If set to true) Specifies that the user agent logs off the user and the machine agent logs off all the users before the job starts.

xsd:boolean rebootAfterLast

Not applicable.

xsd:boolean logoffAfterLast

Not applicable.

xsd:boolean ShutdownAfterLast

Not applicable.

Remarks

Only one flag may be set to true at a time.

FindSoftwarePackageProcedure

The FindSoftwarePackageProcedure method finds a software package procedure given its name and task type, and given the name and version of the software package with which it is associated.

Parameters

xsd:string sessionId (")

The session identifier.

xsd:string softwarePackageName (")

The name of the software package to which the requested procedure should belong.

xsd:string softwarePackageVersion (")

The version of the software package to which the requested procedure should belong.

xsd:string softwarePackageProcedureName (")

The name of the software package procedure to find.

SoftwarePackageProcedureTask softwarePackageProcedureTask

The type of the software package procedure to find.

Return Values

xsd:string FindSoftwarePackageProcedureReturn

If a software package procedure is found its UUID is returned.

Remarks

The names and versions must be specified exactly. This method does not allow wildcards. The software package and procedure names are case insensitive.

More information:

[SoftwarePackageProcedureTask](#) (see page 97)

SoftwarePackageProcedurePropertiesRequired

This sequence of elements specifies which procedure properties are required when retrieving a software package procedure's properties.

- xsd:boolean softwarePackageProcedureIdRequired
- xsd:boolean softwarePackageProcedureNameRequired
- xsd:boolean softwarePackageProcedureTaskRequired
- xsd:boolean softwarePackageProcedureTypeRequired
- xsd:boolean commentRequired
- xsd:boolean userSelectableRequired
- xsd:boolean externalProcedureRequired
- xsd:boolean filedByRequired
- xsd:boolean operatingSystemTypeRequired
- xsd:boolean subPathRequired
- xsd:boolean procToRunRequired
- xsd:boolean parametersRequired
- xsd:boolean procedureRelationRequired
- xsd:boolean activatedByRequired
- xsd:boolean activatedDateTimeRequired
- xsd:boolean softwarePackageIdRequired
- xsd:boolean softwarePackageNameRequired
- xsd:boolean softwarePackageVersionRequired
- xsd:boolean softwarePackageTypeRequired
- xsd:boolean softwarePackageVolumIdRequired
- xsd:boolean softwarePackageVolumeNameRequired
- xsd:boolean internalMaskRequired
- xsd:boolean operationMaskRequired
- xsd:boolean bootBeforeMaskRequired
- xsd:boolean bootAfterMaskRequired
- xsd:boolean dependancyQueryIdRequired
- xsd:boolean managementTypeMaskRequired

Remarks

For more details, see [SoftwarePackageProcedureProperties](#) (see page 843).

SoftwarePackageProcedureTypeMask

This mask is a sequence of variables that can be marked to indicate which types of software package procedure is to be applied to the current request.

xsd:boolean all

All types of procedures.

xsd:boolean userSelectable

Procedure that can be used from the CA ITCM catalog.

xsd:boolean embedded

Procedure included in the original package (not added).

xsd:boolean added

Procedure added to a sealed software.

xsd:boolean converted

Added procedure with already registered or external files.

xsd:boolean detected

Virtual procedure for detected software.

xsd:boolean install

Procedure used to install software.

xsd:boolean uninstall

Procedure used to uninstall software.

xsd:boolean configure

Procedure used to configure software.

xsd:boolean activate

Procedures used to activate software.

SoftwarePackagePropertiesRequired

This sequence of elements specifies which software package properties are required when retrieving a software package's properties.

- xsd:boolean softwarePackageIdRequired
- xsd:boolean softwarePackageNameRequired
- xsd:boolean softwarePackageVersionRequired
- xsd:boolean basedOnSoftwarePackageIdRequired
- xsd:boolean basedOnSoftwarePackageNameRequired
- xsd:boolean basedOnSoftwarePackageVersionRequired
- xsd:boolean supplierRequired
- xsd:boolean commentRequired
- xsd:boolean sourcePathRequired
- xsd:boolean softwarePackageTypeRequired
- xsd:boolean userSelectableRequired
- xsd:boolean softwarePackageDataSourceRequired
- xsd:boolean softwarePackageFiledByRequired
- xsd:boolean registrationDateTimeRequired
- xsd:boolean changeDateTimeRequired
- xsd:boolean packageTypeRequired
- xsd:boolean packetSizeRequired
- xsd:boolean buildOnSizeRequired
- xsd:boolean archivePathRequired
- xsd:boolean archiveCommentRequired
- xsd:boolean archivedByRequired
- xsd:boolean archiveFlagRequired
- xsd:boolean archDateTimeRequired
- xsd:boolean flagsRequired

Remarks

For more details, see [SoftwarePackageProperties](#) (see page 847).

SoftwarePackageVolumeElementPropertiesRequired

This sequence of elements specifies which volume element properties are required when retrieving a software volume element's properties.

- `xsd:boolean volumeElementTypeRequired`
- `xsd:boolean volumeElementNameRequired`
- `xsd:boolean volumeElementSubPathRequired`
- `xsd:boolean volumeElementSizeRequired`
- `xsd:boolean volumeElementAttributesRequired`
- `xsd:boolean volumeElementChangeDateTimeRequired`

Remarks

For more details, see [SoftwarePackageVolumeElementProperties](#) (see page 850).

SoftwarePackageVolumeProperties

`xsd:string volumeId`

The UUID identifying the software package volume

`xsd:string volumeIdFile`

The unique name of a file that identifies the volume.

`xsd:string volumeName`

The name of the software package volume.

`xsd:long volumeNumber`

The zero-based volume number.

`SourceMedium volumeType`

The type of media that hosts the volume.

`xsd:string volumeSubPath`

The path of the volume, relative to the path of the software package to which the volume belongs.

`xsd:string softwarePackageld`

The UUID identifying the software package to which this volume belongs.

Remarks

None

SoftwarePackageVolumePropertiesRequired

This sequence of elements specifies which volume properties are required when retrieving a software package volume's properties.

- `xsd:boolean volumeIdRequired`
- `xsd:boolean volumeIdFileRequired`
- `xsd:boolean volumeNameRequired`
- `xsd:boolean volumeNumberRequired`
- `xsd:boolean volumeTypeRequired`
- `xsd:boolean volumeSubPathRequired`
- `xsd:boolean softwarePackageIdRequired`

Remarks

For more details, see [SoftwarePackageVolumeProperties](#) (see page 836)

SoftwarePackageProcedureDependencyProperties

This sequence defines the properties for the software package procedure dependencies.

xsd:string softwarePackageProcedureDependencyId

The UUID identifying the procedure dependency.

xsd:string softwarePackageProcedureId

The UUID identifying the software package procedure with which the dependency is associated.

xsd:string dependantSoftwarePackageProcedureId

The UUID identifying the software package procedure on which the dependency's associated procedure is dependent.

xsd:string softwarePackageName

The name of the software package associated with this procedure dependency.

xsd:string softwarePackageVersion

The version of the software package associated with this procedure dependency.

xsd:string softwarePackageProcedureName

The name of the software package procedure with which this dependency is associated.

SoftwarePackageProcedureFiledBy filedBy

The entity by which the software package procedure with which this dependency is associated, was filed

xsd:long orderNumber

A number which denotes the order in which the procedure should be executed within its group

Remarks

None

SoftwarePackageProcedureDependencyPropertiesRequired

This sequence of elements specifies which dependency properties are required when retrieving a software package procedure dependency's properties.

- **xsd:boolean softwarePackageProcedureDependencyIdRequired**
- **xsd:boolean softwarePackageProcedureIdRequired**
- **xsd:boolean dependantSoftwarePackageProcedureIdRequired**
- **xsd:boolean softwarePackageNameRequired**
- **xsd:boolean softwarePackageVersionRequired**
- **xsd:boolean softwarePackageProcedureNameRequired**
- **xsd:boolean filedByRequired**
- **xsd:boolean orderNumberRequired**

Remarks

For more details, see [SoftwarePackageProcedureDependencyProperties](#) (see page 837).

SoftwarePackageProcedureGroupProperties

This element defines the properties for the software package procedure group.

xsd:string softwarePackageProcedureGroupId

The UUID identifying the software package procedure group.

xsd:long noOfParentSoftwarePackageProcedureGroups

The number of parent software package procedure groups to which this group is linked.

xsd:long noOfSoftwarePackageProcedures

The number of software package procedures linked to the procedure group.

xsd:string name

The display name of the software package procedure group.

xsd:string comment

A comment for the software package procedure group.

DateTime creationDate

The date and time at which the software package procedure group was created.

DateTime changeDate

The date and time at which the software package procedure group was last updated.

AttributeMask attributeMask

A series of attributes that describe the software package procedure group

Remarks

None

SoftwarePackageProcedureGroupPropertiesRequired

This sequence of elements specifies which software package procedure group properties are required when retrieving a software package procedure group's properties.

- `xsd:boolean softwarePackageProcedureGroupIdRequired`
- `xsd:boolean noOfParentSoftwarePackageProcedureGroupsRequired`
- `xsd:boolean noOfSoftwarePackageProceduresRequired`
- `xsd:boolean nameRequired`
- `xsd:boolean commentRequired`
- `xsd:boolean creationDateRequired`
- `xsd:boolean changeDateRequired`
- `xsd:boolean attribMaskRequired`

Remarks

For more details, see [SoftwarePackageProcedureGroupProperties](#) (see page 838).

SoftwarePackageProcedureManagementTypeMask

This sequence of elements specifies whether a software package can be run on a 'software Managed'

xsd:boolean sd

Execute by the ordinary sd agent.

xsd:boolean sm

Execute by the agent that handles packages created by CA Software Management Packager.

Remarks

- By default, sd is set to true and sm is set to false
- Both Boolean flags may be set at once to allow job execution on both managed and non-managed targets.

SoftwarePackageProcedureOperationMask

The SoftwarePackageProcedureOperationMask sequence includes a series of flags that indicate options for the operation of a software package procedure.

xsd:boolean offline

Executes the procedure in the off-line mode from the scalability server

xsd:boolean prompt

Displays a message on the target computer before the job starts.

xsd:boolean calendarControl

Controls the execution by the job execution calendar defined for this target agent.

xsd:boolean shield

Prevents the user from being logged-on while the job executes.

xsd:boolean lock**xsd:boolean noPromptCancel**

Enables the user to cancel a *prompt user* job.

xsd:boolean background**xsd:boolean rerun**

Runs the Activate or Configure job every time the software policy is evaluated.

xsd:boolean permActArea**xsd:boolean execPromptTimeout**

Runs the job if the prompt timeout expires. If the value is not specified, the job is postponed.

xsd:boolean resolveQueryGroups

Resolves the target groups before the job is set up.

xsd:boolean autoDeliverItem

Stores the package in the scalability server library.

xsd:boolean defaultItemProc

Specifies that this is the default procedure for this package with the task defined for the procedure.

xsd:boolean renewFailedOnly

Renews only the failed jobs.

Remarks

More than one flag may be set to true at any one time.

SoftwarePackageProcedureOperationMask2

The SoftwarePackageProcedureOperationMask2 sequence includes a series of flags that indicate options for the operation of a software package procedure.

xsd:boolean offline

Executes the procedure in the off-line mode from the scalability server.

xsd:boolean prompt

Displays a message on the target computer before the job starts.

xsd:boolean calendarControl

Controls the execution by the job execution calendar defined for this target agent.

xsd:boolean shield

Prevents the user from being logged-on while the job executes.

xsd:boolean lock

xsd:boolean noPromptCancel

Enables the user to cancel a *prompt user* job.

xsd:boolean background

xsd:boolean rerun

Runs the Activate or Configure job every time the software policy is evaluated.

xsd:boolean permActArea

xsd:boolean execPromptTimeout

Runs the job if the prompt timeout expires. If the value is not specified, the job is postponed.

xsd:boolean resolveQueryGroups

Resolves the target groups before the job is set up.

xsd:boolean autoDeliverItem

Stores the package in the scalability server library.

xsd:boolean defaultItemProc

Specifies that this is the default procedure for this package with the task defined for the procedure.

xsd:boolean renewFailedOnly

Renews only the failed jobs.

xsd:boolean useDefaultJobMessage

(If enabled) Specifies that the software package procedure comment is used as the default user message for jobs created using this procedure.

Remarks

More than one flag may be set to true at any one time.

SoftwarePackageProcedureProperties

xsd:string softwarePackageProcedureId

The UUID identifying the software package procedure.

xsd:string softwarePackageProcedureName

The name of the software package procedure.

SoftwarePackageProcedureTask softwarePackageProcedureTask

The task type of the software package procedure.

SoftwarePackageProcedureType softwarePackageProcedureType

The type of the software package procedure.

xsd:string comment

A comment for the software package procedure

xsd:boolean userSelectable

Indicates whether or not the procedure is allowed to be included in the Software Delivery Catalog

xsd:boolean externalProcedure

A flag that indicates whether the procedure is external to SD or not.

SoftwarePackageProcedureFiledBy filedBy

The entity by which the software package procedure was filed.

xsd:long operatingSystemType

xsd:string subPath

The path of the procedure within the volume.

xsd:string procToRun

The procedure command that is to be run.

xsd:string parameters

A string value containing the input parameters for the procedure.

SoftwarePackageProcedureRelation procedureRelation

The path of the procedure, relative to the path of the software package to which the procedure belongs.

SoftwarePackageProcedureFiledBy activatedBy

The entity by which the software package procedure was activated.

DateTime activatedDateTime

The time or date that the procedure was activated.

xsd:string softwarePackageId

The UUID identifying the software package to which the procedure is associated.

xsd:string softwarePackageName

The name of the software package to which the procedure is associated.

xsd:string softwarePackageVersion

The version identifier of the software package to which the procedure is associated.

SoftwarePackageType softwarePackageType

The type of the software package to which the procedure is associated.

xsd:string softwarePackageVolumId

The UUID identifying the volume of the software package to which the procedure is associated.

xsd:string softwarePackageVolumeName

The name of the volume of the software package to which the procedure is associated.

SoftwarePackageProcedureOperationMask internalMask

Internal property for the procedure operation.

SoftwarePackageProcedureOperationMask operationMask

Operation property for the procedure operation.

SoftwarePackageProcedureBootMask bootBeforeMask

This element indicates the reboot options requested for the software package procedure before the procedure is executed.

SoftwarePackageProcedureBootMask bootAfterMask

This element indicates the reboot options requested for the software package procedure after the procedure is executed.

xsd:string dependancyQueryId

If there is a prerequisite defined for the procedure, this is the uuid of the query that checks if the condition is fulfilled.

**SoftwarePackageProcedureManagementTypeMask
managementTypeMask**

This element indicates whether or not the requested software package for the procedure is allowed to run on Software Managed software targets.

Remarks

None

SoftwarePackageProcedureProperties2

This sequence of elements specifies the software package procedure properties. This sequence also includes the functionality that helps you shut down a target computer after completion of the jobs.

xsd:string softwarePackageProcedureId

Specifies the UUID identifying the software package procedure.

xsd:string softwarePackageProcedureName

Specifies the name of the software package procedure.

SoftwarePackageProcedureTask softwarePackageProcedureTask

Specifies the task type of the software package procedure. If no value is specified, the default value, INSTALL, is used.

SoftwarePackageProcedureType softwarePackageProcedureType

Specifies the type of the software package procedure. If no value is specified, the default value, EXEFILE, is used.

xsd:string comment

Specifies a comment for the software package procedure.

xsd:boolean userSelectable

Specifies whether to allow the procedure to be included in the SD Catalog. If no value is provided, the default value, false, is used.

xsd:boolean externalProcedure

Specifies whether the procedure is external to SD. If no value is provided, the default value, false, is used.

SoftwarePackageProcedureFiledBy filedBy

Specifies the entity by which the software package procedure is filed.

xsd:long operatingSystemType

Specifies the type of the operating system. If no value is provided, the default value, SDAPI_OS_ANY, is used.

xsd:string subPath

Specifies the path of the procedure within the volume.

xsd:string procToRun

Runs the procedure command.

xsd:string parameters

Specifies a string value containing the input parameters for the procedure.

SoftwarePackageProcedureRelation procedureRelation

Specifies the path of the procedure relative to the path of the software package to which the procedure belongs.

SoftwarePackageProcedureFiledBy activatedBy

Specifies the entity by which the software package procedure was activated.

DateTime activatedDateTime

Specifies the time or date when the procedure is activated.

xsd:string softwarePackageId

Specifies the UUID identifying the software package to which the procedure is associated.

xsd:string softwarePackageName

Specifies the name of the software package to which the procedure is associated.

xsd:string softwarePackageVersion

Specifies the version identifier of the software package to which the procedure is associated.

SoftwarePackageType softwarePackageType

Specifies the type of the software package to which the procedure is associated.

xsd:string softwarePackageVolumId

Specifies the UUID identifying the volume of the software package to which the procedure is associated.

xsd:string softwarePackageVolumeName

Specifies the name of the volume of the software package to which the procedure is associated.

SoftwarePackageProcedureOperationMask2 operationMask

Specifies the operation property for the procedure operation.

SoftwarePackageProcedureBootMask2 bootBeforeMask

Specifies the restart options requested for the software package procedure before executing the procedure.

SoftwarePackageProcedureBootMask2 bootAfterMask

Specifies the restart options requested for the software package procedure after executing the procedure.

xsd:string dependancyQueryId

Specifies UUID of the query that checks whether the condition is fulfilled, provided a prerequisite is defined for the procedure.

**SoftwarePackageProcedureManagementTypeMask
managementTypeMask**

Specifies whether to allow the requested software package for the procedure to run on the *Software Managed* software targets. If no value is provided, the default value, SD, is used.

Remarks

None

SoftwarePackageProperties

xsd:string softwarePackageId

The UUID identifying the software package

xsd:string softwarePackageName

The name of the software package

xsd:string softwarePackageVersion

The version of the software package.

xsd:string basedOnSoftwarePackageId

A software package may be created or imported based on another software package, this element specifies the UUID of the other software package

xsd:string basedOnSoftwarePackageName

A software package may be created or imported based on another software package, this element specifies the name of the other software package

xsd:string basedOnSoftwarePackageVersion

A software package may be created based on another software package, this element specifies the version of the other software package

xsd:string supplier

The supplier of the software package

xsd:string comment

A comment for the software package

xsd:string sourcePath

SoftwarePackageType softwarePackageType

xsd:boolean userSelectable

SourceMedium softwarePackageDataSource

The source medium for the software package data source.

SoftwarePackageFiledBy softwarePackageFiledBy

This specifies the user who created the software package

DateTime registrationDateTime

This specifies the date and time the software package was created

DateTime changeDateTime

This specifies the date and time the software package was last changed

SoftwarePackagingType packageType

The only type of package you can create is a GENERIC package type.

xsd:long packetSize

xsd:long buildOnSize

xsd:string archivePath

If the software package has been archived this specifies the archive path

xsd:string archiveComment

If the software package has been archived this specifies the archive comment

xsd:string archivedBy

If the software packages has been archived this specifies who performed the archive

ArchivePath archiveFlag

DateTime archDateTime

If the software package has been archived this specifies the date and time the archive took place

SoftwarePackageMask softwarePackageMask

Remarks

None

SoftwarePackageProperties2

xsd:string softwarePackageId

The UUID identifying the software package

xsd:string softwarePackageName

The name of the software package

xsd:string softwarePackageVersion

The version of the software package

xsd:string basedOnSoftwarePackageId

A software package may be created or imported based on another software package. This element specifies the UUID of the other software package.

xsd:string basedOnSoftwarePackageName

A software package may be created or imported based on another software package. This element specifies the name of the other software package.

xsd:string basedOnSoftwarePackageVersion

A software package may be created based on another software package. This element specifies the version of the other software package.

xsd:string supplier

The supplier of the software package

xsd:string comment

A comment for the software package

xsd:string sourcePath

SoftwarePackageType softwarePackageType

xsd:boolean userSelectable**SourceMedium softwarePackageDataSource**

The source medium for the software package data source

SoftwarePackageFiledBy softwarePackageFiledBy

The user who created the software package

DateTime registrationDateTime

The date and time the software package was created

DateTime changeDateTime

The date and time the software package was last changed

SoftwarePackagingType packageType

The only type of package you can create is a GENERIC package type.

xsd:long packetSize

xsd:long buildOnSize

xsd:string archivePath

If the software package has been archived, this element specifies the archive path.

xsd:string archiveComment

If the software package has been archived, this element specifies the archive comment.

xsd:string archivedBy

If the software package has been archived, this element specifies the user who performed the archive.

ArchivePath archiveFlag

DateTime archDateTime

If the software package has been archived, this element specifies the date and time that the archive occurred.

SoftwarePackageMask softwarePackageMask

PackageFormat format

PackageFormat identifies whether the package is regular or virtual.

Remarks

None

SoftwarePackageVolumeElementProperties

This sequence defines the properties for the software package volume elements.

ElementType volumeElementType

The type of the element (file or directory)

xsd:string volumeElementName

The display name of the element.

xsd:string volumeElementSubPath

The relative path within the volume in which the volume element can be found.

xsd:long volumeElementSize

The size (in bytes) of the software element.

VolumeElementAttributes volumeElementAttributes

Attributes that describe the volume element's properties.

DateTime volumeElementChangeDateTime

The date and time that the element's properties were last changed.

Remarks

None

SoftwarePolicyExceptionProperties

This sequence defines the properties for a software policy exception.

xsd:string unitUUID

The UUID identifying the unit that has violated the software policy.

xsd:string unitName

The display name of the unit that has violated the software policy.

UnitType unitType

The unit type of the unit that has violated the software policy.

Remarks

None

SoftwarePolicyProperties

This sequence defines the properties for the software policy.

xsd:string softwarePolicyId

The UUID identifying the software policy.

xsd:SoftwarePolicyState state

The current state of the software policy

xsd:boolean seal

This element indicates whether the policy is currently sealed.

xsd:long noOfParentSoftwarePolicies

The number of software policies of which this policy is a child or member.

xsd:long noOfChildSoftwarePolicies

The number of child or member policies belonging to this software policy.

xsd:long noOfSoftwareJobs

The number of software jobs associated with this software policy.

xsd:string name

The display name of the software policy.

xsd:string comment

A comment for the software policy.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

The type of evaluation to be used for the software policy

SoftwareJobContainerPropertyMask

softwareJobContainerPropertyMask

The UUID identifying the unit group to which the policy is applied

xsd:string unitGroupId

The UUID identifying the unit group to which the policy is applied.

DateTime creationDate

The date and time at which the software policy was created

DateTime changeDate

The date and time at which the software policy was last updated.

AttributeMask attributeMask

A series of attributes that describe the software policy.

xsd:string evaluationCalendar

The name of a previously-created Unicenter calendar that can be assigned to the policy to govern the times at which evaluation will take place.

xsd:long evaluationPeriod

The time in hours between the evaluations of this policy

DateTime nextEvaluationDate

The date and time at which the policy will next be evaluated

DateTime evaluationDate

The date and time at which the policy was last evaluated.

EvaluationTypeMask evaluationTypeMask

Property values that indicate when the policy should be evaluated

Remarks

None

SoftwarePolicyProperties3

The SoftwarePolicyProperties3 sequence defines the properties for the software policy.

xsd:string softwarePolicyId

Specifies the UUID identifying the software policy.

xsd:SoftwarePolicyState state

Specifies the current state of the software policy.

xsd:boolean seal

Specifies whether the policy is currently sealed.

xsd:long noOfParentSoftwarePolicies

Specifies the number of software policies of which this policy is a child or a member.

xsd:long noOfChildSoftwarePolicies

Specifies the number of child or member policies belonging to this software policy.

xsd:long noOfSoftwareJobs

Specifies the number of software jobs associated with this software policy.

xsd:string name

Specifies the display name of the software policy.

xsd:string comment

Specifies the comment for the software policy.

xsd:long priority

Specifies the priority of the software policy.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

Specifies the type of evaluation to be used for the software policy.

SoftwareJobLinkage softwareJobLinkage

Specifies the linkage type of the software policy.

xsd:string unitGroupId

Specifies the UUID identifying the unit group to which the policy is applied.

DateTime creationDate

Specifies the date and time when the software policy was created.

DateTime changeDate

Specifies the date and time when the software policy was last updated.

AttributeMask attributeMask

Specifies a series of attributes that describe the software policy.

xsd:string evaluationCalendar

Specifies the name of a previously created calendar that you can assign to the policy to govern the time for the evaluation to take place.

xsd:long evaluationPeriod

Specifies the time in hours between the evaluations of this policy.

DateTime nextEvaluationDate

Specifies the date and time when you want to evaluate the policy next time.

DateTime evaluationDate

Specifies the date and time when the policy was last evaluated.

AdhocTargetEvaluation adhocTargetEvaluation

Specifies the property values that indicate when to evaluate the policy.

xsd:string userMessage

Specifies the custom administrator message.

Remarks

None

SoftwarePolicyPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a software policy's properties:

- xsd:boolean softwarePolicyIdRequired
- xsd:boolean stateRequired
- xsd:boolean sealRequired
- xsd:boolean noOfParentSoftwarePoliciesRequired
- xsd:boolean noOfChildSoftwarePoliciesRequired
- xsd:boolean noOfSoftwareJobsRequired
- xsd:boolean nameRequired
- xsd:boolean commentRequired
- xsd:boolean SoftwarePolicyEvaluationTypeRequired
- xsd:boolean softwareJobContainerPropertyMaskRequired
- xsd:boolean unitGroupIdRequired
- xsd:boolean creationDateRequired
- xsd:boolean changeDateRequired
- xsd:boolean attribMaskRequired
- xsd:boolean evaluationCalendarRequired
- xsd:boolean evaluationPeriodRequired
- xsd:boolean nextEvaluationDateRequired
- xsd:boolean evaluationDateRequired
- xsd:boolean evaluationTypeMaskRequired

Remarks

For more details, see [SoftwarePolicyProperties](#) (see page 851).

SoftwarePolicyPropertiesRequired2

This sequence of elements specifies which properties are required when retrieving a software policy's properties:

- xsd:boolean softwarePolicyIdRequired
- xsd:boolean stateRequired
- xsd:boolean sealRequired
- xsd:boolean noOfParentSoftwarePoliciesRequired
- xsd:boolean noOfChildSoftwarePoliciesRequired
- xsd:boolean noOfSoftwareJobsRequired
- xsd:boolean nameRequired
- xsd:boolean commentRequired
- xsd:boolean priorityRequired
- xsd:boolean SoftwarePolicyEvaluationTypeRequired
- xsd:boolean softwareJobLinkageRequired
- xsd:boolean unitGroupIdRequired
- xsd:boolean creationDateRequired
- xsd:boolean changeDateRequired
- xsd:boolean attribMaskRequired
- xsd:boolean evaluationCalendarRequired
- xsd:boolean evaluationPeriodRequired
- xsd:boolean nextEvaluationDateRequired
- xsd:boolean evaluationDateRequired
- xsd:boolean adhocTargetEvaluationRequired

Remarks

For more details, see [SoftwarePolicyProperties2](#).

SoftwarePolicyPropertiesRequired3

The SoftwarePolicyPropertiesRequired3 sequence of elements specifies which properties are required when retrieving the properties of a software policy.

xsd:boolean softwarePolicyIdRequired

Specifies true or false depending on whether the softwarePolicyId parameter is required.

xsd:boolean stateRequired

Specifies true or false depending on whether the state parameter is required.

xsd:boolean sealRequired

Specifies true or false depending on whether the seal parameter is required.

xsd:boolean noOfParentSoftwarePoliciesRequired

Specifies true or false depending on whether the noOfParentSoftwarePolicies parameter is required.

xsd:boolean noOfChildSoftwarePoliciesRequired

Specifies true or false depending on whether the noOfChildSoftwarePolicies parameter is required.

xsd:boolean noOfSoftwareJobsRequired

Specifies true or false depending on whether the noOfSoftwareJobs parameter is required.

xsd:boolean nameRequired

Specifies true or false depending on whether the name parameter is required.

xsd:boolean commentRequired

Specifies true or false depending on whether the comment parameter is required.

xsd:boolean priorityRequired

Specifies true or false depending on whether the priority parameter is required.

xsd:boolean SoftwarePolicyEvaluationTypeRequired

Specifies true or false depending on whether the SoftwarePolicyEvaluationType parameter is required.

xsd:boolean softwareJobLinkageRequired

Specifies true or false depending on whether the softwareJobLinkage parameter is required.

xsd:boolean unitGroupIdRequired

Specifies true or false depending on whether the unitGroupId parameter is required.

xsd:boolean creationDateRequired

Specifies true or false depending on whether the creationDate parameter is required.

xsd:boolean changeDateRequired

Specifies true or false depending on whether the changeDate parameter is required.

xsd:boolean attribMaskRequired

Specifies true or false depending on whether the attribMask parameter is required.

xsd:boolean evaluationCalendarRequired

Specifies true or false depending on whether the evaluationCalendar parameter is required.

xsd:boolean evaluationPeriodRequired

Specifies true or false depending on whether the evaluationPeriod parameter is required.

xsd:boolean nextEvaluationDateRequired

Specifies true or false depending on whether the nextEvaluationDate parameter is required.

xsd:boolean evaluationDateRequired

Specifies true or false depending on whether the evaluationDate parameter is required.

xsd:boolean adhocTargetEvaluationRequired

Specifies true or false depending on whether the adhocTargetEvaluation parameter is required.

xsd:int userMessageRequired

Checks whether or not the custom administrator message is required.

Remarks

None

SoftwareJobContainerProperties2

This sequence defines the properties for the software job container.

xsd:string softwareJobContainerId

Specifies the UUID identifying the software job container.

SoftwareJobContainerState state

Specifies the current state of the software job container

xsd:boolean seal

This element indicates whether the container is currently sealed.

xsd:long noOfParentSoftwareJobContainers

Specifies the number of software job containers by which this container is contained.

xsd:long noOfChildSoftwareJobContainers

Specifies the number of software job containers that are contained by this container.

xsd:long noOfSoftwareJobs

Specifies the number of software jobs contained in this software job container.

xsd:string name

Specifies the display name of the job container.

xsd:string comment

A comment for the job container.

xsd:long priority

Priority of the container

SoftwareJobContainerPropertyMask

softwareJobContainerPropertyMask

Mask for the software job container property

DateTime creationDate

Specifies the Date and Time when the job container was created.

DateTime changeDate

Specifies the Date and Time when the job container was last updated.

AttributeMask attribMask

Specifies a series of attributes that describe the software job container.

xsd:long waitCount

Specifies the number of jobs in the container that are not yet activated.

xsd:long activeCount

Specifies the number of jobs in the container that have been activated.

xsd:long errorCount

Specifies the number of jobs in the container that have failed.

xsd:long okCount

Specifies the number of jobs in the container that have successfully completed.

xsd:long renewState

If the job container has been renewed, indicates the current state of that renewal.

SoftwareJobContainerPropertiesRequired2

This sequence of elements specifies which properties are required when retrieving a software job container's properties.

- xsd:boolean softwareJobContainerIdRequired
- xsd:boolean stateRequired
- xsd:boolean sealRequired
- xsd:boolean noOfParentSoftwareJobContainersRequired
- xsd:boolean noOfChildSoftwareJobContainersRequired
- xsd:boolean noOfSoftwareJobsRequired
- xsd:boolean nameRequired
- xsd:boolean commentRequired
- xsd:boolean priorityRequired
- xsd:boolean softwareJobContainerPropertyMaskRequired
- xsd:boolean creationDateRequired
- xsd:boolean changeDateRequired
- xsd:boolean attribMaskRequired
- xsd:boolean waitCountRequired
- xsd:boolean activeCountRequired
- xsd:boolean errorCountRequired
- xsd:boolean okCountRequired
- xsd:boolean renewStateRequired

SoftwarePolicyState

This sequence has the following properties:

- xsd:boolean hasExceptions
- xsd:boolean sealedAndEvaluation

SoftwareJobContainerOrder

xsd:string softwareJobContainerId (m)

UUID of the software job container whose order is to be added.

xsd:long position

This is the position the order will get in the container and the order it is handled at the Domain manager. 0 means last. 1 means first. If the value is greater than the number of orders it is added last in the container. By default, the position is 0 (that is, at the end).

xsd:boolean positionSupplied

SoftwarePolicyOrder

xsd:string softwarePolicyId (m)

Specifies the UUID of the software policy whose order is to be added.

xsd:long position

Specifies the position the order will get in the container and the order it is handled at the Domain manager. 0 means last. 1 means first. If the value is greater than the number of orders it is added last in the container. By default, the position is 0 (that is, at the end).

xsd:boolean positionSupplied

SoftwarePackageProcedureOrder

xsd:string softwarePackageProcedureId (m)

Specifies the UUID of the software package procedure whose order is to be added.

xsd:long position

The position the order will get in the container and the order it is handled at the Domain manager. 0 means last. 1 means first. If the value is greater than the number of orders it is added last in the container. By default, the position is 0 (that is, at the end).

xsd:boolean positionSupplied

SoftwareByName

Members of sequence SoftwareByName.

xsd:string softwarePackageName (m)

Specifies the name of software package.

xsd:string softwarePackageVersion (m)

Specifies software package version of the specified software package.

xsd:string softwarePackageProcedureName (m)

Specifies the name of the job to be removed.

xsd:string softwarePackageInstallProcedureName

Specifies the install procedure name of the job to be removed.

SoftwareJobLinkage

Members of sequence SoftwareJobLinkage:

SoftwareJobLinkageType softwareJobLinkageType

The default value is NOLINKAGE.

xsd:boolean rollback

This parameter is considered only if softwareLinkageType is TRANSACTION. The default value is true.

xsd:boolean cascade

This parameter is considered only if softwareLinkageType is TRANSACTION or SYNCHRONIZED. The default value is false.

SoftwareJobLinkage

This sequence defines the linkage type for the software policy.

SoftwareJobLinkageType softwareJobLinkageType;

- xsd:boolean rollBack;
- xsd:boolean cascade;

SoftwarePackage

Members of sequence SoftwarePackage:

xsd:string softwarePackageName (m)

Specifies the software package name

xsd:string softwarePackageVersion (m)

Mandatory parameter. Specifies the software package version.

SoftwarePackageOrder

xsd:string softwarePackageId (m)

UUID of the software package whose order is to be added.

xsd:long position

This is the position the order will get in the container and the order it is handled at the Domain manager. 0 means last. 1 means first. If the value is greater than the number of orders it is added last in the container. By default, the position is 0 (that is, at the end).

xsd:boolean positionSupplied

SoftwarePackageProcedure

Members of sequence SoftwarePackageProcedure:

xsd:string softwarePackageProcedureName

Specifies the software package procedure name.

xsd:string softwarePackageName (m)

Specifies the software package name.

xsd:string softwarePackageVersion (m)

Specifies the software package version.

SoftwarePolicyProperties2

This sequence defines the properties for the software policy.

xsd:string softwarePolicyId

The UUID identifying the software policy.

xsd:SoftwarePolicyState state

The current state of the software policy.

xsd:boolean seal

This element indicates whether the policy is currently sealed.

xsd:long noOfParentSoftwarePolicies

The number of software policies of which this policy is a child or member.

xsd:long noOfChildSoftwarePolicies

The number of child or member policies belonging to this software policy.

xsd:long noOfSoftwareJobs

The number of software jobs associated with this software policy.

xsd:string name

The display name of the software policy.

xsd:string comment

A comment for the software policy.

xsd:long priority

The priority of the software policy.

SoftwarePolicyEvaluationType softwarePolicyEvaluationType

The type of evaluation to be used for the software policy.

SoftwareJobLinkage softwareJobLinkage

The linkage type of the software policy.

xsd:string unitGroupId

The UUID identifying the unit group to which the policy is applied

DateTime creationDate

The date and time when the software policy was created.

DateTime changeDate

The date and time when the software policy was last updated.

AttributeMask attributeMask

A series of attributes that describe the software policy.

xsd:string evaluationCalendar

The name of a previously-created Unicenter calendar that can be assigned to the policy to govern the times when evaluation will take place.

xsd:long evaluationPeriod

The time in hours between the evaluations of this policy.

DateTime nextEvaluationDate

The date and time when the policy will next be evaluated.

DateTime evaluationDate

The date and time when the policy was last evaluated.

AdhocTargetEvaluation adhocTargetEvaluation

Property values that indicate when the policy should be evaluated.

SSPackageInfo

xsd:string productName;

Product name.

xsd:string packageName;

Package name.

xsd:string packageVersion;

Package version.

xsd:string platform;

Package platform.

xsd:string language;

Language of the package.

SSPackageInfoRequired

xsd:bool bProductNameRequired

Product name flag.

xsd:bool bPackageNameRequired

Package name flag.

xsd:bool bPackageVersionRequired

Package version flag.

xsd:bool bPlatformRequired

Platform flag.

xsd:bool bLanguageRequired

Language flag.

StagingCredentials

xsd:string targetUserName

Name of the user to use for deployment.

xsd:bool bTargetUserSupplied

Target user supplied flag.

xsd:string targetPassword

Password pertaining to the user to be used for deployment.

xsd:bool bTargetPwdSupplied

Target password supplied flag.

dsm:ArrayOfMachineCredentials arrayOfTargetMachineCredentials

Array of MachineCredentials structure which contains credential information for each target machine. Here each target machine is a scalability server.

xsd:bool bTargetMachCredentialsSupplied

Array of target machine credentials supplied flag.

UnitInventoryComponentProperties

xsd:string componentIdString ("")

An xml string that identifies the inventory component.

xsd:string name ("")

The display name of the inventory component.

Remarks

None

UnitInventoryComponentPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a unit inventory component's properties.

- xsd:boolean componentIdStringRequired
- xsd:boolean nameRequired

Remarks

For more details, see [UnitInventoryComponentProperties](#) (see page 868).

UnitInventoryGroupProperties

This sequence of elements specifies which properties are required when retrieving a unit inventory group's properties.

xsd:string groupIdString (^m)

The UUID identifying the inventory group.

xsd:string name (^m)

The display name of the inventory group.

Remarks

None

UnitInventoryGroupPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a unit inventory group's properties.

- **xsd:boolean groupIdStringRequired**
- **xsd:boolean nameRequired**

Remarks

For more details, see [UnitInventoryGroupProperties](#) (see page 869).

UnitInventoryItemProperties

xsd:string itemIdString

An xml string that identifies the inventory item

xsd:string name

The display name of the inventory item.

xsd:string value

The current value of the inventory item.

xsd:string previousValue

The previous value of the inventory item.

StringValue valueType

The type of the value of the inventory item.

DateTime dateCollected

The date and time at which the inventory item value was collected.

Remarks

None

UnitInventoryItemPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a unit inventory item's properties.

- xsd:boolean itemIdStringRequired
- xsd:boolean nameRequired
- xsd:boolean valueRequired
- xsd:boolean previousValueRequired
- xsd:boolean valueTypeRequired
- xsd:boolean dateCollectedRequired

Remarks

For more details, see [UnitInventoryItemProperties](#) (see page 869).

UnitGroupProperties

This sequence of elements specifies the properties of a unit group.

xsd:string groupUUID

The UUID identifying the unit group

xsd:string domainUUID

The UUID identifying the CA ITCM domain for the given group

xsd:string groupLabel

The name of the unit group .

GroupInheritance groupInheritance

This flag indicates the nature of the inheritance of permissions for members of the group

GroupSystem groupSystem

This element indicates whether or not the group is a CA ITCM system group; for example, a value of 1 indicates that the group is user-defined and not a system group. A value of 100 or more indicates that the group is a special predefined system group.

The possible values are:

Unclassified System Group = 0,

User-Defined Group = 1 (Indicates that the group was defined by a CA ITCM user and is not a system group)

Directory Group = 2 (Indicates that this is not a system group, but is a group that maps to a group in an external configured directory.)

- All Computers (\$AllComputer) = 100
- All User Profiles (\$AllUsers) = 101
- All User Accounts (\$AllComputerUsers) = 102
- All Computer User Groups (\$AllComputerUserGroups) = 103
- All Domains (\$AllDomains) = 104
- All Queries (\$AllQueries) = 105
- All Jobs (\$AllJobs) = 106
- All Event Logs (\$AllEventLogs) = 107
- All Servers (\$AllServers) = 108
- All Server Groups (\$AllServerGroups) = 109
- All Domain Groups (\$AllDomainGroups) = 110

GroupFlag groupFlag

This flag indicates whether the group is local or global

xsd:string evaluationUUID

This UUID is not supplied if the group is static, but if the group is dynamically populated, the evaluationUUID is the UUID of the engine that evaluates the query associated with this group.

xsd:long evaluationFrequency

The time (in seconds) between each re-evaluation of the query that governs the group's members.

DateTime lastEvaluationDateTime

The last time at which the members of the group were evaluated.

xsd:string queryUUID

The UUID identifying the query on which basis units are defined as members of this group.

xsd:long memberType

A numeric value that indicates the type of members that the group may contain. This is one of the following:

Asset Group = 1

The group can contain AssetGrps, Computers, Users, and Computer users.

Domain Group = 4

The group can contain Domains and Domain groups

Query Group = 6

The group can contain Queries and Query groups

Server Group= 7

The group can contain Server groups, and Servers

Job Group = 8

The group can contain Jobs and Job Groups

Event Group = 9

The group can contain event log objects

Directory Group = 10

The group can contain directory objects

Values between 100 and 200 represent URC-specific groups

Values between 300 and 400 represent USD-specific groups

Software Group = 300

The group can contain Software definitions

Procedure Group = 301

The group can contain Software procedures

xsd:string groupDescription

Describes the unit group

SecurityGroup securityGroup

Indicates if the unit group is a security group

xsd:string queryLabel

The display name or description of the query associated with this unit group.

xsd:long queryType

The type of the query that defines the members of this unit group. The type indicates on which database tables the query is based.

xsd:string engineLabel

The display name of the engine responsible for evaluating the members of this group.

xsd:string engineHostname

The host name of the engine responsible for evaluating the members of this group.

xsd:string engineDescription

The description of the engine responsible for evaluating the members of this group.

xsd:string creationUserAccount

The name of the user account that created the unit group

DateTime creationdate

The date and time on which the unit group was created.

xsd:string lastUpdatedUserAccount

The name of the user account that last updated the group's properties.

DateTime lastUpdatedDate

The date and time on which the unit group's properties were last updated.

xsd:long groupVersionNumber

The version number of the group's record. A version number is incremented each time a record is updated.

Remarks

For more information, see `propertiesRequired`, `createProperties`, and `setProperties`.

SetUnitGroupProperties2

This sequence of elements specifies which unit group properties are to be set.

- xsd:string groupLabel (m)
- xsd:boolean groupLabelSupplied
- xsd:long evaluationFrequency
- xsd:boolean evaluationFrequencySupplied
- DateTime lastEvaluationDateTime
- xsd:boolean lastEvaluationDateTimeSupplied
- xsd:string queryUUID
- xsd:boolean queryUUIDSupplied
- xsd:string groupDescription
- xsd:boolean groupDescriptionSupplied
- xsd:string engineName
- xsd:boolean engineNameSupplied
- xsd:long inheritPermissions
- xsd:boolean inheritPermissionsSupplied

Remarks

For more details, see [UnitGroupProperties](#) (see page 767).

UnitGroupPropertiesForSoftwarePolicy

Members of sequence UnitGroupPropertiesForSoftwarePolicy:

xsd:string unitGroupName (m)

If *UnitGroupPropertiesForSoftwarePolicy* is supplied to the API the created software policy is attached to the specified unit group. If the unit group does not exist, it is created and the policy attached to it.

xsd:string parentUnitGroupName

If it is supplied, the group is searched/created inside it. If supplied parent group is not found, the system defined unit group “all computers and users” is taken as default parent group. The same system defined group is taken as default parent group if parent unit group is not supplied by user.

xsd:boolean parentUnitGroupNameSupplied (m)

True or false depending on the value of parentUnitGroupName.

xsd:string comment

Accepts comment string for new unit group. This parameter is ignored if the unitGroup already exists.

xsd:boolean commentSupplied (m)

True or false depending on the value of commentForGroup.

xsd:string queryName

Accepts name for query to be attached to the new group. This parameter is ignored if the specified unit group already exists.

xsd:boolean queryNameSupplied (m)

True or false depending on the value of queryNameForGroup.

xsd:string engineName

Specifies the name of the query evaluation engine to be associated with unit group.

xsd:boolean engineNameSupplied (m)

True or false depending on the engineNameForGroup value.

xsd:long evaluationPeriod

Period in which the query is evaluated, specified in minutes. The default value is 1440 minutes. Maximum value allowed is 10080 minutes.

xsd:boolean evaluationPeriodSupplied (m)

True or false depending on the value of evaluationPeriodForGroup.

UnitGroupPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a unit group's properties.

- xsd:boolean groupUUIDRequired
- xsd:boolean domainUUIDRequired
- xsd:boolean groupLabelRequired
- xsd:boolean groupInheritanceRequired
- xsd:boolean groupSystemRequired
- xsd:boolean groupFlagRequired
- xsd:boolean evaluationUUIDRequired
- xsd:boolean evaluationLocationRequired
- xsd:boolean evaluationFrequencyRequired
- xsd:boolean lastEvaluationDateTimeRequired
- xsd:boolean queryUUIDRequired
- xsd:boolean memberTypeRequired
- xsd:boolean groupDescriptionRequired
- xsd:boolean securityGroupRequired
- xsd:boolean queryLabelRequired
- xsd:boolean queryTypeRequired
- xsd:boolean engineLabelRequired
- xsd:boolean engineHostnameRequired
- xsd:boolean engineDescriptionRequired
- xsd:boolean creationUserAccountRequired
- xsd:boolean creationdateRequired
- xsd:boolean lastUpdatedUserAccountRequired
- xsd:boolean lastUpdatedDateRequired
- xsd:boolean groupVersionNumberRequired

Remarks

For more details, see [UnitGroupProperties](#) (see page 870).

UnitProperties

This sequence of elements specifies the properties of a unit (user account, user profile, or computer).

- UnitType unitType
- UserAccountProperties userAccountProperties
- ComputerProperties computerProperties
- UserProfileProperties userProfileProperties

Remarks

Only the one member structure that matches the unit type is to be populated.

UnitPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a unit's properties.

- UserAccountPropertiesRequired userAccountPropertiesRequired
- ComputerPropertiesRequired computerPropertiesRequired
- UserProfilePropertiesRequired userProfilePropertiesRequired

Remarks

For more details, see [UnitProperties](#) (see page 877).

UnitPropertyFilter

This sequence of elements specifies a search filter when searching for or listing units.

UnitProperty unitProperty

The identifier for the unit property on which the search will be based.

This field is mandatory if unitProperty is set to one of the following values:

- UNIT_USERACCOUNT_UUID
- UNIT_USERACCOUNT_NAME
- UNIT_USERACCOUNT_URI
- UNIT_USERACCOUNT_LABEL
- UNIT_USERACCOUNT_DOMAIN_UUID
- UNIT_USERACCOUNT_DOMAIN_LABEL
- UNIT_USERACCOUNT_CREATION_USER
- UNIT_USERACCOUNT_LAST_UPDATED_USER
- UNIT_COMPUTER_UUID
- UNIT_COMPUTER_LABEL
- UNIT_COMPUTER_DOMAIN_UUID
- UNIT_COMPUTER_PRIMARY_NETWORK_ADDRESS
- UNIT_COMPUTER_HOST_UUID
- UNIT_COMPUTER_ASSET_SOURCE_UUID
- UNIT_COMPUTER_HOST_NAME
- UNIT_COMPUTER_SERIAL_NUMBER
- UNIT_COMPUTER_DISC_SERIAL_NUMBER
- UNIT_COMPUTER_PRIMARY_MAC_ADDRESS
- UNIT_COMPUTER_ASSET_TAG
- UNIT_COMPUTER_URI
- UNIT_COMPUTER_SYSTEM_ID
- UNIT_COMPUTER_VENDOR_NAME
- UNIT_COMPUTER_DOMAIN_LABEL
- UNIT_COMPUTER_CREATION_USER
- UNIT_COMPUTER_LAST_UPDATED_USER
- UNIT_USERPROFILE_UUID
- UNIT_USERPROFILE_LABEL
- UNIT_USERPROFILE_CREATION_USER
- UNIT_USERPROFILE_LAST_UPDATED_USER

xsd:long searchValue

If the property to search on is a numeric property, this parameter specifies the value of the search filter to be applied. This field is mandatory if unitProperty is set to one of the following values:

- UNIT_USERACCOUNT_USAGE_LIST
- UNIT_USERACCOUNT_CREATION_DATE
- UNIT_USERACCOUNT_LAST_UPDATED_DATE
- UNIT_USERACCOUNT_VERSION_NUMBER
- UNIT_COMPUTER_CLASSID
- UNIT_COMPUTER_USAGE_LIST
- UNIT_COMPUTER_CREATION_DATE
- UNIT_COMPUTER_LAST_UPDATED_DATE
- UNIT_COMPUTER_VERSION_NUMBER
- UNIT_USERPROFILE_REFERENCE_COUNT
- UNIT_USERPROFILE_USAGE_LIST
- UNIT_USERPROFILE_CREATION_DATE
- UNIT_USERPROFILE_LAST_UPDATED_DATE
- UNIT_USERPROFILE_VERSION_NUMBER

Remarks

None

UnitQueryPropertiesRequired

This sequence of elements specifies which properties are required when retrieving query properties:

- xsd:boolean queryIdRequired
- xsd:boolean queryNameRequired
- xsd:boolean queryTypeRequired
- xsd:boolean queryDomainIdRequired

Remarks

None.

UnitQueryProperties

This sequence defines the properties for the queries.

xsd:string queryId

Specifies the UUID identifying the query.

xsd:string queryName

Specifies the display name of the query.

xsd:string queryDomainId

Specifies the domain UUID identifying the query.

QueryType queryType

Specifies the type of the query.

Remarks

None.

UnitSoftwareJobProperties

This sequence defines the properties for a UnitSoftwareJob.

xsd:string unitSoftwareJobId

The UUID identifying the software job for a specific unit.

xsd:string unitId

The UUID identifying the unit to which the software job has been applied.

xsd:string unitName

The name of the unit to which the software job has been applied.

UnitType unitType

The type of the unit to which the software job has been applied.

xsd:string unitNetAddress

The network address of the unit to which the software job has been applied (where appropriate for the unitType)

xsd:string softwareJobId

The UUID of the software job that has been applied to the given unit.

xsd:string softwareJobName

The name of the software job that has been applied to the given unit.

SoftwareJobState softwareJobState

The current state of the specified software job on the given unit.

xsd:long softwareJobOrderNumber

The order number for the unit's given software job.

xsd:string softwarePackageId

The UUID identifying the software package associated with the unit's given software job.

xsd:string softwarePackageName

The name of the software package associated with the unit's given software job.

xsd:string softwarePackageVersion

The version of the software package associated with the unit's given software job.

SoftwarePackageClass softwarePackageType

The class of the software package associated with the unit's given software job.

xsd:string softwarePackageProcedureId

The UUID identifying the software package procedure associated with the unit's given software job.

xsd:string softwarePackageProcedureName

The name of the software package procedure associated with the unit's given software job.

DateTime activationDateTime

The date and time at which the unit's given software job was activated.

DateTime creationDateTime

The date and time at which the unit's given software job was created.

DateTime completionDateTime

The date and time at which the unit's given software job completed.

SoftwareJobType softwareJobType

The type of the unit's given software job.

xsd:string orderedBy

The user account that started or ordered the unit's software job

UnitSoftwareJobState unitSoftwareJobState

The current state of the unit's given software job

UnitSoftwareJobState unitSoftwareJobRenewState

If the job has been renewed, this element indicates the current state of that renewal.

SoftwareJobPermissionMask unitSoftwareJobPermissionMask

The permissions applicable to the unit's software job.

xsd:string errorMessage

The text of an error message that has been generated by the software job.

xsd:long errorCode

The code of the error (if any) generated by the software job.

xsd:string userDefinedName

A user-defined identifier for this unit software job

xsd:string comment

A comment for the unit's software job

xsd:string unitInstallationSoftwareJobId

The UUID of the installation job associated with this unit software job

xsd:string unitInstallationSoftwarePackageProcedureId

The UUID of the installation procedure associated with this unit software job

xsd:string unitInstallationSoftwarePackageProcedureName

The name of the installation procedure associated with this unit software job

UnitSoftwareJobTaskMask softwareJobTaskMask

The type of the task the unit software job is concerned with, for example, install or configure

xsd:long percentSent

The percentage of the software job that has been sent to its target.

SoftwarePackagingType softwarePackagingType

The type of the software package, for example, generic, PIF, or MS

xsd:string userParams

A string value containing the user-defined parameters for the given unit's software job.

xsd:string unitHostUUID

An agent-generated UUID that identifies the given unit

Remarks

For more information, see [SoftwareJobProperties](#) (see page 768), [SoftwarePackageProperties](#) (see page 847), and [SoftwarePackageProcedureProperties](#) (see page 843).

UnitSoftwareJobProperties2

This sequence defines the properties for a UnitSoftwareJob.

xsd:string unitSoftwareJobId

The UUID identifying the software job for a specific unit

xsd:string unitId

The UUID identifying the unit to which the software job has been applied

xsd:string unitName

The name of the unit to which the software job has been applied

UnitType unitType

The type of the unit to which the software job has been applied

xsd:string unitNetAddress

The network address of the unit to which the software job has been applied (where appropriate for the unitType)

xsd:string softwareJobId

The UUID of the software job that has been applied to the given unit

xsd:string softwareJobName

The name of the software job that has been applied to the given unit

SoftwareJobState softwareJobState

The current state of the specified software job on the given unit

xsd:long softwareJobOrderNumber

The order number for the unit's given software job

xsd:string softwarePackagId

The UUID identifying the software package associated with the unit's given software job

xsd:string softwarePackageName

The name of the software package associated with the unit's given software job

xsd:string softwarePackageVersion

The version of the software package associated with the unit's given software job

SoftwarePackageClass softwarePackageType

An enumerated list of classes to which a software package may belong.

xsd:string softwarePackageProcedureId

The UUID identifying the software package procedure associated with the unit's given software job

xsd:string softwarePackageProcedureName

The name of the software package procedure associated with the unit's given software job

DateTime activationDateTime

The date and time at which the unit's given software job was activated

DateTime creationDateTime

The date and time at which the unit's given software job was created

DateTime completionDateTime

The date and time at which the unit's given software job completed

SoftwareJobType softwareJobType

The type of the unit's given software job

xsd:string orderedBy

The user account that started or ordered the unit's software job

UnitSoftwareJobState unitSoftwareJobState

The current state of the unit's given software job

UnitSoftwareJobState unitSoftwareJobRenewState

If the job has been renewed, this element indicates the current state of that renewal.

SoftwareJobPermissionMask unitSoftwareJobPermissionMask

The permissions applicable to the unit's software job

xsd:string errorMessage

The text of an error message that has been generated by the software job

xsd:long errorCode

The code of the error (if any) generated by the software job

xsd:string userDefinedName

A user-defined identifier for this unit software job

xsd:string comment

A comment for the unit's software job

xsd:string unitInstallationSoftwareJobId

The UUID of the installation job associated with this unit software job

xsd:string unitInstallationSoftwarePackageProcedureId

The UUID of the installation procedure associated with this unit software job

xsd:string unitInstallationSoftwarePackageProcedureName

The name of the installation procedure associated with this unit software job

UnitSoftwareJobTaskMask softwareJobTaskMask

The type of the task the unit software job is concerned with (for example, install or configure)

xsd:long percentSent

The percentage of the software job that has been sent to its target

SoftwarePackagingType softwarePackagingType

The type of the software package (for example, generic, PIF, or MSI)

xsd:string userParams

A string value containing the user-defined parameters for the given unit's software job

xsd:string unitHostUUID

An agent-generated UUID that identifies the given unit

PackageFormat format

PackageFormat identifies whether the package is regular or virtual.

Remarks

For more information, see [SoftwareJobProperties](#) (see page 768), [SoftwarePackageProperties](#) (see page 847), and [SoftwarePackageProcedureProperties](#) (see page 843).

UnitSoftwareJobPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a software job unit's properties. For more details, see `UnitSoftwareJobProperties`.

- `xsd:boolean unitSoftwareJobIdRequired`
- `xsd:boolean unitIdRequired`
- `xsd:boolean unitNameRequired`
- `xsd:boolean unitTypeRequired`
- `xsd:boolean unitNetAddressRequired`
- `xsd:boolean softwareJobIdRequired`
- `xsd:boolean softwareJobNameRequired`
- `xsd:boolean softwareJobStateRequired`
- `xsd:boolean softwareJobOrderNumberRequired`
- `xsd:boolean softwarePackageIdRequired`
- `xsd:boolean softwarePackageNameRequired`
- `xsd:boolean softwarePackageVersionRequired`
- `xsd:boolean softwarePackageTypeRequired`
- `xsd:boolean softwarePackageProcedureIdRequired`
- `xsd:boolean softwarePackageProcedureNameRequired`
- `xsd:boolean activationDateTimeRequired`
- `xsd:boolean creationDateTimeRequired`
- `xsd:boolean completionDateTimeRequired`
- `xsd:boolean softwareJobTypeRequired`
- `xsd:boolean orderedByRequired`
- `xsd:boolean unitSoftwareJobStateRequired`
- `xsd:boolean unitSoftwareJobRenewStateRequired`
- `xsd:boolean unitSoftwareJobPermissionMaskRequired`
- `xsd:boolean errorMessageRequired`
- `xsd:boolean errorCodeRequired`
- `xsd:boolean userDefinedNameRequired`
- `xsd:boolean commentRequired`
- `xsd:boolean unitInstallationSoftwareJobIdRequired`
- `xsd:boolean unitInstallationSoftwarePackageProcedureIdRequired`
- `xsd:boolean unitInstallationSoftwarePackageProcedureNameRequired`
- `xsd:boolean softwareJobTaskRequired`

- xsd:boolean percentSentRequired
- xsd:boolean softwarePackagingTypeRequired
- xsd:boolean userParamsRequired
- xsd:boolean unitHostUUIDRequired

Remarks

None

UnitSoftwareJobTaskMask

This sequence of elements indicates the type of a task being created or retrieved.

- xsd:boolean install
- xsd:boolean uninstall
- xsd:boolean configure
- xsd:boolean activate
- xsd:boolean deliver
- xsd:boolean undeliver

Remarks

Only one of these flags is to be set at any one time.

UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroupByName

This method unlinks the software package procedures from software package procedures group.

Parameters

xsd:string sessionId (m)

Session Id obtained from the login to the web service.

xsd: string softwarePackageProceduresGroupName (m)

Specifies name of an existing software package group from which the software packages need to be unlinked.

ArrayOfSoftwarePackage softwarePackageProcedures (m)

An array list of sequence SoftwarePackage. Each member of type SoftwarePackage contains details specifying the software package to be unlinked.

Return Values

This is a void method that does not expose a return value.

Remarks

None

UsageListMask

This sequence of elements specifies which CA ITCM components are installed and in use on a given computer.

- xsd:boolean engine
- xsd:boolean commonServer
- xsd:boolean sessionMessaging
- xsd:boolean commonAPI
- xsd:boolean commonAgent
- xsd:boolean notificationServer
- xsd:boolean dtsAgent
- xsd:boolean dtsTOS
- xsd:boolean dtsNOS
- xsd:boolean dtsSOS
- xsd:boolean sdAgent
- xsd:boolean sdServer
- xsd:boolean sdManagerNATF
- xsd:boolean sdManagerDM
- xsd:boolean sdManagerTM
- xsd:boolean sdManagerIM
- xsd:boolean sdProxyAgent
- xsd:boolean osimBootServer
- xsd:boolean osimPxeClient
- xsd:boolean amAgent
- xsd:boolean amSwUsageAgent
- xsd:boolean amServer
- xsd:boolean amManager
- xsd:boolean amMeteringServer
- xsd:boolean amPerformanceAgent
- xsd:boolean csmAgent
- xsd:boolean csmServer
- xsd:boolean csmAPI
- xsd:boolean csmACT
- xsd:boolean ccnfAgent
- xsd:boolean rcAgent
- xsd:boolean rcManager

- xsd:boolean babldClient
- xsd:boolean babldServer
- xsd:boolean babldManager

Remarks

None

UserAccountProperties

This sequence of elements represents the properties of a user account.

xsd:string userAccountName

The name identifying the user account

UsageListMask userUsageList

The mask for the usage list

xsd:string userAccountURI

The fully-qualified Uniform Resource Identifier of the user account.

xsd:string userAccountLabel

The display name of the user account.

xsd:string domainLabel

The display name of the domain for the given user account.

xsd:string creationUserAccount

The name of the user account that created the user account's record.

DateTime creationDate

The date and time on which the user account's record was created.

xsd:string lastUpdatedUserAccount

The name of the user account that last updated this user account record's properties.

DateTime lastUpdatedDate

The date and time on which this user account's properties were last updated.

xsd:long userAccountVersionNumber

The version number of the user account's record. A version number is incremented each time a record is updated.

Remarks

None

UserAccountPropertiesRequired

This sequence of elements specifies which properties are required when retrieving a user account's properties.

- xsd:boolean userAccountUUIDRequired
- xsd:boolean userAccountNameRequired
- xsd:boolean userAccountUsageListRequired
- xsd:boolean userAccountURIRequired
- xsd:boolean userAccountLabelRequired
- xsd:boolean userAccountDomainUUIDRequired
- xsd:boolean domainLabelRequired
- xsd:boolean creationUserAccountRequired
- xsd:boolean creationDateRequired
- xsd:boolean lastUpdatedUserAccountRequired
- xsd:boolean lastUpdatedDateRequired
- xsd:boolean userAccountVersionNumberRequired

Remarks

For more details, see [UserAccountProperties](#) (see page 890).

UserAccountPropertyFilter

This sequence of elements specifies a search filter when searching for or listing user accounts.

UserAccountProperty userAccountProperty

The identifier for the user account property on which the search will be based.

xsd:string searchString

If the property to search on is a string property, this parameter specifies the value of the search filter to be applied.

This field is mandatory if userAccountProperty is set to one of the following values

- USERACCOUNT_UUID
- USERACCOUNT_NAME

- USERACCOUNT_USAGE_LIST
- USERACCOUNT_URI
- USERACCOUNT_LABEL
- USERACCOUNT_DOMAIN_UUID
- USERACCOUNT_DOMAIN_LABEL
- USERACCOUNT_CREATION_USER
- USERACCOUNT_LAST_UPDATED_USER

xsd:long searchValue

If the property to search on is a numeric property, this parameter specifies the value of the search filter to be applied. This field is mandatory if userAccountProperty is set to one of the following values

- USERACCOUNT_CREATION_DATE
- USERACCOUNT_LAST_UPDATED_DATE
- USERACCOUNT_VERSION_NUMBER

Remarks

None

UserProfileProperties

This sequence defines the user profile properties.

xsd:string userProfileLabel

Defines the display name of user profiles.

xsd:long userProfileReferenceCount

Defines the count of current records of this user profile.

UsageListMask computerUserUsageList

Defines the mask for the usage list for computer user.

xsd:string userProfileCreationUser

Defines the name of the user account that created the user profile record.

DateTime computerUserCreationDate

Defines the date and time of user profile record creation.

xsd:string userProfileLastUpdatedUser

Defines the name of the user account that last updated the user profile properties.

DateTime computerUserLastUpdatedDate

Defines the date and time of the last update of user profile properties.

xsd:long userProfileVersionNumber

Defines the user profile record version number. The version number increments each time that a record is updated.

xsd:string computerPrimaryNetworkAddress

Defines the IP address (in 0.0.0.0 format) of the computer that is associated with the user profile.

xsd:string computerHostName

Defines the host name of the computer that is associated with the user profile.

xsd:string computerSerialNumber

Defines the serial number of the computer that is associated with the user profile.

xsd:string computerDiscSerialNumber

Defines the serial number of the computer that is associated with the user profile.

xsd:string computerPrimaryMACAddress

Defines the primary MAC address of the computer that is associated with the user profile.

xsd:string computerAssetTag

Defines the serial number of the computer disc that is associated with the user profile.

xsd:long computerClassID

Defines a numeric value that indicates the class of the computer that is associated with the user profile (for example, the Operating System).

Note: For more information about the values a class ID can take, see [ComputerClassID](#) (see page 50).

UsageListMask computerUsageList

Defines the mask for the computer usage list.

xsd:string computerURI

Defines the fully qualified Uniform Resource Identifier (URI) of the computer that is associated with the user profile.

xsd:string computerSystemID

Defines the user-defined identifier for the computer system that is associated with the user profile.

xsd:string computerVendorName

Defines the vendor name of the computer unit that is associated with the user profile.

xsd:string computerDomainLabel

Defines the display name of the CA ITCM domain of the computer that is associated with the user profile.

xsd:string computerCreationUser

Defines the name of the user account that created the computer record that is associated with the user profile.

DateTime computerCreationDate

Defines the date and time on which the computer record associated with the user profile was created.

xsd:string computerLastUpdatedUser

Defines the name of the user account that last updated the properties of the computer record that is associated with the user profile.

DateTime computerLastUpdatedDate

Defines the date and time on which the properties of the computer record that is associated with the user profile were most recently updated.

xsd:long computerVersionNumber

Defines the version number of the record from the computer that is associated with the user profile. The version number increments each time that a record is updated.

UserProfilePropertiesRequired

This sequence of elements specifies which properties are required when retrieving a user profile's properties.

- xsd:boolean userProfileUUIDRequired
- xsd:boolean userAccountUUIDRequired
- xsd:boolean computerUUIDRequired
- xsd:boolean userProfileLabelRequired
- xsd:boolean userProfileReferenceCountRequired
- xsd:boolean userProfileUsageListRequired
- xsd:boolean computerDomainUUIDRequired
- xsd:boolean userProfileCreationUserRequired
- xsd:boolean userProfileCreationDateRequired
- xsd:boolean userProfileLastUpdatedUserRequired
- xsd:boolean userProfileLastUpdatedDateRequired
- xsd:boolean userProfileVersionNumberRequired
- xsd:boolean computerPrimaryNetworkAddressRequired
- xsd:boolean computerHostUUIDRequired
- xsd:boolean computerAssetSourceUUIDRequired
- xsd:boolean computerHostNameRequired
- xsd:boolean computerSerialNumberRequired
- xsd:boolean computerDiscSerialNumberRequired
- xsd:boolean computerPrimaryMACAddressRequired
- xsd:boolean computerAssetTagRequired
- xsd:boolean computerClassIDRequired
- xsd:boolean computerUsageListRequired
- xsd:boolean computerURIRequired
- xsd:boolean computerSystemIDRequired
- xsd:boolean computerVendorNameRequired
- xsd:boolean computerDomainLabelRequired
- xsd:boolean computerCreationUserRequired
- xsd:boolean computerCreationDateRequired
- xsd:boolean computerLastUpdatedUserRequired
- xsd:boolean computerLastUpdatedDateRequired
- xsd:boolean computerVersionNumberRequired

Remarks

For more details, see [UserProfileProperties](#) (see page 892).

UserProfilePropertyFilter

This sequence of elements specifies a search filter when searching for or listing user profiles.

UserProfileProperty userProfileProperty

The identifier for the user profile property on which the search will be based.

xsd:string searchString

If the property to search on is a string property, this parameter specifies the value of the search filter to be applied. This field is mandatory if userProfileProperty is set to one of the following values

- CU_USERPROFILE_UUID
- CU_USERACCOUNT_UUID
- CU_COMPUTER_UUID
- CU_LABEL
- CU_COMPUTER_DOMAIN_UUID
- CU_CREATION_USER
- CU_LAST_UPDATED_USER
- CU_COMPUTER_PRIMARY_NETWORK_ADDRESS
CU_COMPUTER_HOST_UUID
- CU_COMPUTER_ASSET_SOURCE_UUID
- CU_COMPUTER_HOST_NAME
- CU_COMPUTER_SERIAL_NUMBER
CU_COMPUTER_DISC_SERIAL_NUMBER
CU_COMPUTER_PRIMARY_MAC_ADDRESS
- CU_COMPUTER_ASSET_TAG
- CU_COMPUTER_URI

- CU_COMPUTER_SYSTEM_ID
- CU_COMPUTER_VENDOR_NAME
- CU_COMPUTER_DOMAIN_LABEL
- CU_COMPUTER_CREATION_USER
- CU_COMPUTER_LAST_UPDATED_USER

xsd:long searchValue

If the property to search on is a numeric property, this parameter specifies the value of the search filter to be applied. This field is mandatory if userProfileProperty is set to one of the following values

- CU_REFERENCE_COUNT
- CU_USAGE_LIST
- CU_CREATION_DATE
- CU_LAST_UPDATED_DATE
- CU_VERSION_NUMBER
- CU_COMPUTER_CLASSID
- CU_COMPUTER_USAGE_LIST
- CU_COMPUTER_CREATION_DATE
- CU_COMPUTER_LAST_UPDATED_DATE
- CU_COMPUTER_VERSION_NUMBER

Remarks

None

VolumeElementAttributes

This sequence of elements specifies properties of a volume element.

- `xsd:boolean readOnly`
- `xsd:boolean hidden`
- `xsd:boolean system`
- `xsd:boolean directory`
- `xsd:boolean archive`
- `xsd:boolean normal`
- `xsd:boolean temporary`
- `xsd:boolean atomicWrite`
- `xsd:boolean xactionWrite`
- `xsd:boolean compressed`

Remarks

None

CreateSoftwarePackageProperties3

The CreateSoftwarePackageProperties3 sequence specifies the software package properties that are initialized when a software package is created.

xsd:string softwarePackageName (m)

Specifies the name of the software package.

xsd:string softwarePackageVersion (m)

Specifies the version of the software package.

xsd:string supplier

Specifies the supplier of the package.

xsd:boolean supplierSupplied

Specifies whether the supplier value is specified.

xsd:string comment

Specifies a comment for the software package.

xsd:boolean commentSupplied

Specifies whether the comment is specified.

xsd:string basedOnSoftwarePackageId

Specifies that the new software package is based on the UUID of the given package ID.

xsd:boolean basedOnSoftwarePackageIdSupplied

Specifies whether to create a software package that is based on the UUID of the package.

xsd:boolean enableChecksum

Specifies whether to enable the checksum integrity check.

xsd:boolean enableChecksumSupplied

Specifies whether the enableChecksum value is specified.

xsd:int packageType

Specifies whether a software package is a generic SD package (0) or a Debian wrapper package (9).

xsd:string metadataXML

Specifies the metadata for the debian wrapper package.

Note: If you selected the package type as 9, specify the xsd:string metadataXML, otherwise package creation fails.

SetSoftwarePackageProperties3

The SetSoftwarePackageProperties3 sequence of elements specifies the software package properties that you want to set while editing the package.

xsd:string softwarePackageId (m)

Specifies the UUID of the package whose properties need to be edited.

xsd:string softwarePackageName

Specifies the new name of the package.

xsd:boolean softwarePackageNameSupplied

Specifies whether to update a package name.

xsd:string softwarePackageVersion

Specifies the new version of the package.

xsd:boolean softwarePackageVersionSupplied

Specifies whether to update a package version.

xsd:string supplier

Specifies the new supplier of the package.

xsd:boolean supplierSupplied

Specifies whether to update the supplier.

xsd:string comment

Specifies a new comment for the package.

xsd:boolean commentSupplied

Specifies whether to update a comment.

xsd:string archiveComment

Specifies the archive comment for the software package.

xsd:boolean archiveCommentSupplied

Specifies whether to update an archive comment.

xsd:boolean enableChecksum

Specifies whether to enable the package integrity check.

xsd:boolean enableChecksumSupplied

Specifies whether to enable the enableChecksum parameter.

xsd:int packageType

Specifies whether a software package is a generic SD package (0) or a Debian wrapper package (9).

Note: This parameter is not supported in this release.

xsd:boolean packageTypeSupplied

Specifies whether to enable the packageType parameter.

Note: This parameter is not supported in this release.

xsd:string packagemetadataXML

Specifies new metadata for the Debian wrapper package.

xsd:Boolean metadataSupplied

Specifies if metadata must be updated.

Note: If the package is in a sealed state, you can edit comments, enableChecksum fields but cannot change the package type.

SoftwarePackageProperties3

The SoftwarePackageProperties3 sequence of elements specifies software package properties.

xsd:string softwarePackageId

Specifies the UUID that identifies a software package.

xsd:string softwarePackageName

Specifies the name of the software package.

xsd:string softwarePackageVersion

Specifies the version of the software package.

xsd:string basedOnSoftwarePackageId

Specifies the ID of the software package based on which you want to create the new package.

xsd:string basedOnSoftwarePackageName

Specifies the name of the software package based on which you want to create the new package.

xsd:string basedOnSoftwarePackageVersion

Specifies the version of the software package based on which you want to create the new package.

xsd:string supplier

Specifies the supplier of the software package.

xsd:string comment

Specifies a comment for the software package.

xsd:string sourcePath

Specifies the path where the installation files are registered.

Xsd:int packageState

Specifies the type of the software package. The following table describes the states of software packages:

Possible values	Possible states of software packages
0	Sealed state
1	Unsealed and open state
2	Detected state
3	Archived

xsd:boolean userSelectable

Indicates whether the package is displayed in the Software Delivery Catalog.

Default: False

Xsd:int sourceMedium

Specifies the source for the software package data source. The following table describes the sources of software packages:

Possible values	Possible sources of software packages
0	DISKETTE
1	HARDDISK
2	UNDEFINED
3	TAPE
4	CD

xsd:int packageFiledBy

Specifies the user who has created the software package. The item is filed by a local administrator. The following table describes the users of software packages:

Possible values	Possible users of software packages
0	CENTRAL Specifies an item that is distributed and filed by an enterprise administrator.
1	TD-LOCAL Specifies a software delivery item that is filed by a local administrator.
2	TDSWD Specifies a software item that is detected and filed by the local Software Detector.

Possible values	Possible users of software packages
3	TDSWD-CENTRAL Specifies a software item that is filed at the enterprise manager which in turn, is based on a report from the domain manager about a locally detected software item.
4	TD Specifies a software delivery item that is automatically filed during installation.

DateTime registrationDateTime

Specifies the date and time at which a software package is registered.

DateTime changeDateTime

Specifies the date and time at which the software package was last modified.

Xsd:int packageType

Specifies the type of software package. The following table describes the types of software packages:

Possible values	Types of software packages
0	Generic
1	SXP.
2	PIF
3	PKG
4	MSI
5	PALM
6	WINCE
7	RPM
8	Nokia
9	DebWrap

xsd:long packetSize

Specifies the standard package size of the package in KB.

xsd:long buildOnSize

Specifies the extended package size in KB with added procedures.

xsd:string archivePath

Specifies the archive path of the software package.

xsd:string archiveComment

Specifies the archive comment for the software package.

xsd:string archivedBy

Specifies the user who archived the package.

xsd:int archiveAt

Indicates whether the specified path is archived or not and also specifies the location at the server or enterprise manager. The following table describes whether the specified path is archived or not:

Possible values	Specified path is Archived or not
0	Not archived
1	Archived on the admin client
2	Domain manager
3	Enterprise manager

dsm:DateTime archDateTime

Specifies the archived date and time for the software package.

Xsd:int checksum**Xsd:int packageFormat**

Specifies whether a package is regular (0) or Virtual (1).

Xsd:string packagemetadataXML

Specifies the metadata of the Debian wrapper package.

SoftwarePackagePropertiesRequired3

The `dsml:SoftwarePackagePropertiesRequired3` sequence of elements specifies the software package properties that are required when retrieving a software package property.

- `xsd:boolean softwarePackageIdRequired`
- `xsd:boolean softwarePackageNameRequired`
- `xsd:boolean softwarePackageVersionRequired`
- `xsd:boolean basedOnSoftwarePackageIdRequired`
- `xsd:boolean basedOnSoftwarePackageNameRequired`
- `xsd:boolean basedOnSoftwarePackageVersionRequired`
- `xsd:boolean supplierRequired`
- `xsd:boolean commentRequired`
- `xsd:boolean sourcePathRequired`
- `xsd:boolean packageStateRequired`
- `xsd:boolean userSelectableRequired`
- `xsd:boolean softwarePackageDataSourceRequired`
- `xsd:boolean softwarePackageFiledByRequired`
- `xsd:boolean registrationDateTimeRequired`
- `xsd:boolean changeDateTimeRequired`
- `xsd:boolean packageTypeRequired`
- `xsd:boolean packetSizeRequired`
- `xsd:boolean buildOnSizeRequired`
- `xsd:boolean archivePathRequired`
- `xsd:boolean archiveCommentRequired`
- `xsd:boolean archivedByRequired`
- `xsd:boolean archiveAtRequired`
- `xsd:boolean archDateTimeRequired`
- `xsd:boolean flagsRequired`
- `xsd:boolean metadataRequired`

SoftwarePackageProcedureProperties4

The dsm: SoftwarePackageProcedureProperties4 sequence specifies the software package procedure properties.

xsd:string softwarePackageProcedureId

Specifies the UUID of the software package procedure.

xsd:string softwarePackageProcedureName

Specifies the name of the software package procedure.

SoftwarePackageProcedureTask softwarePackageProcedureTask

Specifies the task type of the software package procedure.

Default: INSTALL

Xsd:int procedureType

Specifies the states of the software package procedure. The following table describes the states of software package procedures:

Possible values	Types of the software package procedures
0	SIP
1	CMDFILE
2	EXEFILE
3	SWDET
4	MSI
5	SXP
6	PKG
7	PIF
8	IPS
9	PALM
10	WINCE
11	RPM
12	NOKIA
13	DebWrap

xsd:string comment

Specifies a comment for the software package procedure.

xsd:boolean userSelectable

Specifies whether to allow the procedure to be included in the SD Catalog.

Default: False

xsd:boolean externalProcedure

Specifies that the procedure is external to SD.

Default: False

SoftwarePackageProcedureFiledBy filedBy

Specifies the entity by which the software package procedure is filed.

xsd:long operatingSystemType

Specifies the type of the operating system.

Default: SDAPI_OS_ANY

xsd:string subPath

Specifies the path of the procedure within the volume.

xsd:string procToRun

Specifies the procedure command.

xsd:string parameters

Specifies the input parameters for the procedure.

SoftwarePackageProcedureRelation procedureRelation

Specifies the path of the procedure relative to the path of the software package to which the procedure belongs.

SoftwarePackageProcedureFiledBy activatedBy

Specifies the entity by which the software package procedure was activated.

DateTime activatedDateTime

Specifies the date and time when the procedure is activated.

xsd:string softwarePackageId

Specifies the UUID of the software package to which the procedure is associated.

xsd:string softwarePackageName

Specifies the name of the software package to which the procedure is associated.

xsd:string softwarePackageVersion

Specifies the version of the software package to which the procedure is associated.

Xsd:int softwarePackageState

Specifies the states of the software package to which the procedure is associated. The following table describes the software package to which the procedure is associated:

Possible values	States
0	Sealed
1	Unsealed
2	Detected
3	Archived

xsd:string softwarePackageVolumeld

Specifies the UUID of the volume of the software package to which the procedure is associated.

xsd:string softwarePackageVolumeName

Specifies the name of the volume of the software package to which the procedure is associated.

SoftwarePackageProcedureOperationMask3 operationMask

Specifies the operation property for the procedure operation.

SoftwarePackageProcedureBootMask2 bootBeforeMask

Specifies the restart option that is requested for the software package procedure before executing the procedure.

SoftwarePackageProcedureBootMask2 bootAfterMask

Specifies the restart options that are requested for the software package procedure after executing the procedure.

xsd:string dependancyQueryId

Specifies UUID of the query that checks whether the condition is fulfilled, provided a prerequisite is defined for the procedure.

**SoftwarePackageProcedureManagementTypeMask
managementTypeMask**

Specifies whether to allow the requested software package for the procedure to run on the managed targets.

Default: SD

SoftwarePackageProcedurePropertiesRequired2

The SoftwarePackageProcedurePropertiesRequired2 sequence of elements specifies the procedure properties that are required when retrieving the software package procedure properties.

- xsd:boolean softwarePackageProcedureIdRequired
- xsd:boolean softwarePackageProcedureNameRequired
- xsd:boolean softwarePackageProcedureTaskRequired
- xsd:boolean softwarePackageProcedureTypeRequired
- xsd:boolean commentRequired
- xsd:boolean userSelectableRequired
- xsd:boolean externalProcedureRequired
- xsd:boolean filedByRequired
- xsd:boolean operatingSystemTypeRequired
- xsd:boolean subPathRequired
- xsd:boolean procToRunRequired
- xsd:boolean parametersRequired
- xsd:boolean procedureRelationRequired
- xsd:boolean activatedByRequired
- xsd:boolean activatedDateTimeRequired
- xsd:boolean softwarePackageIdRequired
- xsd:boolean softwarePackageNameRequired
- xsd:boolean softwarePackageVersionRequired
- xsd:boolean softwarePackageStateRequired
- xsd:boolean softwarePackageVolumIdRequired
- xsd:boolean softwarePackageVolumeNameRequired
- xsd:boolean internalMaskRequired
- xsd:boolean operationMaskRequired
- xsd:boolean bootBeforeMaskRequired
- xsd:boolean bootAfterMaskRequired
- xsd:boolean dependancyQueryIdRequired
- xsd:boolean managementTypeMaskRequired

UnitSoftwareJobProperties3

The UnitSoftwareJobProperties3 sequence defines the properties for a unit software job.

xsd:string unitSoftwareJobId

Specifies the UUID of the software job for a specific unit.

xsd:string unitId

Specifies the UUID of the unit to which the software job is applied.

xsd:string unitName

Specifies the name of the unit to which the software job is applied.

UnitType unitType

Specifies the type of the unit to which the software job is applied.

xsd:string unitNetAddress

Specifies the network address of the unit to which the software job is applied, appropriate for the unitType.

xsd:string softwareJobId

Specifies the UUID of the software job that is applied to the given unit.

xsd:string softwareJobName

Specifies the name of the software job that is applied to the given unit.

SoftwareJobState softwareJobState

Specifies the current state of the specified software job on the given unit.

xsd:long softwareJobOrderNumber

Specifies the order number for the given software job.

xsd:string softwarePackageId

Specifies the UUID that identifies the software package that is associated with the given software job unit.

xsd:string softwarePackageName

Specifies the name of the software package that is associated with the given unit software job.

xsd:string softwarePackageVersion

Specifies the version of the software package that is associated with the given software job unit.

SoftwarePackageClass softwarePackageState

Specifies an enumerated list of classes to which a software package belongs.

xsd:string softwarePackageProcedureId

Specifies the UUID that identifies the software package procedure that is associated with the given software job unit.

xsd:string softwarePackageProcedureName

Specifies the name of the software package procedure that is associated with the given software job unit.

DateTime activationDateTime

Specifies the date and time when the software job is activated.

DateTime creationDateTime

Specifies the date and time when the software job is created.

DateTime completionDateTime

Specifies the date and time when the software job is completed.

SoftwareJobType softwareJobType

Specifies the type of the software job.

xsd:string orderedBy

Specifies the user account that started or ordered the software job unit.

UnitSoftwareJobState unitSoftwareJobState

Specifies the current state of the unit software job.

UnitSoftwareJobState unitSoftwareJobRenewState

Specifies that the job is renewed and also indicates the current state of that renewal.

SoftwareJobPermissionMask unitSoftwareJobPermissionMask

Specifies the permissions applicable to the unit software job.

xsd:string errorMessage

Specifies the text of an error message that the software job has generated.

xsd:long errorCode

Specifies the software job generates code of the error.

xsd:string userDefinedName

Specifies a user-defined identifier for the software job unit.

xsd:string comment

Specifies a comment for the unit software job.

xsd:string unitInstallationSoftwareJobId

Specifies the UUID of the installation job that are associated with the given software job unit.

xsd:string unitInstallationSoftwarePackageProcedureId

Specifies the UUID of the installation procedure that is associated with the given software job unit.

xsd:string unitInstallationSoftwarePackageProcedureName

Specifies the name of the installation procedure that is associated with the given software job unit.

UnitSoftwareJobTaskMask softwareJobTaskMask

Specifies the type of the task to which the unit software job is concerned.

Example: Install or Configure

xsd:long percentSent

Specifies the percentage of the software job that is sent to target.

Xsd:Int packageType

Specifies the type of the software package. The following table describes the types of software packages:

Possible values	Types of software packages
0	Generic
1	SXP
2	PIF
3	PKG
4	MSI
5	PALM
6	WINCE
7	RPM
8	Nokia
9	DebWrap

xsd:string userParams

Specifies a string value that contains the user-defined parameters for the given software job unit.

xsd:string unitHostUUID

Specifies an agent-generated UUID which identifies the given unit.

Xsd:int packageformat

Specifies the whether package is regular (0) or Virtual (1).

WSChallengeResponse

This sequence of elements specifies the string and size of the authentication data along with the session ID returned by the CA ITCM manager.

char *authenticationData

Specifies the authentication data

long size

Specifies the size of the authentication data

Char* SessionID

Specifies a unique session ID to use as an input parameter for all further calls to the web service.

WSAuthenticateResponse

This sequence contains the authentication response received from the CA ITCM manager.

bool bAuthorized

Indicates the authentication status based on the response received from the CA ITCM manager.

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

Item

This sequence contains the item uuid and item type.

xsd:string itemUuid

Specifies the uuid of the item.

xsd:int itemType

Indicates the supported values for item Types are 0 – Computer, 1 – User account, 2 – User profile, 3 – Unit group.

UnitGroupProperties2

The properties of unit group:

xsd:string groupLabel

Specifies the name of the group.

xsd:string groupUuid

Specifies the Uuid of the group.

xsd:string description

Specifies the description of the group.

xsd:string domainUuid

Specifies the Uuid of the domain to which this group belongs to.

xsd:int groupType

Specifies whether the group is System, Local or Global group.

- 0 indicates System group
- 1 indicates Local group
- 2 indicates Global group

xsd:string creationUser

Specifies the Creation User.

dsm:DateTime2 creationdate

Specifies the Creation Date time is ISO format.

xsd:string lastUpdatedUser

Specifies the last modified User.

dsm:DateTime2 lastUpdatedDate

Specifies the Last update Date time in ISO format.

xsd:boolean membersInheritPermissions

Specifies the Boolean value which tells whether the group members inherit the security permissions of the group.

xsd:string queryUUID

Specifies the Uuid of the query linked to the group.

xsd:string queryLabel

Specifies the Name of the query.

xsd:string engineLabel

Specifies the Name of the engine used for evaluating the query.

xsd:string evaluationUUID

Specifies the Uuid of the engine that evaluated the query.

xsd:string engineDescription

Describes of the engine.

xsd:long evaluationFrequency

Specifies the Evaluation frequency in minutes.

dsm:DateTime2 lastEvaluationDateTime

Last evaluated date time.

More Information:

[CreateSoftwareJobOrderPolicyProperties](#) (see page 682)

CreateUnitGroupProperties3

You require the following properties for creating a group:

xsd:string groupLabel

Specifies the name of the group to be created.

xsd:string groupDescription

Specifies the description of the group.

xsd:boolean groupDescriptionSupplied

Specifies the True or False depending on the groupDescription value.

xsd:boolean membersInheritPermissions

Set the value to true if you want the group members to inherit the security permissions of the group.

xsd:boolean membersInheritPermissionsSupplied

Specifies the True or False depending on the membersInheritPermissions value.

xsd:string queryUUID

Specifies the uuid of the query that must be linked to the newly created group. If you do not specify, no query is linked to the group.

xsd:boolean queryUUIDSupplied

Specifies the True or false depending on the value of queryUUID.

xsd:string engineName

Specifies the name of the query evaluation engine to be associated with the unit group.

xsd:boolean engineNameSupplied

Specifies the True or False depending on the engineName value.

xsd:long evaluationPeriod

Specifies the Period, in which the query is evaluated, in minutes.

Default:

1440 minutes.

Limit:

10080 minutes.

xsd:boolean evaluationPeriodSupplied

Specifies the True or False depending on the value of evaluation period.

createAssessmentTemplate

The createAssessmentTemplate method creates an assessment template

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

AssessmentTemplate assessmentTemplate

Specifies the information required to create an assessment template.
The assessment template name must be unique.

Return Values

Returns zero, if the assessment template is successfully created;
otherwise, throws AutomationWebServiceException as a fault.

updateAssessmentTemplate

The updateAssessmentTemplate method updates an assessment template.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string oldAssessmentTemplateName

Specifies the name of the old assessment template.

AssessmentTemplate assessmentTemplate

Specifies the information required to create an assessment template.
The assessment template name must be unique.

Return Values

Returns zero, if the assessment template is successfully updated;
otherwise, throws AutomationWebServiceException as a fault.

deleteAssessmentTemplates

The deleteAssessmentTemplates method deletes a list of assessment templates.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string[] assessmentTemplateName

Specifies the list of the assessment template names you want to delete.

Return Values

Returns zero on successful deletion of all the assessment templates; otherwise, throws AutomationWebServiceExceptionList as a fault.

getAssessmentTemplate

The getAssessmentTemplate method returns the properties of a specified assessment template.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string assessmentTemplateName

Specifies the name of the assessment template you want to retrieve.

Return Values

AssessmentTemplate

Contains the properties of the assessment template on success; otherwise, throws AutomationWebServiceException as a fault.

getAssessmentTemplateList

The `getAssessmentTemplateList` method returns a list of assessment templates names and their last modified dates, based on the specified filter criteria.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

AssessmentTemplateFilter[] assessmentTemplateFilterArray

Specifies the filter condition on some of the assessment template properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

dsm:ASSESSMENTTEMPLATEPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int numberOfTemplatesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

GetTemplateListResponse

Returns a list of assessment templates names and their last modified dates, on success; otherwise, throws `AutomationWebServiceException` as a fault.

More information:

[AssessmentTemplateFilter](#) (see page 960)

[GetTemplateListResponse](#) (see page 960)

[ASSESSMENTTEMPLATEPROPERTY](#) (see page 129)

getAssessmentTemplateListDetailed

The `getAssessmentTemplateListDetailed` method returns a list of assessment templates and their details based on the specified filter criteria.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

AssessmentTemplateFilter[] assessmentTemplateFilterArray

Specifies the filter on some of the assessment template properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to `True` to perform an AND operation. Set the parameter to `False` to perform an OR operation.

enum ASSESSMENTTEMPLATEPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int noOfTemplatesRequired

Specifies the number of objects you want to retrieve from the `startIndex` position.

Return Values

GetAssessmentTemplateListDetailedResponse

Returns the list of Assessment templates, filtered and sorted as specified.

More information:

[AssessmentTemplateFilter](#) (see page 960)

[ASSESSMENTTEMPLATEPROPERTY](#) (see page 129)

[GetAssessmentTemplateListDetailedResponse](#) (see page 961)

createAssessmentTask

The createAssessmentTask method creates an assessment job.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string groupName

Specifies the name of group to be assessed.

AssessmentTask assessmentTask

Specifies the information required to create an assessment task. The assessment task name must be unique.

Return Values

Returns zero, if the assessment template is successfully created; otherwise, throws AutomationWebServiceException as a fault.

executeAssessmentTask

The executeAssessmentTask method runs an assessment job.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string assessmentTaskName

Specifies the name of the assessment task you want to run.

Return Values

Returns zero, if the assessment task is successfully executed; otherwise, throws AutomationWebServiceException as a fault.

getAssessmentStatus

The `getAssessmentStatus` method returns the status of assessment for a particular assessment task.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string assessmentTaskName

Specifies the name of the assessment task.

Return Values

AssessmentStatus

Returns the status of assessment for the assessment task; otherwise, throws `AutomationWebServiceException` as a fault.

More information:

[AssessmentStatus](#) (see page 130)

getAssessmentResult

The `getAssessmentResult` method gets the results of the assessment task.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string assessmentTaskName

Specifies the name of the assessment task, the results of which you want to retrieve.

Return Values

AssessmentResult[] assessmentResult

Returns the assessment results of the executed assessment task on success; otherwise, throws `AutomationWebServiceException` as a fault.

More information:

[AssessmentResult](#) (see page 961)

getAssessmentTask

The getAssessmentTask method returns information about the assessment task.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string assessmentTaskName

Specifies assessment task name for which you need the information.

Return Values

Returns the following sequence as a response on success; otherwise, throws AutomationWebServiceException as a fault.

AssessmentTaskInfo assessmentTaskInfo

Contains information about the assessment task.

More information:

[AssessmentTaskInfo](#) (see page 962)

getAssessmentTaskList

The getAssessmentTaskList method returns information about all assessment tasks, based on the specified filter and sort criteria.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

AssessmentTaskFilter[] assessmentTaskFilterArray

Specifies the filter on some of the assessment task properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

enum ASSESSMENTTASKPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int noOfTemplatesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

GetAssessmentTaskListResponse

Returns the list of assessment tasks based on the mentioned filter and sort properties.

More information:

[AssessmentTaskFilter](#) (see page 962)

[GetAssessmentTaskListResponse](#) (see page 963)

[ASSESSMENTTASKPROPERTY](#) (see page 129)

deleteAssessmentTasks

The deleteAssessmentTasks method deletes a list of assessment tasks.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

xsd:string [] assessmentTaskName

Specifies the list of the assessment task names you want to delete.

Return Values

Returns zero on successful deletion of all the assessment tasks; otherwise, throws AutomationWebServiceExceptionList as a fault.

createOSIMTemplate

The createOSIMTemplate method creates an OSIM Template.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

OsimTemplate OSIMTemplate

Specifies attributes required to create the OSIM template.

Return Values

Returns zero on successful creation of OSIM template; otherwise, throws AutomationWebServiceException as a fault.

updateOSIMTemplate

The updateOSIMTemplate method updates an OSIM template.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Xsd:string oldOsimTemplateName

Specifies the name of the old OSIM template.

OsimTemplate OSIMTemplate

Specifies attributes required to update an OSIM template.

Return Values

Returns zero on successful update of OSIM template; otherwise, throws AutomationWebServiceException as a fault.

deleteOSIMTemplates

The deleteOSIMTemplates method deletes a list of OSIM templates.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string [] OSIMTemplateName

Specifies the list of OSIM template names you want to delete.

Return Values

Returns zero on successful deletion of all templates; otherwise, throws `AutomationWebServiceExceptionList` as a fault.

getOSIMTemplate

The `getOSIMTemplate` method returns the properties of an OSIM template.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string OSIMTemplateName

Specifies OSIM template name for which you want to retrieve the properties.

Return Values

Returns the following sequence on success; otherwise, throws `AutomationWebServiceException` as a fault.

OsimTemplate

Returns OSIM template properties.

getOSIMTemplateList

The `getOSIMTemplateList` method returns a list of OSIM template names and their last modified date, based on the specified filter and sort criteria.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

OsimTemplateFilter[] OSIMTemplateFilterArray

Specifies the filter condition on the OSIM template properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to `True` to perform an AND operation. Set the parameter to `False` to perform an OR operation.

enum OSIMTEMPLATEPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int noOfTemplatesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

GetTemplateListResponse

Contains a list of OSIM template names and their last modified date.

More information:

[GetTemplateListResponse](#) (see page 960)

[OsimTemplateFilter](#) (see page 965)

[OSIMTEMPLATEPROPERTY](#) (see page 131)

getOsimTemplateListDetailed

The getOsimTemplateListDetailed method returns the OSIM templates objects list, based on filter and sort criteria.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

OsimTemplateFilter[] OSIMTemplateFilterArray

Specifies the filter condition on the OSIM template properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

enum OSIMTEMPLATEPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int noOfTemplatesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

GetOsimTemplateListDetailedResponse

Specifies the list of OSIM templates with all fields based on the mentioned filter and sort properties.

More information:

[OsimTemplateFilter](#) (see page 965)

[OSIMTEMPLATEPROPERTY](#) (see page 131)

[GetOsimTemplateListDetailedResponse](#) (see page 965)

createSoftwareDeploymentTemplate

The createSoftwareDeploymentTemplate method creates a software deployment template.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

SoftwareDeploymentTemplate swDeployTemplate

Specifies the properties required to create software deployment template.

Return Values

Returns zero on successful creation of software deployment template; otherwise, throws AutomationWebServiceException as a fault.

More information:

[SoftwareDeploymentTemplate](#) (see page 965)

updateSoftwareDeploymentTemplate

The updateSoftwareDeploymentTemplate method updates a software deployment template.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Xsd:string oldSDTemplateName

Specifies the name of the old software deployment template.

SoftwareDeploymentTemplate swDeployTemplate

Specifies properties you want to update in the software deployment template.

Return Values

Returns zero on successful update of the template; otherwise, throws AutomationWebServiceException as a fault.

More information:

[SoftwareDeploymentTemplate](#) (see page 965)

deleteSoftwareDeploymentTemplates

The deleteSoftwareDeploymentTemplates method deletes a list of software deployment templates.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string [] softwareDeploymentTemplateNames

List of software deployment template names you want to delete.

Return Values

Returns zero on successful deletion of all the software deployment templates; otherwise, throws AutomationWebServiceExceptionList as a fault.

getSoftwareDeploymentTemplate

The getSoftwareDeploymentTemplate method gets the properties of a software deployment template.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string swDeployTemplateName

Specifies the name of the software deployment template for which you want to retrieve the properties.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

SoftwareDeploymentTemplate

Returns software deployment template properties.

More information:

[SoftwareDeploymentTemplate](#) (see page 965)

getSoftwareDeploymentTemplateList

The getSoftwareDeploymentTemplateList method returns a list of software template names and their last modified date, based on filter and sort criteria specified.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

SoftwareDeploymentTemplateFilter[] sdTemplateFilterArray

Specifies the filter condition on the properties of the software deployment template.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

enum SOFTWAREDEPOYMENTTEMPLATEPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int numberOfTemplatesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Xsd:boolean refreshList

Specifies whether you want to recompute the result set for each call.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

GetTemplateListResponse

Contains a list of software template names and their last modified date.

More information:

[SoftwareDeploymentTemplateFilter](#) (see page 968)

getSoftwareDeploymentTemplateListDetailed

The getSoftwareDeploymentTemplateListDetailed method returns the software deployment templates object list based on filter and sort criteria specified.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

SoftwareDeploymentTemplateFilter[] sdTemplateFilterArray

Specifies the filter condition on the properties of the software deployment template.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

enum SOFTWAREDEPLOYMENTTEMPLATEPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int numberOfTemplatesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

GetSoftwareDeploymentTemplateListDetailedResponse

Contains the list of software templates along with their details, retrieved based on the specified filter and sort properties.

More information:

[SoftwareDeploymentTemplateFilter](#) (see page 968)

[SOFTWAREDEPLOYMENTTEMPLATEPROPERTY](#) (see page 131)

[GetSoftwareDeploymentTemplateListDetailedResponse](#) (see page 968)

createAutomationProcess

The createAutomationProcess method creates an automation process (migration job).

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

AutomationProcess automationProcess

Specifies the properties of the automation process.

xsd:string [] listOfgroupNames

Specifies a list of computer groups to be migrated.

xsd:string [] listOfComputerNames

Specifies a list of computers to be migrated.

DataHandler dataHandler

Specifies the handle to the machine mapping file.

Return Values

Returns zero on successful initiation of job creation task else throws Automation Web Service Exception.

deleteAutomationProcesses

The deleteAutomationProcesses method deletes a list of automation processes.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string [] automationProcessNames

Specifies the list of automation process names you want to delete.

Return Values

Returns zero on successful deletion of all processes; otherwise, throws AutomationWebServiceExceptionList as a fault.

More information:

[ActionType](#) (see page 134)

executeAutomationProcess

The executeAutomationProcess method initiates the migration or revalidation of an automation process.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the name of the automation process you want to execute.

enum actionType action

Specifies the type of action.

Return Values

Returns zero on successful execution; otherwise, throws AutomationWebServiceException as a fault.

getAutomationProcessList

The getAutomationProcessList method returns basic information about the automation processes, based on the filtering criteria.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

AutomationProcessFilter[] apTemplateFilter

Specifies the filter condition on some of the properties of the automation process.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

enum AUTOMATIONPROCESSPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int noOfAutomationProcessesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

GetAutomationProcessListResponse

Returns a list automation process names and their last modified date.

More information:

[AutomationProcessFilter](#) (see page 969)

[GetAutomationProcessListResponse](#) (see page 969)

[AUTOMATIONPROCESSPROPERTY](#) (see page 132)

getAutomationProcessListDetailed

The getAutomationProcessListDetailed method returns a list of automation process names and their last modified dates, based on the filtering criteria.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

AutomationProcessFilter[] apTemplateFilterArray

Specifies the filter condition on some of the properties of the automation process.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

enum AUTOMATIONPROCESSPROPERTY sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

int noOfAutomationProcessesRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

GetAutomationProcessListDetailedResponse

Returns the properties of the filtered automation processes.

More information:

[AutomationProcessFilter](#) (see page 969)

[GetAutomationProcessListDetailedResponse](#) (see page 970)

[AUTOMATIONPROCESSPROPERTY](#) (see page 132)

getAutomationResult

The getAutomationResult method returns the result of the automation process.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the automation process name for which you want to retrieve the results.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

AutomationResultResponse

Contains the automation process results.

getAutomationResult2

The getAutomationResult2 method returns the results of the automation process. This methods supports sort and filtering operations.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the automation process name for which you want to retrieve the results.

AutomationResultFilter [] automationResultFilter

Specifies the filter on some of the automation result properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

enum AutomationResultProperty sortProperty

Specifies the property based on which you want to sort the results.

xsd:boolean sortAscending

Specifies whether you want to sort the list in ascending order.

int startIndex

Specifies an index number from which you want the objects to be returned from the result set. This is a zero based index.

xsd:int numberOfRecordsRequired

Specifies the number of objects you want to retrieve from the startIndex position.

Return Values

Returns the following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

AutomationResultResponse

Contains the automation process results.

More information:

[AutomationResultResponse](#) (see page 974)

[AutomationResultProperty](#) (see page 133)

getAutomationProcessTargetStatus

The getAutomationProcessTargetStatus method returns the migration status for the specified target computer.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the automation process name. The process name must be a valid and the automation process must at least be in the Validation phase. Otherwise, an error message "No Activity information available" is returned.

xsd:string targetName

Specifies the target computer name.

xsd:string sourceName

Specifies the source computer name.

xsd:string targetUUID

Specifies the UUID of the target computer.

xsd:string sourceUUID

Specifies the UUID of the source computer.

xsd:string processInstanceName

Specifies the IT PAM process instance name.

AutomationTargetStatus status

Specifies the migration status of the target computer.

Return Values

Returns Automation Process Target Status on success; otherwise, throws AutomationWebServiceException as a fault.

More information:

[AutomationTargetStatus](#) (see page 133)

getAutomationProcessStatus

The getAutomationProcessStatus method returns the migration job status.

Parameters**xsd:string sessionId**

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the automation process name. The process name must be a valid and the automation process must at least be in the Validation phase. Otherwise, an error message "No Activity information available" is returned.

Return Values

Returns the automation process status; otherwise, throws AutomationWebServiceException as a fault.

runAutomationSoftwareJobContainer

The runAutomationSoftwareJobContainer method launches the software job container execution from CA IT PAM process.

Note: This method can only be called from a workflow within CA IT PAM process and cannot be used as a standalone method.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd: int containerType

Specifies whether to run the preinstall or postinstall job container. Following are valid values:

0 (GENERIC)

Defines a generic job container. When this parameter is set, the generic job execution is initiated, using the information provided in SD template.

1 (PREINSTALL)

Defines a pre-install job container. When this parameter is set, the pre-install job execution is initiated, using the information provided in SD template.

2 (POSTINSTALL)

Defines a post-install job container. When this parameter is set, the post-install job execution is initiated, using the information provided in SD template.

xsd:string AutomationTaskUUID

Specifies the ID of the automation job which has spawned the CA IT PAM process.

xsd:string machineUUID

Specifies the UUID of the computer participating in automated migrations. This ID identifies an CA IT PAM instance launched for the computer. This ID is unique for each computer participating in the automation job.

Return Values

Returns the following information on success; otherwise, throws AutomationWebServiceException as a fault.

xsd:string jobId

Contains a handle that is used in subsequent calls related to automation job container. There will be one unique automation service JobID per computer regardless of how the jobs are optimized in CA ITCM.

runAutomationOSInstallJob

The runAutomationOSInstallJob method launches the OS deployment execution from CA IT PAM process.

Note: This method can only be called from a workflow within CA IT PAM process and cannot be used as a standalone method.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string AutomationTaskUUID

Specifies the ID of the automation job which has spawned this CA IT PAM process.

xsd:string machineUUID

Specifies the UUID of the computer participating in automated migrations. This ID identifies an CA IT PAM instance launched for the computer. This ID is unique for each computer, participating in an automation job.

Return Values

Returns the following information on success; otherwise, throws AutomationWebServiceException as a fault.

xsd:string jobId

Contains a handle that is used in subsequent calls related to automation job container.

runAutomationGenericJobContainer

The runAutomationGenericJobContainer method executes the automation generic job container. This method will be called from CA IT PAM process. Once a job is scheduled for execution, the job cannot be rerun because the JobID cannot be used to rerun the job. This eliminates the need to call a method to close the generic job container. Based on the BatchMode, the success status of the generic job container is determined.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationTaskUUID

Specified the ID that identifies the automation job that has spawned this ITPAM process.

xsd:string machineUUID

Specifies the ID that identifies the machine participating in automated migrations. Automation services provides this ID to ITPAM process instance.

SoftwareContainerData softwareContainerData,

Specifies software container data.

SoftwareJobData softwareJobData

Specifies software job data.

Return Values

Returns the following information on success; otherwise, throws AutomationWebServiceException as a fault.

xsd:string jobId

Contains a handle that is used in subsequent calls related to automation generic job container.

More information:

[SoftwareContainerData](#) (see page 971)

[SoftwareJobData](#) (see page 972)

getActivityLog

The getActivityLog method provides the activity log for a computer participating in the automation process, identified by computer name.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the automation process name. The process name must be a valid and the automation process must at least be in the Validation phase. Otherwise, an error message "No Activity information available" is returned.

xsd:String machineName

Specifies the computer name for which activity log is required. The computer name must be valid, and must be the original name of the machine. For example, if machine Test001-XP was submitted for Win7 deployment and is being renamed as Test001-W7, then this parameter should be Test001-XP. Otherwise an error "Invalid machine name" is returned.

Return Values

Returns the following sequence on success; otherwise, throws `AutomationWebServiceException` as a fault.

ActivityLog

Contains the `ActivityLog`, which provide chronically ordered activities for each requested state.

getActivityLog2

The `getActivityLog2` method provides the activity log for a computer participating in the automation process, identified by the computer UUID.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the automation process name. The process name must be a valid and the automation process must at least be in the Validation phase. Otherwise, an error message "No Activity information available" is returned.

xsd:String machineUUID

Specifies the computer UUID for which activity log is required.

Return Values

Returns the following sequence on success; otherwise, throws `AutomationWebServiceException` as a fault.

ActivityLog

Contains the `ActivityLog`, which provide chronically ordered activities for each requested state.

setActivationStatus

The setActivationStatus method enables or disables a list of computers for migrations.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

xsd:string automationProcessName

Specifies the automation process name. The process name must be a valid and the automation process must at least be in the Validation phase. Otherwise, an error message "No Activity information available" is returned.

xsd:string [] listOfComputers

Specifies the list of computers on which you want to enable or disable migration.

boolean enableForMigration

Specifies whether you want to enable the list of computer for migration.

Return Values

Returns 0 on success; otherwise, throws AutomationWebServiceException as a fault.

getSupportedOSForAssessment

The getSupportedOSForAssessment method returns a list of OS family ClassIDs supported for assessment.

Parameters

xsd:string sessionId

Specifies the session ID obtained from the DSM web services login.

Return Values

Returns the list of following sequence on success; otherwise, throws AutomationWebServiceException as a fault.

OperatingSystemType

Contains operating system name and its classID.

More information:

[OperatingSystemType](#) (see page 977)

SDTemplateProperties

The SDTemplateProperties sequence includes the properties of CA Service Desk templates.

Xsd:string templateName

Specifies the template Name.

Xsd:string templateId

Specifies the template ID.

SoftwarePackageFilter

The BootServerFilter sequence specifies a search filter that you can use for searching or listing boot servers.

dsm:FILTERCONDITION condition

Specifies the filter condition you want to apply.

dsm:SoftwarePackageProperty biProperty

Specifies the property you want to filter.

xsd:string searchString

Specifies the string you want to search in the name property.

dsm: DateTime2 dateTime

Specifies the date and time value you want to search in the last modified date property.

Xsd:long searchValue

Specifies the search value for xsd:long type properties.

dsm:SoftwarePackagingType pkgType

Specifies the search value for dsm:SoftwarePackagingType type properties.

dsm: PackageFormat format

Specifies the search value for dsm:PackageFormat type properties.

dsm: SoftwarePackageFiledBy filedBy

Specifies the search value for dsm: SoftwarePackageFiledBy type properties.

dsm: ArchivePath archivedAt

Specifies the search value for dsm:ArchivePath type properties.

SoftwarePackageProperties2

The SoftwarePackageProperties2 sequence contains the software package properties.

xsd:string softwarePackageId

Specifies the UUID identifying the software package.

xsd:string softwarePackageName

Specifies the name of the software package.

xsd:string softwarePackageVersion

Specifies the version of the software package.

xsd:string basedOnSoftwarePackageId

Specifies the UUID of the software package based on which this software package is created or from which this software package is imported. A software package can be created or imported based on another software package.

xsd:string basedOnSoftwarePackageName

Specifies the name of the software package based on which this software package is created or from which this software package is imported. A software package can be created or imported based on another software package.

xsd:string basedOnSoftwarePackageVersion

Specifies the version of the software package based on which this software package is created or from which this software package is imported. A software package can be created or imported based on another software package.

xsd:string supplier

Specifies the vendor of the software package.

xsd:string comment

Specifies the comment for the software package.

xsd:string sourcePath

Specifies the package source path.

dsm:SoftwarePackageType softwarePackageType

Specifies the type of software package.

xsd:boolean userSelectable

Specifies whether the package is selectable from the SD Catalog.

dsm:SourceMedium softwarePackageDataSource

Specifies the source medium for the software package data source.

dsm:SoftwarePackageFiledBy softwarePackageFiledBy

Specifies the user who created the software package.

dsm:DateTime registrationDateTime

Specifies the date and time the software package was created.

dsm:DateTime changeDateTime

Specifies the date and time the software package was last changed.

dsm:SoftwarePackagingType packageType

Specifies the type of the software package. The only type of package you can create is a GENERIC package type.

xsd:long packetSize

Specifies the package size.

xsd:long buildOnSize

Specifies the build on size of the package.

xsd:string archivePath

Specifies the archive path of the package, if the package is archived.

xsd:string archiveComment

Specifies the archive comment, if the package is archived.

xsd:string archivedBy

Specifies the user who performed the archive, if the package is archived.

dsm:ArchivePath archiveFlag

Specifies the archive flag of the package, if the package is archived.

dsm:DateTime archDateTime

Specifies the date and time when the package was archived, if the package is archived.

SDProperties

The SDProperties sequence includes the software delivery properties of a computer.

xsd:string jobCalendar

Specifies the name of the job calendar, if any.

xsd:Boolean softwareManagedSystem

Specifies whether to push only the packages created by CA software packager to the computer.

dsm:RACPolicy racPolicy

Specifies the RAC Policy for the computer.

dsm:DownloadMethod downloadMethod

Specifies the download method configured for the computer.

ManagerProperties

The ManagerProperties sequence includes the manager properties.

xsd:string managerUUID

Specifies the UUID of the manager.

xsd:boolean isEnterprise

Specifies whether the manager is an enterprise manager or domain manager.

xsd:string language

Specifies the installed language on manager.

dsm:ManagerVersion version

Specifies the manager version.

ManagerVersion

The ManagerVersion sequence specifies the different versions of the manager that are retrieved.

xsd:int majorVersion

Specifies the CA ITCM major version on the manager.

xsd:int minorVersion

Specifies the CA ITCM minor version on the manager.

xsd:int buildNumber

Specifies the CA ITCM build number on the manager.

xsd:int patchNumber

Specifies the CA ITCM patch number on the manager.

CreateSoftwareJobContainerProperties4

The CreateSoftwareJobContainerProperties4 sequence of elements specifies the job container properties you want initialize when a software job container is created. This sequence is derived from the CreateSoftwareJobContainerProperties3 sequence and includes additional parameters for opening a CA Service Desk ticket when a job fails.

xsd:string parentSoftwareJobContainerId

Specifies the parent software job container ID. If no value is provided, the root “__SD_Jobs” is used as the parent.

xsd:boolean parentSoftwareJobContainerIdSupplied

Specifies true or false depending on whether the parentSoftwareJobContainerId parameter is supplied.

xsd:string name (m)

Specifies the name of the software job container.

xsd:string comment

Specifies the comment for the software job container.

xsd:boolean commentSupplied

Specifies true or false depending on whether the comment parameter is supplied.

xsd:long priority

Specifies the priority that can range from 1 (highest) to 10 (lowest), If no value is provided, the default value, 5, is used.

xsd:long prioritySupplied

Specifies true or false depending on whether the priority parameter is supplied.

SoftwareJobContainerPropertyMask

softwareJobContainerPropertyMask

Specifies the mask for the software job container property. If no value is provided, the default value, linkageBatch and Cascade, is used.

xsd:boolean softwareJobContainerPropertyMaskSupplied

Specifies true or false depending on whether the softwareJobContainerPropertyMask parameter is supplied.

xsd:string userMessage

Specifies the custom administrator message.

xsd:int userMessageSupplied

Checks whether or not the custom administrator message is provided.

Xsd:boolean openTicketOnFailure

Specifies whether to create a CA Service Desk ticket on failure of jobs in job container.

Xsd:String templateName

Specifies the name of the CA Service Desk ticket template for creating ServiceDesk requests. Set this parameter only if you have set the openTicketOnFailure parameter to True.

BootImageFilter

The BootImageFilter sequence specifies a search filter that you can use for searching or listing boot images.

dsm:FILTERCONDITION condition

Specifies the filter condition you want to apply.

dsm: BootImageProperty biProperty

Specifies the property you want to filter.

xsd:string searchString

Specifies the string you want to search in the name property.

dsm: DateTime2 dateTime

Specifies the date and time value you want to search in the last modified date property.

BootImageProperties

The BootImageProperties sequence specifies the boot image properties.

xsd:string bootImageName

Specifies the name of the boot image.

Xsd:string typeIndicator

Specifies the type of the boot image.

Xsd:string platform

Specifies the operating system of the boot image.

dsm:DateTime2 registeredDateTime

Specifies the date and time of the boot image registration.

Xsd:string sdPackageName

Specifies the software delivery package associated with the boot image.

Xsd:string sdPackageVersion

Specifies the version of the software delivery package associated with the boot image.

Xsd:string sdPackageComment

Specifies the comment included in the software delivery package associated with the boot image.

Xsd:string imageFlags

Specifies the FIPS image flag associated with the boot image.

BootServerProperties

The BootServerProperties sequence contains the boot server properties.

xsd:string serverName

Specifies the boot server name.

Xsd:string bsStatus

Specifies the boot server status.

Xsd:string platform

Specifies the operating system of the boot server.

Xsd:string nfsSupport

Specifies the NFS support for boot server.

Xsd:string accessMethod

Specifies the access method configured for the boot server.

dsm:DateTime2 lastReport

Specifies the date and time of last boot server report.

dsm:DateTime2 lastResponse

Specifies the date and time of last boot server response.

dsm:DateTime2 lastReportReq

Specifies the date and time when the last boots server was requested.

Xsd:string pxelInterfaces

Specifies the configured PXE interface.

Xsd:string sdPkgWin

Specifies the software delivery package name for Windows.

Xsd:string sdPkgLinux

Specifies the software delivery package name for Linux.

Xsd:string sdPkgR11

Specifies the software delivery package for r11 version.

OSIMTargetProperties2

The OSIMTargetProperties includes the following parameters:

xsd:string targetName

Specifies the name of the target computer.

xsd:string targetMACAddress

Specifies MAC address of the target computer.

dsm:WS-OSIM-CONFIG-STATE configType

Specifies the configuration state of the target computer.

dsm:DateTime2 stateChangedDateTime

Specifies the changed or activated date and time of the target computer.

xsd:string imageName

Specifies the name of the OS image.

Xsd:string createdDateTime

Specifies the created date and time of the target computer.

OSIMTargetProperties

The following are the OSIMTargetProperties:

xsd:string targetName

Specifies the name of the target computer.

xsd:string targetMACAddress

Specifies MAC Address of the target computer.

dsm:WS-OSIM-CONFIG-STATE configType

Specifies the configuration state of the target computer.

dsm:DateTime2 stateChangedDateTime

Specifies the changed or activated date and time of the target computer.

xsd:string imageName

Specifies the name of the OS image.

BootServerFilter

The BootServerFilter sequence specifies a search filter that you can use for searching or listing boot servers.

dsm:FILTERCONDITION condition

Specifies the filter condition you want to apply.

dsm: BootServerProperty biProperty

Specifies the property you want to filter.

xsd:string searchString

Specifies the string you want to search in the name property.

dsm: DateTime2 dateTime

Specifies the date and time value you want to search in the last modified date property.

ComputerProperties2

The following sequences of elements specify the properties required when you are retrieving the properties of a computer.

xsd:string computerUUID

Specifies the UUID of the computer.

xsd:string computerLabel

Specifies the display name of the computer.

xsd:string computerDomainUUID

Specifies the UUID identifying the CA ITCM domain of the computer.

xsd:string computerPrimaryNetworkAddress

Specifies the IP address of the computer in 0.0.0.0 format.

xsd:string computerHostUUID

Specifies the agent-generated UUID that identifies the computer.

xsd:string computerAssetSourceUUID

Specifies the UUID identifying the common CA asset to which this computer is linked.

xsd:string computerHostName

Specifies the host name of the computer.

xsd:string computerSerialNumber

Specifies the serial number of the computer.

xsd:string computerDiscSerialNumber

Specifies the serial number of the computer disc.

xsd:string computerPrimaryMACAddress

Specifies the MAC address of the computer. The string must have a maximum length of 64 characters.

xsd:string computerAssetTag

Specifies the string value that identifies a computer asset.

xsd:long computerClassID

Specifies the numeric value indicating the class of the computer object. For example, Operating System.

dsm:UsageListMask computerUsageList

Specifies the computer usage list mask.

xsd:string computerURI

Specifies the fully qualified Uniform Resource Identifier of the computer.

xsd:string computerSystemID

Specifies the user-defined identifier for computer.

xsd:string computerVendorName

Specifies the name of the computer vendor.

xsd:string computerDomainLabel

Specifies the display name of the DSM domain to which the computer belongs.

xsd:string computerCreationUser

Specifies the name of the user account that created the computer record.

dsm:DateTime computerCreationDate

Specifies the computer creation date and time.

xsd:string computerLastUpdatedUser

Specifies the name of the user account that last updated the computer properties.

dsm:DateTime computerLastUpdatedDate

Specifies the date and time when the computer properties were last updated.

xsd:long computerVersionNumber

Specifies the version number of the computer record.

Note: The version number is incremented each time a record is updated.

xsd:string agentName

Specifies the agent name of the computer.

xsd:long agentRestrictions

Specifies the agent restrictions.

xsd:string agentDescription

Specifies the agent description.

xsd:string userdata1

Specifies the user defined data1.

xsd:string userdata2

Specifies the user defined data2.

xsd:string userdata3

Specifies the user defined data3.

xsd:string userdata4

Specifies the user defined data4.

xsd:string scalabilityserverName

Specifies the name of the scalability server that the agent is pointing to.

dsm:DateTime2 agentCreationDateTime

Specifies the agent creation data and time.

dsm:DateTime2 lastRunDateTime

Specifies the agent last run date and time.

xsd:string agentIPAddress

Specifies the IP address of the agent.

xsd:long agentTrustlevel

Specifies the trust level of the agent.

xsd:string agentOrigin

Specifies the origin of the agent.

dsm:AMAgentStatus amStatus

Specifies the AM Agent status.

dsm:SDAgentStatus sdStatus

Specifies the SD Agent status.

dsm:RCAgentStatus rcStatus

Specifies the RC Agent status.

dsm:ProtectionStatus protectionStatus

Specifies the Protection Agent status.

dsm:ASPStatus aspStatus

Specifies the ASP status.

dsm:AVStatus avStatus

Specifies the AV status.

dsm:HIPSStatus hipsStatus

Specifies the HIPS status.

QueryFilter

The QueryFilter sequence specifies a search filter that you can use for searching or listing queries.

dsm: QueryFilterCondition filterCondition

Specifies the filter condition you want to apply.

dsm: QueryProperty queryProperty

Specifies the property you want to filter.

xsd:string searchString

Specifies the string you want to search in the name property.

dsm: QueryTargetType targetType

Specifies the query target type.

QueryProperties2

The QueryProperties2 sequence contains the properties of a query.

xsd:string queryUUID

Specifies the UUID of the query.

Xsd:string queryName

Specifies the name of the query.

dsm: QueryTargetType queryType

Specifies the type of the query.

Xsd:boolean replicated

Specifies whether is query is replicated from enterprise manager. If true, it indicates that the query is replicated from enterprise manager.

QueryProperty

The QueryProperty sequence specifies the query properties on which you can apply the sort and filter conditions.

QUERY-LABEL

Specifies the name of the query.

QUERY-TARGET-TYPE

Specifies the target type of the query.

QueryFolderProperties

The QueryFolderProperties sequence contains the properties of a query folder.

Xsd:string folderName

Specifies the name of the query folder.

Xsd: long folderId

Specifies the ID of the query folder.

HWModelFilter

The HWModelFilter sequence specifies a search filter that you can use for searching or listing unique hardware models.

dsm: HWMODEL-FILTER-CONDITION filterCondition

Specifies the filter condition you want to apply.

xsd:string searchString

Specifies the string you want to search in the hardware model name.

Health Monitoring Sequences

UpdateAlertsProperties

The properties of the HM Alert to be updated.

xsd:boolean updateState.

Specifies the state of alert. When set to false, the state of the alert is not updated. However, the userMessage is modified.

xsd:boolean ignorePreviousState.

Defines the state of alert. When set to true, no error message is seen when new state of alert matches with the old state.

dsm:ALERTSTATE alertstate.

Indicates enumerated elements (New = 0, FollowUp = 1 and Cleared = 2), to change the state of the HM Alert.

xsd:boolean updateUserMessage.

Specifies True or False depending on the value of userMessage.

xsd:string userMessage.

Indicates the user description with the HM Alert.

Sequences for Automation Web Services

The Automation Services uses the sequences listed in the subsequent section for automated OS migration functionality.

AssessmentTemplate

The AssessmentTemplate sequence contains the properties of an assessment template. You can use this sequence to refer to a new or existing template.

xsd:string name

Specifies the name of the assessment template.

dsm:AssessmentTemplateType assessmentTemplatetype

Specifies the assessment template type.

dsm:ArrayOfOperatingSystemType operatingSystemsArray

Specifies an array of operating system types.

dsm:ArrayOfParameter parameters

Specifies the assessment attributes supplied, which needs to be used for assessment. There could be 'n' number of parameters. However each type of assessment will use only a fixed set of parameters.

xsd:DateTime lastModifiedDate

Retrieves the last updated date of the template. This parameter contains null value for createAssessmentTemplate and updateAssessmentTemplate methods.

xsd:DateTime creationDate

Retrieves the creation date of the template. This parameter contains null value for createAssessmentTemplate and updateAssessmentTemplate methods.

xsd:String lastModifiedUser

Retrieves the last modified user of the template. This parameter contains null value for createAssessmentTemplate and updateAssessmentTemplate methods.

xsd:String creationUser

Retrieves the creation user of the template. This parameter contains null value for createAssessmentTemplate and updateAssessmentTemplate methods.

xsd:String templateDescription

Retrieves the description of the template.

AssessmentTask

The AssessmentTask sequence contains the properties of an assessment job.

xsd:string taskName

Specifies the name for the assessment job.

xsd:string superUserName

Specifies the super user name. Super user is the administrator of the computer on which the automation service runs.

xsd:string superUserPasswd

Specifies the super user password.

AssessmentTemplate assessmentTemplate

Specifies the assessment template to be used for running the job.

ArrayOfParameter

The ArrayOfParameter sequence specifies the array of assessment parameters.

dsm:Parameter[] parameterList

Specifies an array of assessment parameters.

Parameter

The Parameter sequence specifies the name, data type, and value of a parameter.

dsm:AssessmentTemplateParameterDataType dataType

Specifies the data type of the assessment parameter.

dsm:AssessmentTemplateParameterName name

Specifies the name of the assessment parameter.

xsd:string value

Specifies the value for the assessment parameter.

AssessmentTemplateFilter

The AssessmentTemplateFilter sequence specifies the search filter for searching and listing assessment templates.

dsm:ASSESSMENTTEMPLATEPROPERTY assessmentTemplateProperty

Specifies the property you want to filter.

dsm:FILTERCONDITION condition

Specifies the filter condition you want to apply.

xsd:string searchString

Specifies the string you want to search in the name property.

DateTime dateTime

Specifies the date and time value you want to search in the last modified date property.

GetTemplateListResponse

The GetTemplateListResponse sequence returns the number of templates and basic information about the templates.

int totalNoOfRecords

Specifies the total number of templates available.

ArrayOfTemplateProperties templatePropertiesArray

Specifies the array of template properties.

TemplateProperties

The TemplateProperties sequence specifies the properties of the template.

string templateName

Specifies the name of the template.

DateTime templateLastModifiedDate

Specifies the last modified date of the template.

xsd:DateTime creationDate

Retrieves the creation date of the template. This parameter contains null value for createAssessmentTemplate and updateAssessmentTemplate methods.

xsd:String lastModifiedUser

Retrieves the last modified user of the template. This parameter contains null value for createAssessmentTemplate and updateAssessmentTemplate methods.

xsd:String creationUser

Retrieves the creation user of the template. This parameter contains null value for createAssessmentTemplate and updateAssessmentTemplate methods.

xsd:String templateDescription

Retrieves the description of the template.

GetAssessmentTemplateListDetailedResponse

The GetAssessmentTemplateListDetailedResponse sequence contains the list of templates and the number of records retrieved.

int totalNoOfRecords

Contains the total number of records retrieved after applying the filter condition.

ArrayOfAssessmentTemplate assessmentTemplateArray

Specifies the array of assessment templates.

AssessmentResult

The AssessmentResult sequence contains the result of an assessment job.

ResultGroup grpName

Specifies the group type. The assessment job creates various groups depending on whether computers are ready for OS. This parameter contains the different groups.

xsd:string groupUUID

Specifies the group UUID.

AssessmentTaskInfo

The AssessmentTaskInfo sequence specifies the details of the assessment job.

xsd:string name

Specifies the assessment task name.

AssessmentTemplate assessmentTemplate

Contains assessment template properties.

xsd:string group

Specifies the group name on which this assessment task was executed.

AssessmentStatus status

Specifies the assessment task status.

DateTime creationTime

Specifies creation time of the task.

String creationUser

Specifies who created the task.

DateTime executionFinishTime

Specifies completion time of the task.

DateTime executionStartTime

Specifies the start time of execution of the task.

String executionUser

Specifies execution user who executed the job.

AssessmentTaskFilter

The AssessmentTaskFilter sequence specifies the search filter for searching and listing assessment jobs.

enum ASSESSMENTTASKPROPERTY assessmentTaskProperty

Specifies the property you want to filter.

enum FILTERCONDITION condition

Specifies the filter condition you want to apply.

xsd:string searchString

Specifies the string you want to search in the name property.

Date dateTime

Specifies the date and time value you want to search in the last modified date property.

GetAssessmentTaskListResponse

The GetAssessmentTaskListResponse sequence contains the list of assessment jobs and the number records retrieved.

int totalNoOfRecords

Contains the total number of records retrieved after applying the filter condition.

ArrayOfAssessmentTaskInfo assessmentTaskInfoArray

Specifies the array of assessment tasks.

OsimTemplate

The OsimTemplate sequence contains the details of an OSIM template.

xsd:string name

Specifies OSIM template name.

OsimTemplateType type

Specifies OSIM template type.

GlobalType globalType

Specifies the OS image information as given in [ImageType](#) (see page 964).

ArrayOfGroupType groupTypeArray

Specifies the array of group information mapped against the OS template.

ArrayOfHardWareModelType hwModelTypeArray

Specifies the array of hardware models mapped against the OS template.

Date lastModifiedDate

Specifies the last modified date of the OSIM template.

xsd:DateTime creationDate

Retrieves the creation date of the template. This parameter contains null value for createOsimTemplate and updateOsimTemplate methods.

xsd:String lastModifiedUser

Retrieves the last modified user of the template. This parameter contains null value for createOsimTemplate and updateOsimTemplate methods.

xsd:String creationUser

Retrieves the creation user of the template. This parameter contains null value for createOsimTemplate and updateOsimTemplate methods.

xsd:String templateDescription

Retrieves the description of the template.

CustomizationParameter

The CustomizationParameter sequence specifies the OS installation parameters.

xsd:string name

Specifies OS installation parameter name.

xsd:string value

Specifies OS installation parameter value.

ImageType

The ImageType sequence specifies the OS image you want to use in the OSIM template.

xsd:string name

Specifies the OS image name.

xsd:string imageUUID

Specifies the UUID of the OS Image ID. This ID is used to uniquely identify the OS image to be deployed.

ArrayOfCustomizationParameter customizationParameters

Specifies an array of OS installation parameters. This is a key-value pair.

OsimTemplateFilter

The OsimTemplateFilter sequence specifies the search filter for searching and listing OSIM templates.

dsm:OSIMTEMPLATEPROPERTY OSIMTemplateProperty

Specifies the property you want to filter.

dsm:FILTERCONDITION condition

Specifies the filter condition you want to apply.

xsd:string searchString

Specifies the string you want to search in the name property.

DateTime dateTime

Specifies the date and time value you want to search in the last modified date property.

GetOsimTemplateListDetailedResponse

The GetOsimTemplateListDetailedResponse sequence contains the list of OSIM templates and the number records retrieved.

int totalNoOfRecords

Contains the total number of records retrieved after applying the filter condition.

ArrayOfOsimTemplate osimTemplatesArray

Specifies the array of osim templates

SoftwareDeploymentTemplate

The SoftwareDeploymentTemplate sequence contains the properties of a software deployment template. You can use this sequence to refer to a new or existing template.

xsd:string name

Specifies a unique name for the software deployment.

SoftwareJobContainerType preInstallJobs

Specifies the software packages to be installed before the OS Deployment.

SoftwareJobContainerType postInstallJobs

Specifies the software packages to be installed after the OS Deployment.

SoftwarePackageMappingType packageMapping

Defines the mapping between an old and new software packages.

xsd:DateTime lastModifiedDate

Specifies last modified date of the template.

xsd:DateTime creationDate

Retrieves the creation date of the template. This parameter contains null value for createSoftwareDeploymentTemplate and updateSoftwareDeployemntTemplate methods.

xsd:String lastModifiedUser

Retrieves the last modified user of the template. This parameter contains null value for createSoftwareDeploymentTemplate and updateSoftwareDeployemntTemplate methods.

xsd:String creationUser

Retrieves the creation user of the template. This parameter contains null value for createSoftwareDeploymentTemplate and updateSoftwareDeployemntTemplate methods.

xsd:String templateDescription

Retrieves the description of the template.

SoftwareJobContainerType

The SoftwareJobContainerType sequence specifies the job properties of the software job container.

SoftwarePackageType [] softwarePackages

Specifies the array of software packages.

xsd:string userMessage

Specifies the user message.

boolean promptUser

Specifies whether you want to prompt the user with the message.

boolean allowCancellation

Specifies whether you want to allow the user to cancel the job.

boolean executeOnPromptTimeout

Specifies whether you want to execute jobs, if the user prompt times out.

boolean preventUserLogOn

Specifies whether you want to prevent users from logging in while job executes.

long promptTimeOutHours

Specifies the number of hours for which you want to display a user prompt before forced execution of the job starts.

Default: 24 hours

long jobTimeOutHours

Specifies the time (in hours) after which, if the job is unable to complete, the job will cease attempts to complete.

Default: 48 hours

boolean storePackageAtSSLibrary

Specifies whether you want to store packages in the scalability server library.

SoftwarePackageType

The SoftwarePackageType sequence specifies the properties of the software package.

xsd:string name

Specifies the software package name.

xsd:string version

Specifies the software package version.

xsd:string procedure

Specifies the software package procedure you want to use.

int bootLevelPriorExec

Specifies the restart options requested for the software job before the job is executed.

Default: noReboot

int bootLevelAfterExec

Specifies the restart options requested for the software job after the job is executed.

Default: noReboot

xsd:string userMessage

Specifies the user message.

SoftwarePackageMappingType

The SoftwarePackageMappingType sequence specifies the mapping between an old and a new software package. If you want to replace a software with a newer version of the software after OS deployment, you can map such software using this sequence.

SoftwarePackageType source package

Specifies the package type defining the source software package.

SoftwarePackageType targetpackage

Specifies the package type defining the target software package.

SoftwareDeploymentTemplateFilter

The SoftwareDeploymentTemplateFilter sequence specifies the search filter for searching and listing software deployment templates.

dsm:SOFTWAREDEPLOYMENTTEMPLATEPROPERTY sdTemplateProperty

Specifies the property you want to filter.

dsm:FILTERCONDITION condition

Specifies the filter condition you want to apply.

xsd:string searchString

Specifies the string you want to search in the name property.

DateTime dateTime

Specifies the date and time value you want to search in the last modified date property.

GetSoftwareDeploymentTemplateListDetailedResponse

The GetSoftwareDeploymentTemplateListDetailedResponse sequence contains the list of software deployment templates and the number records retrieved.

int totalNoOfRecords

Contains the total number of records retrieved after applying the filter condition.

ArrayOfSoftwareDeploymentTemplate sdTemplateArray

Specifies the array of available software deployment templates.

AutomationProcess

The AutomationProcess sequence contains the details of an automation process.

xsd:string name

Specifies automation process name. The name must be unique.

xsd:string OSIMTemplateName

Specifies OSIM template name.

xsd:string softwareDeploymentTemplateName

Specifies software deployment template name.

xsd:string processInstanceName

Specifies process instance name.

xsd:string superUserName

Specifies the user name of the CA IT PAM users that you have configured in CA ITCM.

xsd:string superUserpasswd

Specifies the password of the CA IT PAM users that you have configured in CA ITCM.

xsd:DateTime executionTime

Specifies the execution time of the process.

enum AutomationProcessMode automationProcessMode

Specifies automation process mode.

enum AutomationProcessStatus processStatus

Specifies automation process status.

xsd:string StatusText

Specifies additional information regarding status of the automation job created.

AutomationProcessFilter

The AutomationProcessFilter sequence specifies the search filter for searching and listing automation processes.

dsm:AUTOMATIONPROCESSPROPERTY apProperty

Specifies the property you want to filter.

dsm:FILTERCONDITION condition

Specifies the filter condition you want to apply.

xsd:string searchString

Specifies the string you want to search in the name property.

DateTime dateTime

Specifies the date and time value you want to search in the last modified date property.

GetAutomationProcessListResponse

The GetAutomationProcessListResponse sequence contains the list of automation processes and the number records retrieved.

int totalNoOfRecords

Contains the total number of records retrieved after applying the filter condition.

ArrayOfAutomationProcessBasicInfo apBasicInfoArray

Specifies the list of automation processes basic information.

GetAutomationProcessListDetailedResponse

The GetAutomationProcessListDetailedResponse sequence contains the details of the automation processes.

xsd:int totalNoOfRecords

Specifies the total number of available automation processes.

ArrayOfAutomationProcessDetailed automationProcessArrayDetailed

Specifies the list of automation processes.

AutomationProcessListDetailed

The AutomationProcessListDetailed contains the detailed information about the automation process.

OsimTemplate OSIMTemplate

Specifies the OSIM template parameters used in the automation process.

SoftwareDeploymentTemplate SDTemplate

Specifies the software deployment template parameters used in the automation process.

String additionalInfo

Provides additional information in case of a failure.

DateTime creationTime

Specifies the creation time of the job.

String creationUser

Specifies the user who created the job.

DateTime executionFinishTime

Specifies the completion time of the job.

DateTime executionStartTime

Specifies the start time of the job execution.

String executionUser

Specifies the super user name provided while creating the job.

String jobName

Specifies the name of the job.

AutomationProcessMode mode

Specifies the mode of the created job.

AutomationProcessStatus status

Specifies the current status of the job.

AutomationProcessBasicInfo

The AutomationProcessBasicInfo contains the details of the basic information about the automation processes.

xsd:string name

Specifies the automation process name.

xsd:string processInstanceName

Specifies process instance name.

enum AutomationProcessMode processMode

Specifies automation process mode.

enum AutomationProcessStatus processStatus

Specifies automation process status.

String additionalInfo

Provides additional information in case of job execution or job creation failure.

DateTime creationTime

Specifies creation time of the job.

String creationUser

Specifies the user who created the job.

DateTime executionFinishTime

Specifies the completion time of the job.

DateTime executionStartTime

Specifies the start time of the job.

String executionUser

Specifies the user who executed the job.

SoftwareContainerData

The SoftwareContainerData sequence contains details about the software container used in automation process.

xsd:boolean BatchMode

Specifies whether you want to execute the jobs in batch mode. If set to false the jobs will execute in No Linkage mode.

xsd:string name

Specifies the software container name.

xsd:long priority

Specifies the job priority.

Range: 1 (highest) to 10 (lowest)

Default: 5

xsd:string userMessage

Specifies the custom user message.

SoftwareJobData

The SoftwareJobData sequence contains details about the software jobs.

xsd:dateTime activationTime

Specifies the execution date. If not specified, the default value is the current date and time of the domain manager computer.

xsd:boolean allowCancellation

Allows end users to cancel the prompt for the software job.

xsd:int bootLevelAfterExec

Specifies the restart options required for the software job after the job is executed.

Default: 0 (noReboot)

xsd:int bootLevelPriorExec

Specifies the restart options required for the software job before the job is executed.

Default: 0 (noReboot)

xsd:string deliveryCalendar

Specifies the name of a delivery calendar to be used for scheduling the job. The calendar controls the days, dates, and times when delivery is permitted. The time cells that are not set in the calendar define the exclusion intervals. If the name of the calendar is unknown to SD, the job is not created.

xsd:dateTime deliveryTime

Specifies the delivery date and time. If not supplied, the default value is the current date and time of the domain manager computer.

xsd:boolean executeOnPromptTimeout

Specifies whether to execute jobs, if prompt times out.

xsd:boolean ignoreJobCalendarsOnTargetComputers

Specifies whether to ignore job calendars, if any, on target computers.

xsd:long jobTimeOutHours

Specifies the number of hours after the job the will timeout if it was unable to complete.

Default: 168 hours

xsd:boolean jobsTriggerSS

Specifies whether the job must trigger the scalability server.

xsd:boolean logonShield

Specifies whether to prevent users from being logged on while job executes.

xsd:string name

Specifies the software job name.

xsd:long promptTimeOutHours

Specifies the number of hours for which you want to display a user prompt before forced execution of the job starts.

Default: 24 hours

xsd:boolean promptUser

Specifies a message to be prompted on the target computer before the job starts.

xsd:boolean resolveQueryGroups

Specifies whether to resolve the target groups before the job is set up.

xsd:boolean runAtShutdown

Specifies whether to run while the system is allowed to shutdown.

xsd:boolean runofflineFromSS

Specifies whether to execute procedure offline from the scalability server.

SoftwarePackageData softwarePackage

Specifies the software package data.

xsd:boolean storePackageAtSSLibrary

Specifies whether to store the packages in the scalability server library.

xsd:boolean timesRelativeToEM

Specifies whether the time is relative to enterprise manager.

xsd:boolean useDeliveryCalender

Specifies whether the deliveryCalendar parameter is supplied.

xsd:string userParameters

Specifies the user-defined parameters.

SoftwarePackageData

The SoftwarePackageData sequence contains details about the software package.

xsd:string packageName

Specifies the software package name.

xsd:string procedureName

Specifies the name of the procedure to be run.

xsd:string versionNumber

Specifies the version number of the software package.

AutomationResultResponse

The AutomationResultResponse sequence contains the result of the automation process.

int totalNumberOfTargets

Specifies the number of computers involved in the automation process.

enum AutomationProcessStatus jobStatus

Specifies the job status. This is the overall job status.

ArrayOfAutomationProcessTargetStatus status

Specifies the array of target status.

String additionalInfo

Specifies additional information in case of job execution failure.

SoftwareJobContainerParameter

The SoftwareJobContainerParameter sequence specifies the details of software job container.

xsd:string parameterName

Specifies the name of the software job container parameter.

xsd:string parameterValue

Specifies the value of the software job container parameter.

ActivityLog

The ActivityLog sequence contains the activity log details:

xsd:string activityMessage

Specifies the activity message.

enum LogLevel logLevel

Specifies the log level.

xsd:dateTime timestamp

Specifies the timestamp of the message.

ActivityLogResponse

The ActivityLogResponse sequence contains the following information:

ArrayOfActivityLog activityLogArray

Specifies an array of activity log entries.

GlobalType

The GlobalType sequence contains the following information:

ImageType imageType

Specifies information about the OS image that is applicable for all the target computers.

GroupType

The GroupType sequence specifies the following information about the group type. This sequence can have multiple group information mapped against an OS image.

xsd:string name

Specifies the name of the group.

ImageType imageType

Specifies information about the OS image mapped to the group.

HardwareModelType

The HardwareModelType sequence contains the following information:

xsd:string name

Specifies name of the hardware model.

ImageType imageType

Specifies information about the OS image mapped to the hardware model.

AutomationProcessTargetStatus

The AutomationProcessTargetStatus sequence contains the following information:

xsd:string targetName

Specifies the target name.

xsd:string targetUUID

Specifies the UUID of the target computer.

enum AutomationTargetStatus status

Specifies the automation status of the target computer.

ArrayOfAutomationProcessTargetStatus

The ArrayOfAutomationProcessTargetStatus sequence contains the following information:

AutomationProcessTargetStatus [] targetStatusArray

Specifies an array of target status.

AutomationResultFilter

The AutomationResultFilter sequence contains the following information:

enum AutomationResultProperty automationResultProperty

Specifies the automation result property.

enum FilterCondition filterCondition

Specifies the filter condition.

xsd:string searchString

Specifies the string you want to search in the specified property.

AutomationTargetStatus status

Specifies the target status.

AutomationWebServiceException

The AutomationWebServiceException sequence contains the following information:

xsd:string errorCode

Specifies the error code.

xsd:string errorMessage

Specifies the error message.

xsd:string additionalInfo

Specifies additional error details, if any.

AutomationWebServiceExceptionList

The AutomationWebServiceExceptionList sequence contains the following information:

ArrayOfAutomationWebServiceException
automationWebServiceExceptionArray

Specifies the array of AutomationWebServiceExceptions.

ArrayOfAutomationWebServiceException

The ArrayOfAutomationWebServiceException sequence contains the following information:

AutomationWebServiceException []
automationWebServiceExceptionArray

Specifies the array of AutomationWebServiceExceptions.

OperatingSystemType

The OperatingSystemType sequence contains the following information:

xsd:string name

Specifies the operating system name.

xsd:long classID

Specifies the operating system classID.

ArrayOfAutomationProcessListDetailed

The ArrayOfAutomationProcessListDetailed sequence includes the automation process properties.

AutomationProcessListDetailed[] automationProcessListDetailed

Specifies the list of automation processes with detailed information about OSIM and SD Options.

ArrayOfAutomationProcess

The ArrayOfAutomationProcess sequence includes the automation process properties.

AutomationProcess[] automationProcess

Specifies an array of automation process properties.

ArrayOfAutomationProcessBasicInfo

The ArrayOfAutomationProcessBasicInfo sequence contains basic information about the automation process.

AutomationProcessBasicInfo[] apBasicInfo

Specifies an array of basic information about the automation processes.

ArrayOfAutomationProcessTargetStatus

The ArrayOfAutomationProcessTargetStatus sequence specifies a list of target statuses.

AutomationProcessTargetStatus [] targetStatusArray

Specifies the list of target statuses.

ArrayOfActivityLog

The ArrayOfActivityLog sequence contains the following information:

ActivityLog [] activityLog

Specifies the array of activity logs.

ArrayOfAssessmentTaskInfo

The ArrayOfAssessmentTaskInfo sequence contains the following information:

AssessmentTaskInfo [] assessmentTasks

Specifies an array of assessment tasks.

ArrayOfAssessmentTemplate

The ArrayOfAssessmentTemplate sequence contains the following information:

AssessmentTemplate [] assessmentTemplates

Specifies an array of assessment templates.

ArrayOfGroupType

The ArrayOfGroupType sequence contains the following information:

GroupType [] groupType

Specifies an array of group type. This parameter can have multiple group information mapped against an OS image. For example:
Group1 -> OSImage1, Group3-> OSImage2.

ArrayOfHardWareModelType

The ArrayOfHardWareModelType sequence contains the following information:

HardWareModelType [] hwModelType

Specifies an array of hardware model types. This parameter can have multiple hardware models mapped against OS images. For example:HWModelName->OSImage1, HWModel2->OSImage3.

ArrayOfOsimTemplate

The ArrayOfOsimTemplate sequence contains the following information:

OsimTemplate [] osimTemplates

Specifies the array of osim templates

ArrayOfSoftwareDeploymentTemplate

The ArrayOfSoftwareDeploymentTemplate sequence contains the following information:

SoftwareDeploymentTemplate [] sdTemplates

Specifies an array of software deployment templates.

ArrayOfSoftwarePackageMappingType

The ArrayOfSoftwarePackageMappingType sequence contains the following information:

SoftwarePackageMappingType [] softwarePackageMappingTypeArray

Specifies an array of software package mapping types.

ArrayOfSoftwarePackageType

The ArrayOfSoftwarePackageType sequence contains the following information:

SoftwarePackageType [] softwarePackageTypeArray

Specifies an array of software package types.

ArrayOfTemplateProperties

The ArrayOfTemplateProperties sequence contains the following information:

TemplateProperties [] templates

Specifies an array of template properties.

ArrayOf_xsd_string

The ArrayOf_xsd_string sequence contains the following information:

xsd:string [] stringArray

Specifies an array of string values.

ArrayOfCustomizationParameter

The ArrayOfCustomizationParameter sequence contains the following information:

CustomizationParameter [] customizationParameters

Specifies an array OS installation parameters that you want to customize. This parameter provides a placeholder for OS deployment parameters.

ArrayOfOperatingSystemType

The ArrayOfOperatingSystemType sequence contains the following information:

OperatingSystemType[] operatingSystemTypeArray

Specifies the array of operating system type.

Array of Elements

This section represents all the arrays that form a wrapper around its complex types. For example, if GetSoftwarePackages returns an array of SoftwarePackageProperties structures, this is encapsulated as an element of ArrayOfSoftwarePackageProperties structures. The following is a list of the elements:

- ArrayOf[ComputerProperties](#) (see page 700)
- ArrayOfComputerPropertiesByName
- ArrayOf[ComputerPropertyFilter](#) (see page 671)
- ArrayOfComputerPropertiesHolder
- ArrayOfComputerStatusInfoHolder

- `ArrayOfDeliveryDistributionContainerProperties`
- `ArrayOfDeploymentPackageProperties` (see page 697)
- `ArrayOfDiscoveredSoftwareProperties` (see page 700)
- `ArrayOfDistributionContainerOrderProperties` (see page 701)
- `ArrayOfDistributionContainerProperties` (see page 691)
- `ArrayOfDistributionDomainProperties` (see page 702)
- `ArrayOfDomainGroupProperties` (see page 704)
- `ArrayOfDomainGroupPropertiesHolder`
- `ArrayOfDomainProperties` (see page 705)
- `ArrayOfDomainPropertyFilter` (see page 747)
- `ArrayOfDomainPropertiesHolder`
- `ArrayOfFetchItemJobOrderProperties` (see page 710)
- `ArrayOfFileSystemElementPath` (see page 710)
- `ArrayOfParameterInfo` (see page 707)
- `ArrayOfParameterInfo arrayOfParamInfo` (see page 707)
- `ArrayOfQueryFragment` (see page 82)
- `ArrayOfQueryFragment2` (see page 82)
- `ArrayOfQueryProperties` (see page 716)
- `ArrayOfQueryResultProperties` (see page 708)
- `ArrayOfSecurityAreas`
- `ArrayOfSecurityAreaByName` (see page 207)
- `ArrayOfSecurityAreasByName`
- `ArrayOfSecurityAreasByNameHolder`
- `ArrayOfSecurityAreasHolder`
- `ArrayOfSecurityAreaProperties` (see page 722)
- `ArrayOfSecurityClassPermissions` (see page 722)
- `ArrayOfSecurityClassesProperties` (see page 723)
- `ArrayOfSecurityClassType` (see page 86)
- `ArrayOfSecurityGroupPermissionsForAllSecurityProfiles` (see page 724)
- `ArrayOfSecurityGroupProfilePermission` (see page 724)

- ArrayOf[SecurityObjectLinkageProperties](#) (see page 724)
- ArrayOf[SecurityObjectOwner](#) (see page 725)
- ArrayOf[SecurityObjectPermissionsForAllSecurityProfiles](#) (see page 726)
- ArrayOf[SecurityObjectProfilePermission](#) (see page 724)
- ArrayOf[SecurityObjectsSecurityAreaLinkages](#) (see page 725)
- ArrayOf[SecurityProfileProperties](#) (see page 726)
- ArrayOf[SecurityProfilesSecurityGroupsPermissions](#) (see page 989)
- ArrayOf[SetSecurityClassPermissions](#) (see page 190)
- ArrayOf[SetSecurityGroupPermissions](#) (see page 728)
- ArrayOf[SetSecurityObjectPermissions](#) (see page 728)
- ArrayOf[SetSecurityObjectsSecurityAreaLinkages](#) (see page 189)
- ArrayOfSoftwareJob
- ArrayOf[SoftwareJobContainerProperties](#) (see page 786)
- ArrayOf[SoftwareJobProperties](#) (see page 768)
- ArrayOf[SoftwarePackageGroupProperties](#) (see page 828)
- ArrayOf[SoftwarePackageProcedureDependencyProperties](#) (see page 837)
- ArrayOf[SoftwarePackageProcedureGroupProperties](#) (see page 838)
- ArrayOf[SoftwarePackageProcedureProperties](#) (see page 843)
- ArrayOf[SoftwarePackageProperties](#) (see page 847)
- ArrayOf[SoftwarePackageVolumeElementProperties](#) (see page 850)
- ArrayOf[SoftwarePackageVolumeProperties](#) (see page 836)
- ArrayOf[SoftwarePolicyExceptionProperties](#) (see page 851)
- ArrayOf[SoftwarePolicyProperties](#) (see page 851)
- ArrayOfSoftwarePolicyPropertiesByName
- ArrayOfSSPackageInfo
- Arrayofstring

- ArrayOf[UnitInventoryComponentProperties](#) (see page 868)
- ArrayOfUnitInventoryGroupProperties
- ArrayOf[UnitInventoryItemProperties](#) (see page 869)
- ArrayOf[UnitProperties](#) (see page 877)
- ArrayOf[UnitPropertyFilter](#) (see page 877)
- ArrayOf[UnitSoftwareJobProperties](#) (see page 880)
- ArrayOf[UserAccountProperties](#) (see page 890)
- ArrayOf[UserAccountPropertyFilter](#) (see page 891)
- ArrayOfUserAccountPropertiesHolder
- ArrayOf[UserProfileProperties](#) (see page 892)
- ArrayOf[UserProfilePropertyFilter](#) (see page 897)
- ArrayOfUserProfilePropertiesHolder

ArrayOfSoftwareJob

Members of sequence softwareJobs:

ArrayOfSoftwareJobProperties3

The ArrayOfSoftwareJobProperties3 array includes the array size and the software job properties.

xsd:int __size

Specifies the size of the array.

dsm:SoftwareJobProperties3

Specifies the software job properties.

ArrayOfTargetProperties2

Specifies the array of OSIM target properties, one for each target computer.

dsm:OSIMTargetProperties2 props

Contains OSIM target computer properties.

ArrayOfSoftwarePackageProcedureProperties2

The ArrayOfSoftwarePackageProcedureProperties2 array includes the array size and the software package procedure properties.

xsd:int __size

Specifies the size of the array.

dsm: SoftwarePackageProcedureProperties2

Specifies the software package procedure properties.

ArrayOfBootServerFilter

The ArrayOfBootServerFilter sequence contains array of boot server filter conditions, one for each boot server property.

dsm:BootServerFilter filter

Specifies the filter conditions for some of the properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

ArrayOfBootImageProperties

The ArrayOfBootImageProperties sequence contains an array of boot image properties, one for each boot image.

dsm: BootImageProperties biProps

Contains boot image properties for each image.

ArrayOfBootServerProperties

The ArrayOfBootServerProperties sequence contains an array of boot server properties, one for each boot server.

dsm: BootServerProperties bsProps

Contains boot server properties.

ArrayOfUnitSoftwareJobProperties3

The ArrayOfUnitSoftwareJobProperties3 sequence contains the array of UnitSoftwareJobProperties3.

dsm:UnitSoftwareJobProperties3

Contains unit software job properties.

ArrayOfBootImageFilter

The ArrayOfBootImageFilter sequence contains array of boot image filter conditions, one for each boot image property.

dsm:BootImageFilter filter

Specifies the filter conditions for some of the properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

ArrayOfSoftwarePackageProperties2

The ArrayOfSoftwarePackageProperties2 array includes the array size and the software package properties.

xsd:int__size

Specifies the size of the array.

dsm:SoftwarePackageProperties2

Contains the software package properties, including Package Format.

Remarks

None

ArrayOfSoftwarePackageProcedureProperties4

The ArrayOfSoftwarePackageProcedureProperties4 sequence contains the array of SoftwarePackageProcedureProperties4.

- SoftwarePackageProcedureProperties4
- softwarePackageProcedurePropertiesArray

ArrayOfSoftwarePackageFilter3

The ArrayOfSoftwarePackageFilter3 sequence of elements contains the array of software package filter conditions.

dsm: SoftwarePackageFilter3 filter

Specifies the filter condition for each property.

xsd:boolean matchall

Specifies whether the software package set to AND operation (True) in filter conditions or OR operation (False).

SoftwarePackageFilter

Specifies a search filter when you are searching or listing boot servers.

dsm: FILTERCONDITION condition

Specifies the applied filtercondition.

dsm: SoftwarePackageProperty bsProperty

Specifies the search property.

xsd:string searchString

Specifies the search value for xsd:string type.

dsm: DateTime2 dateTime

Specifies the search value for dsm: DateTime2 type property.

Xsd:long searchValue

Specifies the search value for xsd:long type properties.

ArrayOfSoftwarePackageProperties3

The ArrayOfSoftwarePackageProperties3 sequence of elements contains the array of software package properties.

- dsm:SoftwarePackageProperties3 softwarePackageProperties

ArrayOfUnitSoftwareJobProperties2

The ArrayOfUnitSoftwareJobProperties2 array includes the array size and the unit software job properties.

xsd:int __size

Specifies the size of the array.

dsm:UnitSoftwareJobProperties2

Contains the unit software job properties, including Package Format.

Remarks

None

ArrayOfSecurityAreaByName

xsd:int __size

Number of [security area](#) (see page 750)s.

dsm:SecurityAreaByName securityArea

It contains the SecurityArea names and the SecurityAreaEnabled value.

ArrayOfComponents

The ArrayOfComponents sequence contains the array of components, one for each component.

dsm:COMPONENTS component

Contains the component value.

ArrayOfComponentStatus

The ArrayOfComponentStatus sequence contains an array of components and their statuses, one for each component.

dsm:COMPONENTS component

Specifies the component value.

dsm:COMPONENT-STATUS status

Specifies the component status.

ArrayOfConfigurationParams

The ArrayOfConfigurationParams contains an array of configuration parameters one for each parameter.

dsm:ConfigurationParams params

Contains the configuration parameter name.

ArrayOfConfigurationParamValues

The ArrayOfConfigurationParamValues sequence contains an array of configuration parameters and its values for each parameter.

dsm:ConfigurationParams params

Contains the configuration parameter name.

xsd:int paramValue

Contains the parameter value.

ArrayOfComputerProperties2

Defines the Computerproperties2 sequence.

dsm:ComputerProperties2 computerPropertiesArray

Contains computer properties.

ArrayOfQueryFilter

The ArrayOfQueryFilter sequence contains an array of query filter conditions, one for each property.

dsm:QueryFilter filter

Specifies the filter conditions for some of the properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

ArrayOfHWModelFilter

The ArrayOfHWModelFilter sequence contains an array of Hardware model filter conditions

dsm: HWModelFilter filter

Specifies the filter conditions for hardware models.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

ArrayOfQueries

The ArrayOfQueries sequence contains query properties, one for each query.

dsm: QueryProperties2 queryProps

Contains query properties for each query.

ArrayOfQueryFolderProperties

The ArrayOfQueryFolderProperties sequence contains the array of query folder properties, one for each query folder.

dsm: QueryFolderProperties properties

Contains query folder properties for each folder.

ArrayOfSecurityClassPermissions

An array of [SecurityClassPermissions](#) (see page 722) – one for each security profile.

ArrayOfSecurityProfilesSecurityGroupsPermissions

xsd:int __size

Number of security groups.

**dsm:SecurityProfilesSecurityGroupsPermissions
securityGroupsPermissionsArray**

It contains security permissions for the security group with respect to the security profile.

ArrayOfComponents

The ArrayOfComponents sequence contains the array of components, one for each component.

dsm:COMPONENTS component

Contains the component value.

ArrayOfComponentStatus

The ArrayOfComponentStatus sequence contains an array of components and their statuses, one for each component.

dsm:COMPONENTS component

Specifies the component value.

dsm:COMPONENT-STATUS status

Specifies the component status.

ArrayOfConfigurationParams

The ArrayOfConfigurationParams contains an array of configuration parameters one for each parameter.

dsm:ConfigurationParams params

Contains the configuration parameter name.

ArrayOfConfigurationParamValues

The ArrayOfConfigurationParamValues sequence contains an array of configuration parameters and its values for each parameter.

dsm:ConfigurationParams params

Contains the configuration parameter name.

xsd:int paramValue

Contains the parameter value.

ArrayOfComputerProperties2

Defines the Computerproperties2 sequence.

dsm:ComputerProperties2 computerPropertiesArray

Contains computer properties.

ArrayOfQueryFilter

The ArrayOfQueryFilter sequence contains an array of query filter conditions, one for each property.

dsm:QueryFilter filter

Specifies the filter conditions for some of the properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

ArrayOfQueries

The ArrayOfQueries sequence contains query properties, one for each query.

dsm:QueryProperties2 queryProps

Contains query properties for each query.

ArrayOfQueryFolderProperties

The ArrayOfQueryFolderProperties sequence contains the array of query folder properties, one for each query folder.

dsm:QueryFolderProperties properties

Contains query folder properties for each folder.

ArrayOfHWModelFilter

The ArrayOfHWModelFilter sequence contains an array of Hardware model filter conditions

dsm:HWModelFilter filter

Specifies the filter conditions for hardware models.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

ArrayOfSDTemplateProperties

The ArrayOfSDTemplateProperties sequence includes an array of CA Service Desk template properties, one for each template in the list.

dsm:SDTemplateProperties props

Contains CA Service Desk template properties.

ArrayOfSoftwarePackageFilter

The ArrayOfSoftwarePackageFilter sequence contains an array of software package filter conditions, one for each property.

dsm:SoftwarePackageFilter filter

Specifies the filter conditions for some of the properties.

xsd:boolean matchall

Specifies whether you want all the filter conditions to be satisfied. Set the parameter to True to perform an AND operation. Set the parameter to False to perform an OR operation.

ArrayOfSoftwarePackageProperties2

The ArrayOfSoftwarePackageProperties2 sequence contains an array of software package properties, one for each package.

dsm:SoftwarePackageProperties2

Contains the software package properties.

ArrayOfOSIMProperties

Contains an array of [OSIMProperties](#) (see page 798).

dsm:OSIMProperties osimProperties

Contains the name and value pairs for each OSIM property.

ArrayOfOSIMTargetProperties

The ArrayOfOSIMTargetProperties sequence includes [ArrayOfOSIMProperties](#) (see page 992).

dsm:ArrayOfOSIMProperties osimPropertiesArray

Includes an array of OSIM properties.

ArrayOfOSIMTargetInstallationRequestsProperties

dsm:ArrayOfOSIMProperties osimPropertiesArray

The ArrayOfOSIMTargetInstallationRequestsProperties sequence includes [ArrayOfOSIMProperties](#) (see page 992).

ArrayOfGetAssetScriptJobsProperties

The ArrayOfGetAssetScriptJobsProperties sequence contains the array of script job properties.

dsm:GetAssetScriptJobsProperties getAssetScriptJobsPropertiesArray

Contains the array of asset script job properties.

More information:

[GetAssetScriptJobsProperties](#) (see page 797)

ArrayOfDayOfTheWeekDependency

The ArrayOfDayOfTheWeekDependency sequence is a scheduling option used to specify the days in a week when you want the job to be run.

dsm:DayOfTheWeekDependency

dayOfTheWeekDependencyValueArray

Specifies the days in a week when you want the job to be run.

ArrayOfTriggerInfo

The ArrayOfTriggerInfo sequence is used to specify the list of objects to trigger.

dsm:TriggerInfo triggerInfoArray

Specifies the array that contains the job check information.

ArrayOfAMExtraCmdParam

The ArrayOfAMExtraCmdParam is used to specify the additional parameters for executing asset management job check.

dsm:AMExtraCmdParam cmdParamsArray

Specifies the array of asset management specific parameters.

More information:

[AMExtraCmdParam](#) (see page 110)

ArrayOfOSIMparameterDesc

dsm:OSIMparameterDesc osimParameterDesc

The ArrayOfOSIMparameterDesc includes the [OSIMparameterDesc](#) (see page 799) sequence.

ArrayOfOSIMPropertiesWithDesc

dsm:OSIMPropertiesWithDesc osimProperties

The ArrayOfOSIMPropertiesWithDesc includes the [OSIMPropertiesWithDesc](#) (see page 799) sequence which specifies the OSIM property name, property value, and its descriptions.

ArrayOfOSIMIRParametersWithDesc

The ArrayOfOSIMIRParametersWithDesc specifies the configuration state of the installation request and the properties of corresponding installation.

dsm:OSIMPropertiesWithDesc osimProperties

Specifies the properties of Installation request.

dsm:WS_OSIM_CONFIG_STATE configstate

Specifies the configuration state of the installation request.

ArrayOfOSIMTargetPropertiesWithDesc

The ArrayOfOSIMTargetPropertiesWithDesc specifies all the installation request properties for an OSIM computer.

dsm:ArrayOfOSIMIRParametersWithDesc osimIRArray

Specifies the installation request properties Array.

ArrayOfOSIMImageProperties

Specifies the properties of [OS image](#) (see page 992).

dsm:ArrayOfOSIMProperties osimPropertiesArray

ArrayOfServerProperties

dsm:ScalabilityServerProperties props

Specifies the array of [scalability server properties](#) (see page 799).

ArrayOfBootServerProperties

dsm:ArrayOfOSIMProperties osimPropertiesArray

Specifies the array of [boot server properties](#). (see page 992)

ArrayOfOSImageFilter

Specifies the Array of OS image filter conditions, one for each OS image property.

dsm:OSImageFilter osImageFilterArray

Contains OS image filter conditions.

ArrayOfOSImageProperties

Specifies the array of OS image properties, one for each OS image.

dsm:OSImageProperties props

Contains properties of the OS images that are registered at a domain manager.

ArrayOfOSIMTargetFilter

Specifies the array of OSIM computer filter conditions, one for each OSIM target property:

dsm:OSIMTargetFilter osimTargetFilterArray

Contains the computer filter conditions.

ArrayOfTargetProperties

Specifies the Array of OSIM target properties, one for each target computer

dsm:OSIMTargetProperties props

Contains OSIM target computer properties.

ArrayOfComputerPropertyFilter2

Specifies the Array of computer filter conditions. One for each computer property.

dsm:ComputerPropertyFilter2 computerPropertyFilterArray

Contains the computer filter conditions, properties, and values.

ArrayOfComputerProperties2

Defines the Computerproperties2 sequence.

dsm:ComputerProperties2 computerPropertiesArray

Contains computer properties.

ArrayOfOSImageProperties

Specifies the array of OS image properties, one for each OS image.

dsm:OSImageProperties props

Contains properties of the OS images that are registered at a domain manager.

ArrayOfParameterInfo arrayOfParamInfo

Deployment Package parameters list

xsd:int __size

Number of parameters.

dsm:ParameterInfo paramInfo

It contains array of [ParameterInfo](#) (see page 707) instances.

ArrayOfItems

This sequence has one or more Item sequences.

dsm: Item item.

ArrayOfstring

This sequence holds one or more Item sequences.

xsd:string stringElements

Defines the UUID of the item.

Limitations

The Unicenter CA ITCM Web Services have the following limitations:

- In Sd Client APIs, a single session cannot open multiple listhandles simultaneously and this affects open, get, and close APIs. If one open function has returned a listhandle, it must be closed before you use another open function.
- Unicenter CA ITCM Web Services Requires UTF-8 Encoding.
Unicenter CA ITCM Web Services requires all input strings to be UTF-8 encoded. All strings output from the Unicenter CA ITCM Web Services will be UTF-8 encoded.
- No support at Domain Manager Level for functionalities related to distribution job containers.
- The length of calendar names must not exceed 8 characters.

Configuration Parameters

The administrator for the webservices can configure the following parameters:

garbagecollectortimeout

Indicates the time out period for the web services to invoke the garbagecollector. By default, this is set to 300 seconds.

maxclients

Indicates the maximum number of client sessions that web services can support at a given point of time. By default, this is set to 100 sessions.

sessiontimeout

Indicates the timeout period for the web services to close the inactive sessions. By default, this is set to 1200 (20 minutes) seconds.

Unifiedloginenabled

Indicates if the unified login is enabled. By default, it is set to disabled.

SSL Configuration

Secure Access to CA ITCM Web Services can be enabled by configuring SSL on IIS and Apache Web Servers.

To Configure SSL on IIS Web Server, see the Microsoft website.

To Configure SSL on Apache Web Server, see the Apache website.

Passing System Defined Groups as Parameters

For all the webservices:

if you want to use the system defined groups as a parameter then use the following labels.

\$AllComputer
\$AllUsers
\$AllComputerUsers
\$AllDomains
\$AllQueries
\$AllJobs
\$AllEventLogs
\$AllServers
\$AllServerGroups
\$AllDomainGroups
\$AllExternalAssets
\$AllBackupUsers
\$AllBackupJobs
\$AllBackupRestoreJobs

Example:

If you want to use “All Computers” as a parameter to the API DeleteSoftwarePoliciesInUnitGroupsByName then use “\$Allcomputer”.

Chapter 3: Web Services Messages

The following messages apply to the CA ITCM Web Services.

WSA000053

Unable to allocate memory

Reason:

Memory could not be allocated.

Action:

Restart DSM Webservice. If the problem persists, contact CA Technical Support.

WSA000101

Invalid parameter 'sessionId'

Reason:

You may not have passed a sessionId.

Action:

You need to pass a valid session id retrieved from the DSM Web Service login method.

WSA000102

Session expired, please login again

Reason:

You may have called a WSAPI after the session has expired. The default time is 20 minutes.

Action:

You need to increase the sessionTimeout using WSAPI SetTimeoutValue or by pinging DSMWebservice using ResetTimeout before the session expires.

WSA000103

Session not found, please login again

Reason:

You may have passed a nonexistent SessionId.

Action:

You need to pass a valid SessionId retrieved from the Login method.

WSA000104

Failed to generate a new session id

Reason:

The DSM webservice failed to generate the Session Id.

Action:

Restart the DSM webservices. If the problem persists, contact CA Technical Support.

WSA000105

Session id already exists

Reason:

DSM Webservice might not have generated a Unique SessionId.

Action:

Restart the DSM webservices. If the problem persists, contact CA Technical Support.

WSA000106

Invalid operation

Reason:

An invalid operation has been performed.

Action:

Check the sequence of operations (WSAPI calls).

WSA000109

SD Component is not Installed

Reason:

You may be accessing the DSM Webservice component specific to SD and the SD component is not installed on the manager computer.

Action:

Install the SD component on the manager computer. If the problem persists, contact CA Technical Support.

WSA000110

AM Component is not Installed

Reason:

You may be accessing the DSM Webservice component specific to AM and the AM component is not installed on the manager computer.

Action:

Install AM component on the manager computer. If the problem persists, contact CA Technical Support.

WSA000111

Invalid SD handle. SD Session already aborted by manager, please login again

Reason:

Because of some internal error SD Manager has terminated the session.

Action:

Login again. If the problem persists, contact CA Technical Support.

WSA000112

Maximum number of clients are connected, please try after some time

Reason:

The number of users connected to webservices may have exceeded the default limit as set by the application server or by webservices.

Action:

Try to connected again later. If the problem persists, contact CA Technical Support.

WSA000114

AM requires authentication

or

SD requires authentication

Reason

The Web Service client is trying to access the asset management or software delivery functionality without first authenticating to the respective manager.

Action

The Web Service client must authenticate to the asset management or software delivery manager before accessing the respective functionality.

WSA000201

Invalid parameter 'userAccountName'

Reason:

You may not have passed userAccountName in the DSM Webservice API call.

Action:

You need to pass a valid userAccountName. If the problem persists, contact CA Technical Support.

WSA000202

Invalid parameter 'userAccountPropertiesRequired'

Reason:

You may not have passed userAccountPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid userAccountPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000203

Invalid parameter 'computerName'

Reason:

You may not have passed computerName in the DSM Webservice API call.

Action:

You need to pass a valid computer Name. If the problem persists, contact CA Technical Support.

WSA000204

Invalid parameter 'computerPropertiesRequired'

Reason:

You may not have passed computerPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid computerPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000205

Invalid parameter 'userProfileLabel'

Reason:

You may not have passed userProfileLabel in the DSM Webservice API call.

Action:

You need to pass a valid userProfileLabel, If the problem persists, contact CA Technical Support.

WSA000206

Invalid parameter 'userProfilePropertiesRequired'

Reason:

You may not have passed userProfilePropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid userProfilePropertiesRequired, If the problem persists, contact CA Technical Support.

WSA000207

Invalid parameter 'userAccount UUID'

Reason:

You may not have passed userAccountUUID in the DSM Webservice API call.

Action:

You need to pass a valid userAccountUUID. If the problem persists, contact CA Technical Support.

WSA000208

Invalid parameter 'computer UUID'

Reason:

You may not have passed computer UUID in the DSM Webservice API call.

Action:

You need to pass a valid computer UUID. If the problem persists, contact CA Technical Support.

WSA000209

Invalid parameter 'userProfile UUID'

Reason:

You may not have passed userProfile UUID in the DSM Webservice API call.

Action:

You need to pass a valid userProfile UUID, If the problem persists, contact CA Technical Support.

WSA000210

Invalid parameter 'unit UUID'

Reason:

You may not have passed unit UUID in the DSM Webservice API call.

Action:

You need to pass a valid unit UUID, If the problem persists, contact CA Technical Support.

WSA000211

Invalid parameter 'unitInventoryComponentPropertiesRequired'

Reason:

You may not have passed unitInventoryComponentPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid unitInventoryComponentPropertiesRequired. If the problem persists, call CA Technical Support.

WSA000212

Invalid parameter 'listHandle'

Reason:

You may not have passed listHandle in the DSM Webservice API call.

Action:

You need to pass a valid listHandle. If the problem persists call CA Technical Support.

WSA000213

Invalid parameter 'noOfUnitInventoryComponents'

Reason:

You may have requested zero or fewer units of UnitInventoryComponents.

Action:

You are expected to request one or more units of UnitInventoryComponents. If the problem persists, contact CA Technical Support.

WSA000214

Invalid parameter 'unitInventoryComponentProperties'

Reason:

You may not have passed unitInventoryComponentProperties in the DSM Webservice API call.

Action:

You need to pass a valid unitInventoryComponentProperties. If the problem persists call CA Technical Support.

WSA000215

Invalid parameter 'unitInventoryGroupPropertiesRequired'

Reason:

You may not have passed unitInventoryGroupPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid unitInventoryGroupPropertiesRequired. If the problem persists call CA Technical Support.

WSA000216

Invalid parameter 'noOfUnitInventoryComponentGroups'

Reason:

You may have requested zero or fewer units of UnitInventoryComponentGroups.

Action:

You are expected to request one or more units of UnitInventoryComponentGroups. If the problem persists, contact CA Technical Support.

WSA000217

Invalid parameter 'unitInventoryGroupProperties'

Reason:

You may not have passed unitInventoryGroupProperties in the DSM Webservice API call.

Action:

You need to pass a valid unitInventoryGroupProperties. If the problem persists call CA Technical Support.

WSA000218

Invalid parameter 'noOfUnitInventoryGroups'

Reason:

You may have requested zero or fewer units of UnitInventoryGroups '

Action:

You are expected to request one or more units of UnitInventoryGroups. If the problem persists, contact CA Technical Support.

WSA000219

Invalid parameter 'unitInventoryItemPropertiesRequired'

Reason:

You may not have passed unitInventoryItemPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid unitInventoryItemPropertiesRequired. If the problem persists call CA Technical Support.

WSA000220

Invalid parameter 'noOfUnitInventoryItems'

Reason:

You may have requested zero or fewer units of UnitInventoryItems

Action:

You are expected to request one or more units of UnitInventoryItems. If the problem persists, contact CA Technical Support.

WSA000221

Invalid parameter 'discoveredSoftwarePropertiesRequired'

Reason:

You may not have passed discoveredSoftwarePropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid discoveredSoftwarePropertiesRequired. If the problem persists call CA Technical Support.

WSA000222

Invalid parameter 'noOfDiscoveredSoftwares'

Reason:

You may have requested zero or fewer units of DiscoveredSoftwares '

Action:

You are expected to request one or more units of DiscoveredSoftwares. If the problem persists, contact CA Technical Support.

WSA000223

Invalid parameter 'unitGroupPropertiesRequired'

Reason:

You may not have passed unitGroupPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid unitGroupPropertiesRequired. If the problem persists call CA Technical Support.

WSA000224

Invalid parameter 'noOfUnitGroups'

Reason:

You may have requested zero or fewer units of UnitGroups '

Action:

You are expected to request one or more units of UnitGroups. If the problem persists, contact CA Technical Support.

WSA000225

Invalid parameter 'unitGroupId'

Reason:

You may not have passed unitGroupId in the DSM Webservice API call.

Action:

You need to pass a valid unitGroupId. If the problem persists call CA Technical Support.

WSA000226

Invalid parameter 'softwarePackageId'

Reason:

You may not have passed 'softwarePackageId' in the DSM Webservice API call.

Action:

You need to pass a valid 'softwarePackageId'. If the problem persists call CA Technical Support.

WSA000227

Invalid parameter 'UnitsoftwareJobPropertiesRequired'

Reason:

You may not have passed UnitsoftwareJobPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid UnitsoftwareJobPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000228

Invalid parameter 'noOfSoftwarePackageInstallations'

Reason:

You may have requested zero or fewer units of SoftwarePackageInstallations.

Action:

You are expected to request one or more units of SoftwarePackageInstallations. If the problem persists, contact CA Technical Support.

WSA000229

Invalid parameter 'createSoftwarePackageVolumeProperties'

Reason:

You may not have passed createSoftwarePackageVolumeProperties in the DSM Webservice API call.

Action:

You need to pass a valid createSoftwarePackageVolumeProperties. If the problem persists, call CA Technical Support.

WSA000230

Invalid parameter 'softwarePackageVolumelds'

Reason:

You may not have passed softwarePackageVolumelds in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageVolumelds. If the problem persists call CA Technical Support.

WSA000231

Invalid parameter 'softwarePackageVolumePropertiesRequired'

Reason:

You may not have passed softwarePackageVolumePropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageVolumePropertiesRequired. If the problem persists call CA Technical Support.

WSA000232

Invalid parameter 'noOfSoftwarePackageVolumes'

Reason:

You may have requested zero or fewer 'noOfSoftwarePackageVolumes'

Action:

You are expected to request one or more units of 'SoftwarePackageVolumes'. If the problem persists, contact CA Technical Support.

WSA000233

Invalid parameter 'softwarePackageVolumeld'

Reason:

You may not have passed softwarePackageVolumeld in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageVolumeld. If the problem persists call CA Technical Support.

WSA000234

Invalid parameter 'setSoftwarePackageVolumeProperties'

Reason:

You may not have passed setSoftwarePackageVolumeProperties in the DSM Webservice API call.

Action:

You need to pass a valid setSoftwarePackageVolumeProperties. If the problem persists, contact CA Technical Support.

WSA000235

Invalid parameter 'elementTypeMask'

Reason:

You may not have passed elementTypeMask in the DSM Webservice API call.

Action:

You need to pass a valid elementTypeMask. If the problem persists, contact CA Technical Support.

WSA000236

Invalid parameter 'softwarePackageVolumeElementPropertiesRequired'

Reason:

You may not have passed softwarePackageVolumeElementPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageVolumeElementPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000237

Invalid parameter 'totalNoOfElements'

Reason:

You may have requested zero or fewer units of 'SoftwarePackageVolumeElements'

Action:

You are expected to request one or more units of 'SoftwarePackageVolumeElement'. If the problem persists, contact CA Technical Support.

WSA000238

Invalid parameter 'softwarePackageName'

Reason:

You may not have passed softwarePackageName in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageName. If the problem persists, contact CA Technical Support.

WSA000239

Invalid parameter 'softwarePackageVersion'

Reason:

You may not have passed softwarePackageVersion in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageVersion. If the problem persists, contact CA Technical Support.

WSA000240

Invalid parameter 'softwarePackageProcedureName'

Reason:

You may not have passed softwarePackageProcedureName in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureName. If the problem persists, contact CA Technical Support.

WSA000241

Invalid parameter 'softwarePackageProcedureProperties'

Reason:

You may not have passed softwarePackageProcedureProperties in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureProperties. If the problem persists, contact CA Technical Support.

WSA000242

Invalid parameter 'softwarePackageProcedureIds'

Reason:

You may not have passed softwarePackageProcedureIds in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureIds. If the problem persists, contact CA Technical Support.

WSA000243

Invalid parameter 'softwarePackageProcedureTypes'

Reason:

You may not have passed softwarePackageProcedureTypes in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureTypes. If the problem persists, contact CA Technical Support.

WSA000244

Invalid parameter 'softwarePackageProcedurePropertiesRequired'

Reason:

You may not have passed softwarePackageProcedurePropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid 'softwarePackageProcedurePropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000245

Invalid parameter 'noOfSoftwarePackageProcedures'

Reason:

You may have requested zero or fewer units of SoftwarePackageProcedures.

Action:

You are expected to request one or more units of SoftwarePackageProcedures. If the problem persists, contact CA Technical Support.

WSA000246

Invalid parameter 'softwarePackageProcedureDependency'

Reason:

You may not have passed softwarePackageProcedureDependency in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureDependency. If the problem persists, contact CA Technical Support.

WSA000247

Invalid parameter 'softwarePackageProcedureDepIds'

Reason:

You may not have passed softwarePackageProcedureDependencyIds in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureDepIds. If the problem persists, contact CA Technical Support.

WSA000248

Invalid parameter 'sourcePath'

Reason:

You may not have passed sourcePath in the DSM Webservice API call.

Action:

You need to pass a valid sourcePath. If the problem persists, contact CA Technical Support.

WSA000249

Invalid parameter 'importSoftwarePackageProperties'

Reason:

You may not have passed importSoftwarePackageProperties in the DSM Webservice API call.

Action:

You need to pass a valid importSoftwarePackageProperties. If the problem persists, contact CA Technical Support.

WSA000250

Invalid parameter 'softwarePackagePropertiesRequired'

Reason:

You may not have passed softwarePackagePropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackagePropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000251

Invalid parameter 'createSoftwarePackageProperties'

Reason:

You may not have passed createSoftwarePackageProperties in the DSM Webservice API call.

Action:

You need to pass a valid createSoftwarePackageProperties. If the problem persists, contact CA Technical Support.

WSA000252

Invalid parameter 'softwarePackageIds'

Reason:

You may not have passed softwarePackageIds in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageIds. If the problem persists, contact CA Technical Support.

WSA000253

Invalid parameter 'softwarePackageProperties'

Reason:

You may not have passed softwarePackageProperties in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProperties. If the problem persists, contact CA Technical Support.

WSA000254

Invalid parameter 'softwarePackageMaskType'

Reason:

You may not have passed softwarePackageMaskType in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageMaskType. If the problem persists, contact CA Technical Support.

WSA000255

Invalid parameter 'noOfSoftwarePackages'

Reason:

You may have requested zero or fewer units of SoftwarePackages.

Action:

You are expected to request one or more units of SoftwarePackages. If the problem persists, contact CA Technical Support.

WSA000256

Invalid parameter 'softwarePackageVolumeSubPath'

Reason:

You may not have passed softwarePackageVolumeSubPath in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageVolumeSubPath. If the problem persists, contact CA Technical Support.

WSA000257

Invalid parameter 'fileSystemElementPath'

Reason:

You may not have passed fileSystemElementPath in the DSM Webservice API call.

Action:

You need to pass a valid fileSystemElementPath. If the problem persists, contact CA Technical Support.

WSA000258

Invalid parameter 'softwarePackageProcedureId'

Reason:

You may not have passed softwarePackageProcedureId in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureId. If the problem persists, contact CA Technical Support.

WSA000259

Invalid parameter 'setSoftwarePackageProcedureProperties'

Reason:

You may not have passed setSoftwarePackageProcedureProperties in the DSM Webservice API call.

Action:

You need to pass a valid setSoftwarePackageProcedureProperties. If the problem persists, contact CA Technical Support.

WSA000260

Invalid parameter 'softwarePackageProcedureDependencyPropertiesRequired'

Reason:

You may not have passed softwarePackageProcedureDependencyPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureDependencyPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000261

Invalid parameter 'noOfSoftwarePackageProcedureDependencies'

Reason:

You may have requested zero or fewer units of 'SoftwarePackageProcedureDependencies'

Action:

You are expected to request one or more units of SoftwarePackageProcedureDependencies. If the problem persists, contact CA Technical Support.

WSA000262

Invalid parameter 'softwarePackageGroupId'

Reason:

You may not have passed softwarePackageGroupId in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageGroupId. If the problem persists, contact CA Technical Support.

WSA000263

Invalid parameter 'softwarePackageGroupIds'

Reason:

You may not have passed softwarePackageGroupIds in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageGroupIds. If the problem persists, contact CA Technical Support.

WSA000264

Invalid parameter 'softwarePackageProcedureGroupId'

Reason:

You may not have passed softwarePackageProcedureGroupId in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureGroupId. If the problem persists, contact CA Technical Support.

WSA000265

Invalid parameter 'softwarePackageProcedureGroupIds'

Reason:

You may not have passed softwarePackageProcedureGroupIds in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureGroupIds. If the problem persists, contact CA Technical Support.

WSA000266

Invalid parameter 'softwarePolicyId'

Reason:

You may not have passed softwarePolicyId in the DSM Webservice API call.

Action:

You need to pass a valid softwarePolicyId. If the problem persists, contact CA Technical Support.

WSA000267

Invalid parameter 'softwarePolicyExceptionPropertiesRequired'

Reason:

You may not have passed softwarePolicyExceptionPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePolicyExceptionPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000268

Invalid parameter 'noOfSoftwarePolicyExceptions'

Reason:

You may have requested zero or fewer units of SoftwarePolicyExceptions.

Action:

You are expected to request one or more units of SoftwarePolicyExceptions. If the problem persists, contact CA Technical Support.

WSA000269

Invalid parameter 'softwareJobPropertiesRequired'

Reason:

You may not have passed softwareJobPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000270

Invalid parameter 'noOfSoftwareJobs'

Reason:

You may have requested zero or fewer units of SoftwareJobs.

Action:

You are expected to request one or more units of SoftwareJobs. If the problem persists, contact CA Technical Support.

WSA000271

Invalid parameter 'softwareJobContainerId'

Reason:

You may not have passed softwareJobContainerId in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobContainerId. If the problem persists, contact CA Technical Support.

WSA000272

Invalid parameter 'softwareJobContainerPropertiesRequired'

Reason:

You may not have passed softwareJobContainerPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobContainerPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000273

Invalid parameter 'noOfParentSoftwareJobContainers'

Reason:

You may have requested zero or fewer units of ParentSoftwareJobContainers.

Action:

You are expected to request one or more units of ParentSoftwareJobContainers. If the problem persists, contact CA Technical Support.

WSA000274

Invalid parameter 'noOfSoftwareJobContainers'

Reason:

You may have requested zero or fewer units of SoftwareJobContainers.

Action:

You are expected to request one or more units of SoftwareJobContainers. If the problem persists, contact CA Technical Support.

WSA000275

Invalid parameter 'softwarePackageGroupPropertiesRequired'

Reason:

You may not have passed softwarePackageGroupPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageGroupPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000276

Invalid parameter 'noOfParentSoftwarePackageGroups'

Reason:

You may have requested zero or fewer units of ParentSoftwarePackageGroups.

Action:

You are expected to request one or more units of ParentSoftwarePackageGroups. If the problem persists, contact CA Technical Support.

WSA000277

Invalid parameter 'softwarePackageProcedureGroupPropertiesRequired'

Reason:

You may not have passed softwarePackageProcedureGroupPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureGroupPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000278

Invalid parameter 'noOfSoftwarePackageProcedureGroups'

Reason:

You may have requested zero or fewer units of SoftwarePackageProcedureGroups.

Action:

You are expected to request one or more units of SoftwarePackageProcedureGroups. If the problem persists, contact CA Technical Support.

WSA000279

Invalid parameter 'noOfSoftwarePackageGroups'

Reason:

You may have requested zero or fewer units of SoftwarePackageGroups.

Action:

You are expected to request one or more units of SoftwarePackageGroups. If the problem persists, contact CA Technical Support.

WSA000280

Invalid parameter 'softwarePolicyPropertiesRequired'

Reason:

You may not have passed softwarePolicyPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid softwarePolicyPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000281

Invalid parameter 'noOfSoftwarePolicies'

Reason:

You may have requested zero or fewer units of SoftwarePolicies.

Action:

You are expected to request one or more units of SoftwarePolicies. If the problem persists, contact CA Technical Support.

WSA000282

Supplied name is invalid

Reason:

You may not have passed name in the following DSM Webservice API call:
FindSoftwarePolicy, **FindSoftwareJobContainer**, **FindSoftwarePackageProcedureGroup**,
or **FindSoftwarePackageGroup**.

Action:

You need to pass a valid name. If the problem persists, contact CA Technical Support.

WSA000284

Invalid parameter 'catalogSoftwarePackageGroup'

Reason:

You may not have passed catalogSoftwarePackageGroup in the DSM Webservice API call.

Action:

You need to pass a valid catalogSoftwarePackageGroup. If the problem persists, contact CA Technical Support.

WSA000285

Invalid parameter 'softwarePackageProcedureGroup'

Reason:

You may not have passed softwarePackageProcedureGroup in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageProcedureGroup. If the problem persists, contact CA Technical Support.

WSA000286

Invalid parameter 'softwarejobContainer'

Reason:

You may not have passed softwarejobContainer in the DSM Webservice API call.

Action:

You need to pass a valid softwarejobContainer. If the problem persists, contact CA Technical Support.

WSA000287

Invalid parameter 'softwarePolicy'

Reason:

You may not have passed softwarePolicy in the DSM Webservice API call.

Action:

You need to pass a valid softwarePolicy. If the problem persists, contact CA Technical Support.

WSA000288

Invalid parameter 'softwarePackageGroup'

Reason:

You may not have passed softwarePackageGroup in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageGroup. If the problem persists, contact CA Technical Support.

WSA000289

Invalid Ids

Reason:

You may not have passed UUIDs in the DSM Webservice API call.

Action:

You need to pass a valid UUIDs. If the problem persists call CA Technical Support.

WSA000290

Invalid parameter 'softwareJobsIds'

Reason:

You may not have passed softwareJobsIds in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobsIds. If the problem persists, contact CA Technical Support.

WSA000291

Invalid parameter 'softwarePackageInstallProcedureId'

Reason:

You may not have passed softwarePackageInstallProcedureId in the DSM Webservice API call.

Action:

You need to pass a valid softwarePackageInstallProcedureId. If the problem persists, contact CA Technical Support.

WSA000292

Invalid parameter 'softwareJobId'

Reason:

You may not have passed softwareJobId in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobId. If the problem persists, contact CA Technical Support.

WSA000293

Invalid parameter 'setSoftwareJobProperties'

Reason:

You may not have passed setSoftwareJobProperties in the DSM Webservice API call.

Action:

You need to pass a valid setSoftwareJobProperties. If the problem persists, contact CA Technical Support.

WSA000294

Invalid parameter 'noOfUnitsoftwareInstallationJobs'

Reason:

You may have requested zero or fewer units of UnitsoftwareInstallationJobs.

Action:

You are expected to request one or more units of UnitsoftwareInstallationJobs. If the problem persists, contact CA Technical Support.

WSA000295

Invalid parameter 'noOfUnitsoftwareJobs'

Reason:

You may have requested zero or fewer units of UnitsoftwareJobs.

Action:

You are expected to request one or more units of UnitsoftwareJobs. If the problem persists, contact CA Technical Support.

WSA000296

Invalid parameter 'noOfSoftwareJobUnits'

Reason:

You may have requested zero or fewer units of SoftwareJobUnit.

Action:

You are expected to request more than or at least one unit of SoftwareJobUnit. If the problem persists, contact CA Technical Support.

WSA000297

Invalid parameter 'softwareJobOrderProperties'

Reason:

You may not have passed softwareJobOrderProperties in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobOrderProperties. If the problem persists, contact CA Technical Support.

WSA000298

Invalid parameter 'softwareJobOrder'

Reason:

You may not have passed softwareJobOrder in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobOrder. If the problem persists, contact CA Technical Support.

WSA000299

Invalid parameter 'softwareJobOrderPolicyProperties'

Reason:

You may not have passed softwareJobOrderPolicyProperties in the DSM Webservice API call.

Action:

You need to pass a valid softwareJobOrderPolicyProperties. If the problem persists, contact CA Technical Support.

WSA000300

Invalid parameter 'noOfUserProperties'

Reason:

You may have requested zero or fewer units of UserProperties.

Action:

You are expected to request one or more units of UserProperties. If the problem persists, contact CA Technical Support.

WSA000301

Invalid parameter 'noOfUserProfiles'

Reason:

You may have requested zero or fewer units of UserProfiles.

Action:

You are expected to request one or more units of UserProfiles. If the problem persists, contact CA Technical Support.

WSA000302

Invalid parameter 'unitGroupIds'

Reason:

You may not have passed unitGroupIds in the DSM Webservice API call.

Action:

You need to pass a valid unitGroupIds. If the problem persists, contact CA Technical Support.

WSA000303

Invalid parameter 'unitGroupProperties'

Reason:

You may not have passed unitGroupProperties in the DSM Webservice API call.

Action:

You need to pass a valid unitGroupProperties. If the problem persists, contact CA Technical Support.

WSA000304

Invalid parameter 'userAccountIds'

Reason:

You may not have passed userAccountIds in the DSM Webservice API call.

Action:

You need to pass a valid userAccountIds. If the problem persists, contact CA Technical Support.

WSA000305

Invalid parameter 'computerIds'

Reason:

You may not have passed computerIds in the DSM Webservice API call.

Action:

You need to pass a valid computerIds. If the problem persists, contact CA Technical Support.

WSA000306

Invalid parameter 'userProfileIds'

Reason:

You may not have passed userProfileIds in the DSM Webservice API call.

Action:

You need to pass a valid userProfileIds. If the problem persists, contact CA Technical Support.

WSA000307

Invalid parameter 'parentUnitGroupId'

Reason:

You may not have passed parentUnitGroupId in the DSM Webservice API call.

Action:

You need to pass a valid parentUnitGroupId. If the problem persists, contact CA Technical Support.

WSA000308

Invalid parameter 'createUnitGroupProperties'

Reason:

You may not have passed createUnitGroupProperties in the DSM Webservice API call.

Action:

You need to pass a valid createUnitGroupProperties. If the problem persists, contact CA Technical Support.

WSA000309

Invalid parameter 'noOfComputers'

Reason:

You may have requested zero or fewer units of Computer.

Action:

You are expected to request one or more units of Computer. If the problem persists, contact CA Technical Support.

WSA000310

Invalid parameter 'unitPropertiesRequired'

Reason:

You may not have passed unitPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid unitPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000311

Invalid parameter 'noOfUnits'

Reason:

You may have requested zero or fewer units of Unit

Action:

You are expected to request one or more units of Unit. If the problem persists, contact CA Technical Support.

WSA000312

Invalid parameter 'queryPropertiesRequired'

Reason:

You may not have passed queryPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid queryPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000313

Invalid parameter 'noOfQueries'

Reason:

You may have requested zero or fewer units of 'Queries'

Action:

You are expected to request more than or at least one unit of Queries. If the problem persists, contact CA Technical Support.

WSA000314

Invalid parameter 'queryId'

Reason:

You may not have passed createSoftwarePackageProperties in the DSM Webservice API call.

Action:

You need to pass a valid createSoftwarePackageProperties. If the problem persists, contact CA Technical Support.

WSA000315

Invalid parameter 'queryResultPropertiesRequired'

Reason:

You may not have passed queryResultPropertiesRequired in the DSM Webservice API call.

Action:

You need to pass a valid queryResultPropertiesRequired. If the problem persists, contact CA Technical Support.

WSA000316

Invalid parameter 'noOfResults'

Reason:

You may have requested zero or fewer units of Query Results.

Action:

You are expected to request one or more units of Query Results. If the problem persists, contact CA Technical Support.

WSA000317

Invalid parameter 'createQueryProperties'

Reason:

You may not have passed createQueryProperties in the DSM Webservice API call.

Action:

You need to pass a valid createQueryProperties. If the problem persists, contact CA Technical Support.

WSA000318

Invalid parameter 'setQueryProperties'

Reason:

You may not have passed setQueryProperties in the DSM Webservice API call.

Action:

You need to pass a valid setQueryProperties. If the problem persists, contact CA Technical Support.

WSA000319

Invalid parameter 'queryIds'

Reason:

You may not have passed queryIds in the DSM Webservice API call.

Action:

You need to pass a valid queryIds. If the problem persists, contact CA Technical Support.

WSA000320

Valid 'passwordbut invalid 'userName'. As unified login is enabled, either both or none should be supplied

Reason:

You may have supplied a password but not userName in the DSMWebservice API call.

Action:

If you supply a valid password you must supply a valid userName.

WSA000321

Valid 'userName' but invalid 'password'. As unified login is enabled, either both or none should be supplied

Reason:

You may have supplied userName but not password in the DSMWebservice API call.

Action:

If you supply a valid userName you must supply a valid password.

WSA000322

Invalid parameter 'manager'

Reason:

You may not have passed manager in the DSM Webservice API call.

Action:

You need to pass a valid manager. If the problem persists, contact CA Technical Support.

WSA000323

Invalid parameter 'setTimeOutValue'

Reason:

You may not have passed a valid value for setTimeOutValue in the DSM Webservice API call.

Action:

You need to pass a valid setTimeOutValue. The permissible range is **60 to 7200** (1 minute to 2 hours). If the problem persists, contact CA Technical Support.

WSA000324

One of the supplied 'unitGroupIds' is not valid

Reason:

You may have passed an invalid unitGroupId.

Action:

You must pass a valid UnitGroupId. If the problem persists, contact CA Technical Support.

WSA000325

One of the supplied 'softwarejobslds' isnot valid

Reason:

You may have passed an invalid softwarejobslds.

Action:

You must pass a valid softwarejobslds. If the problem persists, contact CA Technical Support.

WSA000326

One of the supplied 'unitlds' isnot valid

Reason:

You may have passed an invalid unitlds.

Action:

You must pass a valid unitlds. If the problem persists, contact CA Technical Support.

WSA000327

One of the supplied 'UserAccountlds' is not valid

Reason:

You may have passed an invalid UserAccountlds.

Action:

You must pass a valid UserAccountlds. If the problem persists, contact CA Technical Support.

WSA000328

One of the supplied 'computerlds' isnot valid

Reason:

You may have passed an invalid computerlds.

Action:

You must pass a valid computerlds. If the problem persists, contact CA Technical Support.

WSA000329

One of the supplied 'userProfileIds' isnot valid

Reason:

You may have passed an invalid userProfileIds.

Action:

You must pass a valid userProfileIds. If the problem persists, contact CA Technical Support.

WSA000330

One of the supplied 'softwarePackagelds' isnot valid

Reason:

You may have passed an invalid softwarePackagelds.

Action:

You must pass a valid softwarePackagelds. If the problem persists, contact CA Technical Support.

WSA000331

One of the supplied 'queryIds' isnot valid

Reason:

You may have passed an invalid queryIds.

Action:

You must pass a valid queryIds. If the problem persists, contact CA Technical Support.

WSA000332

One of the supplied 'softwarePackageGroupIds' isnot valid

Reason:

You may have passed an invalid softwarePackageGroupIds.

Action:

You must pass a valid softwarePackageGroupIds. If the problem persists, contact CA Technical Support.

WSA000333

One of the supplied 'softwarePackageProcedureGroupIds' is not valid

Reason:

You may have passed an invalid softwarePackageProcedureGroupIds.

Action:

You must pass a valid softwarePackageProcedureGroupIds. If the problem persists, contact CA Technical Support.

WSA000334

One of the supplied 'softwarePackageProcedureIds' is not valid

Reason:

You may have passed an invalid softwarePackageProcedureIds.

Action:

You must pass a valid softwarePackageProcedureIds. If the problem persists, contact CA Technical Support.

WSA000336

One of the supplied ids is not valid

Reason:

You may have passed an invalid Id.

Action:

You must pass a valid Id. If the problem persists, contact CA Technical Support.

WSA000337

One of the supplied 'softwarePackageVolumelds' is not valid

Reason:

You may have passed an invalid softwarePackageVolumelds.

Action:

You must pass a valid softwarePackageVolumelds. If the problem persists, contact CA Technical Support.

WSA000338

One of the supplied 'softwarePackageProcedureIds' is not valid

Reason:

You may have passed an invalid softwarePackageProcedureIds.

Action:

You must pass a valid softwarePackageProcedureIds. If the problem persists, contact CA Technical Support.

WSA000339

One of the supplied 'softwarePackageVolumeSubpath' is not valid

Reason:

You may have passed an invalid softwarePackageVolumeSubpath.

Action:

You must pass a valid softwarePackageVolumeSubpath. If the problem persists, contact CA Technical Support.

WSA000340

Invalid parameter 'DateTime'

Reason:

You may not have passed DateTime in the DSM Webservice API call.

Action:

You need to pass a valid DateTime. If the problem persists, contact CA Technical Support.

WSA000341

Invalid value for 'month' in parameter 'DateTime'. It should be 1 to 12

Reason:

You may have passed invalid values.

Action:

You must pass values between 1 to 12 for the month parameter.

WSA000342

Invalid value for 'dayOfMonth' in parameter 'DateTime'. It should be 1 to 31

Reason:

You may have passed invalid values.

Action:

You must pass values between 1 to 31 for the dayOfMonth parameter.

WSA000343

Invalid value for 'hour' in parameter 'DateTime'. It should be 0 to 23

Reason:

You may have passed invalid values.

Action:

You must pass values between 0 to 23 for the hour parameter.

WSA000344

Invalid value for 'minute' in parameter 'DateTime'. It should be 0 to 59

Reason:

You may have passed invalid values.

Action:

You must pass values between 0 to 59 for the minute parameter.

WSA000347

Invalid parameter 'userName' or 'password'. You must supply a valid user name and password as unified login is disabled

Reason:

You may not have passed userName or password in WSAPI call.

Action:

You need to pass a valid userName and password if unified login is disabled.

WSA000348

Invalid character(s) in parameter 'password'

Reason:

You may have passed characters that do not conform to base64 encoding algorithm standards.

Action:

Base64 encoding supports both upper and lowercase alphabets, digits ranging from 0 to 9 and + and /.

WSA000349

Invalid character(s) in parameter username

Reason:

You may have passed characters which are not in conformance with base64 encoding algorithm standards

Action:

Base64 encoding supports both upper and lower case alphabets, digits ranging from 0 to 9 and + and /.

WSA000350

Invalid parameter 'volumeName'

Reason:

You may not have passed volumeName in the DSM Webservice API call.

Action:

You need to pass a valid volumeName. If the problem persists, contact CA Technical Support.

WSA000351

Invalid parameter 'volumeld'

Reason:

You may not have passed volumeld in the DSM Webservice API call.

Action:

You need to pass a valid volumeld. If the problem persists, contact CA Technical Support.

WSA000352

Invalid parameter 'procToRun'

Reason:

You may not have passed procToRun in the DSM Webservice API call.

Action:

You need to pass a valid procToRun. If the problem persists, contact CA Technical Support.

WSA000353

Invalid parameter 'jobName'

Reason:

You may not have passed jobName in the DSM Webservice API call.

Action:

You need to pass a valid jobName. If the problem persists, contact CA Technical Support.

WSA000354

Invalid software package group name

Reason:

You may not have passed software package group name in the DSM Webservice API call.

Action:

You need to pass a valid software package group name. If the problem persists, contact CA Technical Support.

WSA000355

Invalid catalog software package group name

Reason:

You may not have passed catalog software package group name in the DSM Webservice API call.

Action:

You need to pass a valid catalog software package group name. If the problem persists, contact CA Technical Support.

WSA000356

Invalid software package procedure group name

Reason:

You may not have passed software package procedure group name in the DSM Webservice API call.

Action:

You need to pass a valid software package procedure group name. If the problem persists, contact CA Technical Support.

WSA000357

Invalid software job container name

Reason:

You may not have passed createSoftwarePackageProperties in the DSM Webservice API call.

Action:

You need to pass a valid createSoftwarePackageProperties. If the problem persists, contact CA Technical Support.

WSA000358

Invalid software policy name

Reason:

You may not have passed software policy name in the DSM Webservice API call.

Action:

You need to pass a valid software policy name. If the problem persists, contact CA Technical Support.

WSA000359

Invalid parameter 'groupUUID'

Reason:

You may not have passed groupUUID in the DSM Webservice API call.

Action:

You need to pass a valid groupUUID. If the problem persists, contact CA Technical Support.

WSA000360

Invalid parameter 'groupLabel'

Reason:

You may not have passed groupLabel in the DSM Webservice API call.

Action:

You need to pass a valid groupLabel. If the problem persists, contact CA Technical Support.

WSA000361

Invalid parameter 'regInfoPath'

Reason:

You may not have passed regInfoPath in the DSM Webservice API call.

Action:

You need to pass a valid regInfoPath. If the problem persists, contact CA Technical Support.

WSA000362

Invalid parameter 'queryName'

Reason:

You may not have passed queryName in the DSM Webservice API call.

Action:

You need to pass a valid queryName. If the problem persists, contact CA Technical Support.

WSA000363

Invalid parameter 'queryTableLabelField'

Reason:

You may not have passed queryTableLabelField in the DSM Webservice API call.

Action:

You need to pass a valid queryTableLabelField. If the problem persists, contact CA Technical Support.

WSA000364

Invalid parameter 'queryTable'

Reason:

You may not have passed queryTable in the DSM Webservice API call.

Action:

You need to pass a valid queryTable. If the problem persists, contact CA Technical Support.

WSA000365

Invalid parameter 'queryTableLabelPrimaryKey'

Reason:

You may not have passed queryTableLabelPrimaryKey in the DSM Webservice API call.

Action:

You need to pass a valid queryTableLabelPrimaryKey. If the problem persists, contact CA Technical Support.

WSA000366

Invalid parameter 'queryFragments'

Reason:

You may not have passed queryFragments in the DSM Webservice API call.

Action:

You need to pass a valid queryFragments. If the problem persists, contact CA Technical Support.

WSA000367

Invalid parameter 'componentIdString'

Reason:

You may not have passed componentIdString in the DSM Webservice API call.

Action:

You need to pass a valid componentIdString. If the problem persists, contact CA Technical Support.

WSA000368

Invalid unit inventory component name

Reason:

You may not have passed createSoftwarePackageProperties in the DSM Webservice API call.

Action:

You need to pass a valid createSoftwarePackageProperties. If the problem persists, contact CA Technical Support.

WSA000369

Invalid parameter 'groupIdString'

Reason:

You may not have passed groupIdString in the DSM Webservice API call.

Action:

You need to pass a valid groupIdString. If the problem persists, contact CA Technical Support.

WSA000370

Invalid unit inventory component name

Reason:

You may not have passed unit inventory component name in the DSM Webservice API call.

Action:

You need to pass a valid unit inventory component name. If the problem persists, contact CA Technical Support.

WSA000371

Invalid UnitIds/UnitGroupIds, at least one of them must be supplied

Reason:

You may not have passed UnitIds and UnitGroupIds in the DSM Webservice API call.

Action:

You need to pass Unitids or UnitGroupIds or both. If the problem persists, contact CA Technical Support.

WSA000372

Invalid parameter 'dependantSoftwarePackageProcedureId'

Reason:

You may not have passed dependantSoftwarePackageProcedureId in the DSM Webservice API call.

Action:

You need to pass a valid dependantSoftwarePackageProcedureId. If the problem persists, contact CA Technical Support.

WSA000392

SessionID cannot be NULL for AM

or

SessionID cannot be NULL for SD

Reason

The Web Service client is trying to authenticate to the asset management or software delivery manager without first authenticating to the common object (CO) manager.

Action

The Web Service client must authenticate to the CO manager before authenticating to the asset management or software delivery manager.

WSA000393

Invalid Authentication Data

Reason

The Web Service client has sent an invalid authentication data.

Action

The Web Service client must verify and send a valid authentication data to Web Services.

WSA000373

Invalid parameter 'createAgentComponents' - At least one agent component must be configured

Reason:

You may not have passed a valid createAgentComponents parameter in the DSM Webservice API call.

Action:

You need to pass a valid createAgentComponents. At least one agent component must be configured on the created computer. If the problem persists, contact CA Technical Support

WSA000376

'unitGroupIds' is a mandatory parameter for Enterprise Manager

Reason:

You may not have passed unitGroupIds in the DSM Webservice API calls.

Action:

unitGroupIds is a mandatory parameter for enterprise manager, You need to pass a valid value for unitGroupIds. If the problem persists, contact CA Technical Support.

WSA000377

No target domain specified. At least one of the parameter, 'addAllDomains', 'domainIds' or 'domainGroupIds' must be supplied

Reason:

You may not have passed all the three; addAllDomains, domainIds and domainGroupIds parameters in the DSM Webservice API call.

Action:

You need to pass at least one of the parameter, addAllDomains, domainIds or domainGroupIds. If the problem persists, contact CA Technical Support

WSA000378

No more objects to read

Reason:

You may be trying to read objects from a list API while there are no more objects available to read.

Action:

Use fromIndex to read objects anywhere from the list. If the problem persists, contact CA Technical Support.

WSA000379

At least one of the scan target options need to be specified

Reason:

None of the scan target options i.e. targetMachines, domain name, IP address range, query name and URI were supplied

Action:

Please pass in one of the scan target options.

WSA000380

EndIP address cannot be specified without a valid start IP address.

Reason:

End IP address was specified without specifying a valid start IP address.

Action:

Specify a valid start IP address along with an end IP address.

WSA000381

Target machines option cannot be combined with other parameters.

Reason:

Target machines parameter was specified along with other scan target option which is not valid.

Action:

If target machines are supplied, it should not be combined with other scan target options.

WSA000382

Domain name cannot be combined with URI, Query and TargetMachines parameters.

Reason:

Domain name was specified along with either URI, Query name or Target machines options.

Action:

If domain name is supplied, it can be combined only with IP address range (start and end IP) and not with other scan target options.

WSA000383

IPRange (StartIP and EndIP) cannot be combined with URI, Query and TargetMachines parameters.

Reason:

IP range was specified along with either URI, Query name or Target machines options.

Action:

If IP range is supplied, it can be combined only with IP address range (start and end IP) and not with other scan target options.

WSA000384

Query name cannot be combined with other scan parameters.

Reason:

Query name was specified along with other scan target options which is not valid.

Action:

If query name is supplied, it should not be combined with other scan target options.

WSA000385

URI cannot be combined with other scan parameters.

Reason:

URI was specified along with other scan target options which is not valid.

Action:

If URI is supplied, it should not be combined with other scan target options

WSA000386

Either default targetuser credentials or targetMachine credentials should be supplied.

Reason:

Target user name, password and targetMachine credentials were both not supplied.

Action:

Pass in the default target user name and password. If they don't have access to all the target machines, then additionally you can pass in the machine specific credentials in targetMachineCredentials.

WSA000387

If the target user name is supplied then the target password should also be supplied.

Reason:

Target user name was supplied but the password was omitted.

Action:

Pass in a valid password for the target user.

WSA000388

If deploying from a scalability server, then scalability server username and password should be supplied.

Reason:

Scalability server username and password were omitted while choosing the option of deploying from scalability server.

Action:

Pass in valid values for scalability server user name and password.

WSA000389

Either targetuser credentials or targetMachines credentials should be supplied for staging.

Reason:

Target user name, password and targetMachine credentials were both not supplied.

Action:

Pass in the default target user name and password. If they don't have access to all the target machines, then additionally you can pass in the machine specific credentials in targetMachineCredentials.

WSA000390

DeploymentPackageFromSS and ScanStagedPackages flags cannot be true at the same time.

Reason:

DeploymentPackageFromSS and ScanStagedPackages were both set to true.

Action:

Either one of them should be false

WSA001001

Invalid parameter <parameter name>

Reason:

You may not have passed mandatory <parameter name> in the DSM Webservice API call.

Action:

You need to pass a valid value to the mandatory parameters. If the problem persists, contact CA Technical Support.

WSA001002

One of the supplied <parameter name> is not valid

Reason:

You may not have passed valid value/s for any/all elements in the array <parameter name> in the DSM Webservice API call.

Action:

You need to pass a valid value for all the supplied elements of the array parameter. If the problem persists, contact CA Technical Support.

HLA000008

Unable to allocate memory

Reason:

Memory could not be allocated.

Action:

Restart DSM Webservice. If problem persists, contact CA Technical Support.

HLA000201

Failed the create the common object (CO) component

Reason:

DSM Webservice could not create CO component.

Action:

Restart CAF. If problem persists, contact CA Technical Support.

HLA000202

Failed the create the common configuration (CCNF) component

Reason:

DSM Webservice could not create CCNF component.

Action:

Restart CAF. If problem persists, contact CA Technical Support.

HLA000204

Failed to create asset management (AM) component

Reason:

DSM Webservice could not create AM component.

Action:

Restart CAF. If problem persists, contact CA Technical Support.

HLA000205

Security Authority is not provided in 'userName'

Reason:

You may not have passed Security Authority in the user name for the Webservice Login method. Username must be of the form :

<securityauthority>://<SecurityProvider>/username

For more details, see the Web Services Reference.

Action:

You need to pass Security Authority like winnt, unixl in the user name. If problem persists, contact CA Technical Support.

HLA000206

User name is not supplied

Reason:

You may not have passed user name but security authority in the Webservice Login method. Username must be of the form :

<securityauthority>://<SecurityProvider>/username

For more details, see the Web Services Reference.

Action:

You need to pass username along with security authority. If problem persists, contact CA Technical Support.

HLA000207

Failed to create XML parser component

Reason:

DSM Webservice could not create XML parser component.

Action:

Restart CAF. If problem persists, contact CA Technical Support.

HLA000208

Failed to initialize XML parser component

Reason:

DSM Webservice could not initialize XML parser component.

Action:

Restart CAF. If problem persists, contact CA Technical Support.

HLA000209

More than one object found

Reason:

You may have passed an object (name) in the DSM Webservice API call which is not unique, so the intended operation can not be performed.

Action:

You need to pass a unique object (name) or use UUID version of the API. If problem persists, contact CA Technical Support.

HLA000210

Scalability/Boot server not found

Reason:

You may have passed a scalability or boot server name in the DSM Webservice API call which doesn't exist.

Action:

You need to pass a valid scalability/boot server name. If problem persists, contact CA Technical Support.

HLA000211

Computer not found

Reason:

You may have passed a computer name in the DSM Webservice API call which doesn't exist.

Action:

You need to pass a valid computer name. If problem persists, contact CA Technical Support.

HLA000212

SD Agent is not configured on the computer

Reason:

You may have passed a computer in which SD agent is not configured, so the intended operation can not be performed.

Action:

You need to pass a valid computer on which SD agent is configured. If problem persists, contact CA Technical Support.

HLA000213

User profile not found

Reason:

You may have passed user profile name in the DSM Webservice API call which doesn't exist.

Action:

You need to pass a valid user profile name. If problem persists, contact CA Technical Support.

HLA000214

User profile name supplied is of invalid format

Reason:

You may have passed user profile name in the DSM Webservice API call which is not in required format.

Action:

You need to pass user profile name in required format. Check the API documentation for more information. If the problem persists, contact CA Technical Support.

HLA000215

Query not found

Reason:

You may have passed a query name that does not exist in a DSM Webservice API call.

Action:

You need to pass a valid query name. If the problem persists, contact CA Technical Support.

HLA000216

Engine not found

Reason:

You may have passed engine name in the DSM Webservice API call which doesn't exist.

Action:

You need to pass a valid engine name. If problem persists, contact CA Technical Support.

HLA000217

Unit group not found

Reason:

You may have passed unit group name in the DSM Webservice API call which doesn't exist.

Action:

You need to pass a valid unit group name. If problem persists, contact CA Technical Support.

HLA000218

Job not found

Reason:

You may have passed reference of a software job in the DSM Webservice API call that doesn't exist.

Action:

You need to pass a valid software job reference. If problem persists, contact CA Technical Support.

HLA000219

No jobs found made with given software package

Reason:

You may have passed reference of a software package in the DSM Webservice API call that is not deployed.

Action:

None

HLA000220

Failed to parse XML document

Reason:

AM Manager responded with invalid XML Document and parsing is failed.

Action:

Restart CAF. If the problem persists, contact CA Technical Support.

HLA000221

Invalid XML Document

Reason:

AM Manager responded with invalid XML Document.

Action:

Restart CAF. If the problem persists, contact CA Technical Support.

HLA000222

No unit inventory components found

Reason:

You may have passed unit that doesn't have any unit inventory components.

Action:

None

HLA000223

No unit inventory items found

Reason:

You may have passed unit that doesn't have any unit inventory items.

Action:

None

HLA000224

No unit inventory groups found

Reason:

You may have passed unit that doesn't have any unit inventory groups.

Action:

None

HLA000225

No discovered softwares found

Reason:

You may have passed unit that doesn't have any discovered softwares.

Action:

None.

HLA000226

No query results found

Reason:

You may have passed query that doesn't have any query results.

Action:

None

HLA000227

No queries found

Reason:

Currently no queries available in the manager.

Action:

None

HLA000228

Failed to get the query result list

Reason:

AM Manager responded with invalid query results XML Document

Action:

Restart CAF. If the problem persists, contact CA Technical Support.

HLA000229

Failed to get the discovered software list

Reason:

AM Manager responded with invalid discovered software XML Document

Action:

Restart CAF. If the problem persists, contact CA Technical Support.

HLA000230

Failed to get the query list

Reason:

AM Manager responded with invalid queries XML Document

Action:

Restart CAF. If the problem persists, contact CA Technical Support.

HLA000231

No software groups found

Reason:

Currently no software groups available on the manager

Action:

None

HLA000232

Could not find target directory path

Reason:

Directory path provided by user is not present on the target system

Action:

You need to pass valid target directory path. If problem persists, contact CA Technical Support.

HLA000233

Failed to copy file target directory path

Reason:

Failed to copy file to the target location.

Action:

Restart CAF. If problem persists, contact CA Technical Support.

HLA000234

No software volume found in the given software package

Reason:

The software package does not contain any software package volume.

Action:

You need to pass software package which contains software volumes. If problem persists, contact CA Technical Support.

HLA000235

Invalid software package volume source path

Reason:

The software package volume source path is empty or it is invalid.

Action:

You need to pass valid software package volume source path.

HLA000236

Invalid security profile handle

Reason:

The handle return by web service api is invalid.

Action:

Restart CAF. If problem persists, contact CA Technical Support.

HLA000237

End of list reached for security profile or security class or security areas

Reason:

There are no more profiles or class or areas in the list.

Action:

None

HLA000238

Invalid number of total security classes

Reason:

Number of security classes requested is more than total number of security classes available.

Action:

You need to enter number which is less than or equal to the total number of security classes.

HLA000239

No Security Areas found on the system

Reason:

Currently, there are no security areas available on the system.

Action:

None

HLA000240

Invalid security profile or security class UUID

Reason:

Currently, there is no security profile or security class available on the machine matching to the given UUID.

Action:

None

HLA000241

Built-in security profile

Reason:

This error code is returned because built-in security profile can not be modified or deleted.

Action:

You need to enter non built-in security profile. If problem persists, contact CA Technical Support.

HLA000242

Invalid security profile name

Reason:

The format of security profile entered is wrong.

Action:

You need to enter security profile in valid format. If problem persists, contact CA Technical Support.

HLA000243

Invalid operation on the enterprise manager

Reason:

Security area functionality is not supported on enterprise manager. It is only supported on domain manager.

Action:

If you are executing security area functionality, then please run it only on domain manager.

HLA000244

Security Area not found on the system

Reason:

One of the security areas entered by the user is not present on the system

Action:

You need to enter valid security area names.

HLA000245

No deployment packages found on the Deployment manager

Reason:

There might be a timeout or no packages listed there. Any problem on the manager would also return this error.

Action:

Restart DMDeploy service on the manager and retry the operation.

HLA000246

Multiple matching packages found for the given input.

Reason:

Returned during scan or deploy operation. This happens when the package information passed in is not sufficient to uniquely identify a package.

Action:

Try to pass in the package information so that a package is uniquely identifiable on the manger.

HLA000247

Validation of package parameters failed, Failed Params :.

Reason:

Returned during the deploy operation. This happens when insufficient or incorrect values are passed for the package parameters.

Action:

Check the failed parameters and pass a valid value for them.

AWS00052

Invalid manager. Verify whether the default manager is passed.

Reason:

Automation Services are supported only on default manager. Verify whether you have specified the default manager.

Action:

Specify the default manager name.

AWS00053

Invalid user name or password passed.

Reason:

Either the user name or the password is null or empty.

Action:

Specify a valid user name and password.

AWS00054

Object which you are trying to delete is successfully deleted.

Reason:

Object which you are trying to delete is successfully deleted.

Action:

NA

AWS00055

Invalid value for provided for the target name.

Reason:

The specified target name is either null or contains an empty string.

Action:

Specify a valid target name.

AWS00103

Session not found. Please login again using ITCM Web Service login.

Reason:

The specified session ID does not exist or is invalid.

Action:

Specify a valid session ID. Execute the CA ITCM Web Service login method to get a valid session ID.

AWS00104

Session already exists with this session ID.

Reason:

Automation Services are initialized by some session ID. You are trying to initialize the automation services that are already initialized.

Action:

Do not initialize if you have already initialized the automation services.

AWS00105

ITCM session validation failed. Verify ITCM session.

Reason:

CA ITCM WS does not have a session with the specified session ID.

Action:

Specify a valid session ID.

AWS00201

Invalid Assessment Template

Reason:

You may have passed null for the Assessment template object.

Action:

Specify a valid Assessment template object.

AWS00202

Invalid Assessment Template Name

Reason:

You may have passed null or empty string for the Assessment template name.

Action:

Specify a valid assessment template name.

AWS00203

Invalid Assessment Template Parameters

Reason:

You may have passed null for the assessment template parameters.

Action:

Specify a valid assessment template parameters.

AWS00204

Invalid Assessment Template Type

Reason:

You may have passed null for the assessment template type.

Action:

Specify a valid assessment template type.

AWS00205

Invalid Assessment Task Name

Reason:

You may have passed null or empty string for the assessment task name.

Action:

Specify a valid assessment task name.

AWS00206

Invalid Group Name

Reason:

You may have passed null or empty string for the group name.

Action:

Specify a valid group name.

AWS00207

Assessment template creation failed

Reason:

The assessment template creation failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00208

Assessment template updation failed

Reason:

Assessment template update failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00209

Some or all of the requested assessment templates deletion failed

Reason:

One or more assessment templates you specified do not exist.

Action:

Verify that the templates do exist. Verify the error details for more information and take appropriate action.

AWS00210

Get Assessment Template Failed

Reason:

The assessment template details cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00211

Invalid list of Assessment Template Names

Reason:

You may have provided null or empty list for assessment template names list.

Action:

Specify a valid list of assessment template names.

AWS00213

Get Assessment Status Failed

Reason:

The assessment status cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00214

Execute Assessment Task failed

Reason:

The assessment task execution failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00215

Get Assessment Result failed

Reason:

The assessment result cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00216

Get Assessment Task Failed

Reason:

The assessment task cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00217

Invalid list of Assessment Task Names

Reason:

You may have provided null or empty list for assessment task names list.

Action:

Specify a valid list of assessment task names.

AWS00218

Some or all of the requested Assessment Tasks Deletion Failed

Reason:

One or more assessment tasks you specified do not exist.

Action:

Verify that the tasks do exist. Verify the error details for more information and take appropriate action.

AWS00219

Assessment Task Deletion Failed

Reason:

The assessment task deletion Failed.

Action:

Specify a valid assessment task name. Verify the error details for more information and take appropriate action.

AWS00220

Assessment Template Deletion Failed

Reason:

The assessment template deletion Failed.

Action:

Specify a valid assessment template name. Verify the error details for more information and take appropriate action.

AWS00221

Invalid Assessment Task

Reason:

You may have passed null for assessment task object.

Action:

Specify a valid assessment task object.

AWS00222

Assessment task creation failed

Reason:

The assessment task creation failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00223

Invalid value provided for the field Operating Systems

Reason:

You may have passed null or empty list for the field Operating Systems Array.

Action:

Specify valid Operating Systems array.

AWS00301

Invalid Osim Template

Reason:

You may have passed null for OSIM template object.

Action:

Specify a valid OSIM template object.

AWS00302

Invalid Osim Template Name

Reason:

You may have passed null or empty string for OSIM template name.

Action:

Specify a valid OSIM template name.

AWS00303

Invalid Osim Template Type

Reason:

You may have passed null for OSIM template type.

Action:

Specify a valid OSIM template type.

AWS00304

Invalid hardware model Type

Reason:

You may have passed null for hardware model type or have passed an empty list for hardware models.

Action:

Specify a valid hardware model type object.

AWS00305

Invalid Global Type

Reason:

You may have passed null for global model type.

Action:

Specify a valid global model type object.

AWS00306

Invalid group model Type

Reason:

You may have passed null for group model type or have passed an empty list for group models.

Action:

Specify a valid group model type object.

AWS00307

Invalid Image Type

Reason:

You may have passed null for image type.

Action:

Specify a valid image type object.

AWS00308

Invalid Image Name

Reason:

You may have passed null or empty string for image name.

Action:

Specify a valid image name.

AWS00309

Invalid Hard Ware Model Name

Reason:

You may have passed null or empty string for hardware model name.

Action:

Specify a valid hardware model name.

AWS00310

Osim Template Creation Failed

Reason:

The OSIM template creation failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00311

Osim Template Updation Failed

Reason:

The OSIM template update failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00312

Osim Template Deletion Failed

Reason:

The OSIM template deletion failed.

Action:

Specify a valid OSIM template name. Verify the error details for more information and take appropriate action.

AWS00316

Some or all of the requested Osim Templates Deletion Failed

Reason:

One or more OSIM templates you specified do not exist.

Action:

Verify that the templates do exist. Verify the error details for more information and take appropriate action.

AWS00313

Invalid list of Osim Template Names

Reason:

You may have provided null or empty list for OSIM template names list.

Action:

Specify a valid list of OSIM template names.

AWS00314

Get Osim Template Failed

Reason:

The OSIM template cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00401

Invalid Software Deployment Template

Reason:

You may have passed null for Software Deployment template object.

Action:

Specify a valid software deployment template object.

AWS00402

Invalid Software Deployment Template Name

Reason:

You may have passed null or empty string for software deployment template name.

Action:

Specify a valid software deployment template name.

AWS00404

Invalid Software Package Mapping Type

Reason:

You may have passed null for Software Package Mapping Type.

Action:

Specify a valid Software Package Mapping type object.

AWS00406

Invalid Pre Install Jobs

Reason:

You may have passed null for Software Job Container Type Object for pre-install jobs.

Action:

Specify a valid Software Job Container Type Object.

AWS00407

Invalid Post Install Jobs

Reason:

You may have passed null for Software Job Container Type Object for post-install jobs.

Action:

Specify a valid Software Job Container Type Object.

AWS00408

Software Deployment Template Creation Failed

Reason:

The software deployment template creation failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00409

Software Deployment Template updation Failed

Reason:

The software deployment template update failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00410

Software Deployment Template Deletion Failed

Reason:

The software deployment template deletion failed.

Action:

Specify a valid software deployment template name. Verify the error details for more information and take appropriate action.

AWS00411

Invalid list of Software Deployment Template Names

Reason:

You may have provided null or empty list for software deployment template names list.

Action:

Specify a valid list of software deployment template names list.

AWS00412

Get Software Deployment Template Failed

Reason:

The software deployment template cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00414

Some or all of the requested Software Deployment Templates Deletion Failed

Reason:

One or more software deployment templates you specified do not exist.

Action:

Verify that the templates do exist. Verify the error details for more details and take appropriate action.

AWS00601

Invalid Automation Process

Reason:

You may have passed null for automation process object.

Action:

Specify a valid automation process object.

AWS00602

Invalid Automation Process Name

Reason:

You may have passed null or empty string for automation process name.

Action:

Specify a valid automation process name.

AWS00603

Invalid Automation Process Mode

Reason:

You may have passed null for automation process mode object.

Action:

Specify a valid automation process mode object.

AWS00604

Create Automation Process Failed

Reason:

The automation process creation failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00605

Execute Automation Process Failed

Reason:

The automation process execution failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00607

Get Activity Log Failed

Reason:

The activity log cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00608

Set Activity Status failed

Reason:

Set Activity Log failed.

Action:

Verify the error details for more information and take appropriate action.

AWS00609

Invalid list of Automation Process Names

Reason:

You may have provided null or empty list for automation process names list.

Action:

Specify a valid list of automation process names list

AWS00610

Automation Process Deletion Failed

Reason:

The automation process deletion failed.

Action:

Specify a valid Automation Process name. Verify the error details for more information and take appropriate action.

AWS00611

Some or all of the requested Automation Processes Deletion Failed

Reason:

One or more Automation Processes you specified do not exist.

Action:

Verify that the Automation Processes do exist. Verify the error details for more details and take appropriate action.

AWS00612

Get Automation Result failed

Reason:

The automation result cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00613

Get Automation Process Status Failed

Reason:

The automation process status cannot be retrieved.

Action:

Verify the error details for more information and take appropriate action.

AWS00701

Invalid Automation Job ID

Reason:

You may have passed null or empty string for automation job ID.

Action:

Specify a valid automation job ID.

AWS00702

Invalid Automation Machine ID

Reason:

You may have passed null or empty string for automation machine ID.

Action:

Specify a valid Automation machine ID.

AWS00703

Invalid software container data

Reason:

You may have passed null for software container data object.

Action:

Specify a valid software container data object.

AWS00704

Invalid software job data

Reason:

You may have passed null for software job data object.

Action:

Specify a valid software job data object.

WSA000394

Invalid parameter 'osimTargetMACAddress'

Reason:

The specified MACAddress is invalid.

Action:

Verify that you have provided the correct MACAddress.

WSA000395**Invalid parameter 'osimTargetLabel'****Reason:**

The value specified for the OSIM computer label is invalid.

Action:

Verify that you have provided the correct computer label.

WSA000396**Invalid parameter 'OSImage'****Reason:**

The value specified for OSImage is invalid.

Action:

Verify that you have provided the correct OSImage name.

WSA000397**Invalid parameter 'Bootserver'****Reason:**

The value specified for Bootserver is invalid.

Action:

Verify that you have provided the correct Bootserver name.

WSA000398**Invalid Filter Condition****Reason:**

The values specified for filter properties are invalid.

Action:

Verify that you have specified proper filter conditions, filter properties, and filter values. For more information, see the sequences related to computer filters.

WSA000399

Invalid value for 'seconds' in parameter 'DateTime'. It should be 0 to 59

Reason:

The value specified for seconds is invalid.

Action:

Verify that you have provided the correct seconds in DateTime2 sequence.

HLA000259

Invalid Query Folder Id. Query folder doesn't exist for the specified folder id.

Reason:

No CA ITCM query folder exists with the specified folder ID.

Action:

Specify a valid folder ID.

HLA000260

Invalid Folder name. Query folder doesn't exist for the specified name.

Reason:

No CA ITCM query folder exists with the specified name.

Action:

Specify a valid query folder name.

WSA000408

Invalid value for List Index.

Reason:

You may have entered a value less than zero.

Action:

Specify a value that is equal to or more than zero.

WSA000409

Invalid Computer classID.**Reason:**

The specified class ID does not exist.

Action:

Specify a valid computer class ID.

WSA000410

Invalid value for number of objects required.**Reason:**

You may have entered a value equal to or less than zero.

Action:

Specify a value that is more than zero.

WSA000400

Invalid Asset Job Properties Supplied**Reason:**

The asset job properties you have specified are invalid.

Action:

Verify that you have provided the correct values for the properties. For more information, see [AssetScriptJobProperties](#) (see page 788).

WSA000401

Invalid Asset Job UUID or Asset Job Name Supplied

Reason:

The asset job UUID or the job name you have specified is not available in the database.

Action:

Verify that you have provided the correct UUID or asset job name.

WSA000402

Invalid Asset Job Properties required

Reason:

The asset job properties you are trying to retrieve do not exist.

Action:

Verify that you have provided the correct property name; check to see if there is any typo. For more information, see [AssetScriptJobProperties](#) (see page 788).

WSA000403

Number of Asset Script jobs can not be less than or equal to zero

Reason:

The list handle you have specified does not contain any asset jobs.

Action:

Verify that you have opened the asset scripts jobs and have provided the correct list handle. For more information, see [OpenAssetScriptJobList](#) (see page 612) and [GetAssetScriptJobs](#) (see page 613).

WSA000404

Invalid value supplied for periodic dependency condition

Reason:

The value you have specified for periodic dependency is invalid.

Action:

Verify that you have provided correct values for periodic dependency. For more information, see [PeriodicDependency](#) (see page 794).

WSA000405

Invalid value supplied for date time range

Reason:

The date and time range you have specified is invalid.

Action:

Specify the correct date and time range. For more information, see [DateTimeRestrictions](#) (see page 792).

WSA000406

Invalid value supplied for unit type

Reason:

The unit type you have specified does not exist.

Action:

Verify that the value you have provided matches [CO_TRIGGER_UNIT_TYPE](#) (see page 110).

Index

A

- AbortDeployment • 159
- AbortDeploymentJob • 159
- ActionType • 134
- ActivateJobCheck • 617
- ActivateOSInstallation • 450
- ActivateOSInstallationByName • 451
- ActivateSoftwareJobContainer** • 212
- ActivityLog • 975
- ActivityLogResponse • 975
- AddDeregisterSoftwareJobContainerOrdersToDistributionContainer • 579
- AddDeregisterSoftwarePackageOrdersToDistributionContainer • 576
- AddDeregisterSoftwarePackageProcedureOrdersToDistributionContainer • 576
- AddDeregisterSoftwarePolicyOrdersToDistributionContainer • 577
- AddFetchItemJobOrdersToDistributionContainer • 578
- AddRegisterSoftwareJobContainerOrdersToDistributionContainer • 578
- AddRegisterSoftwarePackageOrdersToDistributionContainer • 580
- AddRegisterSoftwarePackageProcedureOrdersToDistributionContainer • 580
- AddRegisterSoftwarePolicyOrdersToDistributionContainer • 581
- AdhocTargetEvaluation • 48
- AgentRestrictions • 123
- AMAgentStatus • 119
- AMExtraCmdParam • 110
- Archive Path • 48
- ArchivePath • 48
- Array of Elements • 980
- ArrayOf_xsd_string • 980
- ArrayOfActivityLog • 978
- ArrayOfAMExtraCmdParam • 993
- ArrayOfAssessmentTaskInfo • 978
- ArrayOfAssessmentTemplate • 978
- ArrayOfAutomationProcess • 977
- ArrayOfAutomationProcessBasicInfo • 978
- ArrayOfAutomationProcessListDetailed • 977
- ArrayOfAutomationProcessTargetStatus • 976, 978
- ArrayOfAutomationWebServiceException • 977
- ArrayOfBootImageFilter • 985
- ArrayOfBootImageProperties • 984
- ArrayOfBootServerFilter • 984
- ArrayOfBootServerProperties • 476, 477, 478, 984, 995
- ArrayOfComponents • 987, 990
- ArrayOfComponentStatus • 987, 990
- ArrayOfComputerProperties2 • 530, 532, 988, 990, 996
- ArrayOfComputerPropertyFilter2 • 996
- ArrayOfConfigurationParams • 988, 990
- ArrayOfConfigurationParamValues • 988, 990
- ArrayOfCustomizationParameter • 980
- ArrayOfDayOfTheWeekDependency • 993
- ArrayOfGetAssetScriptJobsProperties • 993
- ArrayOfGroupType • 978
- ArrayOfHardWareModelType • 979
- ArrayOfHWModelFilter • 989, 991
- ArrayOfItems • 996
- ArrayOfOperatingSystemType • 980
- ArrayOfOSImageFilter • 469, 480, 995
- ArrayOfOSImageProperties • 470, 471, 995, 996
- ArrayOfOSIMImageProperties • 994
- ArrayOfOSIMIRParametersWithDesc • 994
- ArrayOfOSIMparameterDesc • 994
- ArrayOfOSIMProperties • 461, 462, 992, 993, 994, 995
- ArrayOfOSIMPropertiesWithDesc • 475, 994
- ArrayOfOSIMTargetFilter • 995
- ArrayOfOSIMTargetInstallationRequestsProperties • 463, 464, 993
- ArrayOfOSIMTargetProperties • 459, 460, 992
- ArrayOfOSIMTargetPropertiesWithDesc • 467, 468, 994
- ArrayOfOsimTemplate • 979
- ArrayOfPackageParameters • 996
- ArrayOfParameter • 959
- ArrayOfParameterInfo arrayOfParamInfo • 996
- ArrayOfQueries • 989, 991
- ArrayOfQueryFilter • 988, 991
- ArrayOfQueryFolderProperties • 989, 991
- ArrayOfSDTemplateProperties • 992
- ArrayOfSecurityAreaByName • 987
- ArrayOfSecurityAreasByName • 987

ArrayOfSecurityClassPermissions • 989	AutomationProcess • 968
ArrayOfSecurityClassPermissionsForProfiles • 989	AutomationProcessBasicInfo • 971
ArrayOfSecurityProfilesSecurityGroupsPermissions • 982, 989	AutomationProcessFilter • 969
ArrayOfSecurityProfilesSecurityGroupsPermissions • 989	AutomationProcessListDetailed • 970
ArrayOfServerProperties • 474, 995	AutomationProcessMode • 132
ArrayOfSoftwareDeploymentTemplate • 979	AUTOMATIONPROCESSPROPERTY • 132
ArrayOfSoftwareJob • 983	AutomationProcessStatus • 132
ArrayOfSoftwareJobProperties3 • 983	AutomationProcessTargetStatus • 976
ArrayOfSoftwarePackageFilter • 992	AutomationResultFilter • 976
ArrayOfSoftwarePackageFilter3 • 986	AutomationResultProperty • 133
ArrayOfSoftwarePackageMappingType • 979	AutomationResultResponse • 974
ArrayOfSoftwarePackageProcedureProperties2 • 984	AutomationTargetStatus • 133
ArrayOfSoftwarePackageProcedureProperties4 • 985	AutomationWebServiceException • 976
ArrayOfSoftwarePackageProperties2 • 985, 992	AutomationWebServiceExceptionList • 977
ArrayOfSoftwarePackageProperties3 • 986	AVAgentStatus • 122
ArrayOfSoftwarePackageType • 979	AWS00052 • 1068
ArrayOfstring • 997	AWS00053 • 1068
ArrayOfTargetProperties • 481, 995	AWS00054 • 1069
ArrayOfTargetProperties2 • 983	AWS00055 • 1069
ArrayOfTemplateProperties • 979	AWS00103 • 1069
ArrayOfTriggerInfo • 993	AWS00104 • 1070
ArrayOfUnitSoftwareJobProperties2 • 987	AWS00105 • 1070
ArrayOfUnitSoftwareJobProperties3 • 984	AWS00201 • 1070
ASPAgentStatus • 122	AWS00202 • 1071
AssessmentResult • 961	AWS00203 • 1071
AssessmentStatus • 130	AWS00204 • 1071
AssessmentTask • 959	AWS00205 • 1071
AssessmentTaskFilter • 962	AWS00206 • 1072
AssessmentTaskInfo • 962	AWS00207 • 1072
ASSESSMENTTASKPROPERTY • 129	AWS00208 • 1072
AssessmentTemplate • 958	AWS00209 • 1073
AssessmentTemplateFilter • 960	AWS00210 • 1073
AssessmentTemplateParameterDataType • 129	AWS00211 • 1073
AssessmentTemplateParameterName • 128	AWS00213 • 1074
ASSESSMENTTEMPLATEPROPERTY • 129	AWS00214 • 1074
AssessmentTemplateType • 128	AWS00215 • 1074
Asset Script Job Methods • 610	AWS00216 • 1074
AssetScriptJobProperties • 788, 1091, 1092	AWS00217 • 1075
AssetScriptJobPropertiesRequired • 796	AWS00218 • 1075
AssignOSImageToOSIMTargets • 480	AWS00219 • 1075
AssignOSIMTargetsToBootServer • 472	AWS00220 • 1076
AssignOSIMTargetsToBootServerByName • 472	AWS00221 • 1076
AttributeMask • 631	AWS00222 • 1076
Authentication and Web Services Management • 135	AWS00223 • 1077
AuthenticationData • 631	AWS00301 • 1077
Automation Web Services End Point • 45	AWS00302 • 1077
	AWS00303 • 1077
	AWS00304 • 1078
	AWS00305 • 1078

AWS00306 • 1078
AWS00307 • 1079
AWS00308 • 1079
AWS00309 • 1079
AWS00310 • 1079
AWS00311 • 1080
AWS00312 • 1080
AWS00313 • 1081
AWS00314 • 1081
AWS00316 • 1080
AWS00401 • 1081
AWS00402 • 1081
AWS00404 • 1082
AWS00406 • 1082
AWS00407 • 1082
AWS00408 • 1082
AWS00409 • 1083
AWS00410 • 1083
AWS00411 • 1083
AWS00412 • 1084
AWS00414 • 1084
AWS00601 • 1084
AWS00602 • 1084
AWS00603 • 1085
AWS00604 • 1085
AWS00605 • 1085
AWS00607 • 1085
AWS00608 • 1086
AWS00609 • 1086
AWS00610 • 1086
AWS00611 • 1087
AWS00612 • 1087
AWS00613 • 1087
AWS00701 • 1087
AWS00702 • 1088
AWS00703 • 1088
AWS00704 • 1088

B

Base64 Encoding Algorithm • 146
BootImageFilter • 950
BootImageProperties • 950
BootImageProperty • 49
BootServerFilter • 953
BootServerProperties • 951
BootServerProperty • 48

C

CA Technologies Product References • 3
CancelOSInstallation • 452
CancelOSInstallationByName • 453
CancelSoftwareJobs • 242
CFComponentCode • 110, 798
Choosing a Target • 37
CloseAssetScriptJobList • 614
CloseDistributionContainerDistributionDomainList • 582
CloseDistributionContainerList • 582
CloseDistributionContainerOrders • 583
CloseDomainGroupDomainGroupMembersList • 598
CloseDomainGroupDomainMembersList • 597, 598
CloseOSImageList • 470
CloseOSIMTargetList • 482
CloseParentSoftwareJobContainerList2 • 428
CloseParentSoftwarePackageGroupList • 289
CloseParentSoftwarePackageJobContainerList • 213
CloseParentSoftwarePackageProcedureGroupList • 352
CloseQueryList • 166
CloseQueryResultList • 166, 212
CloseScalabilityServerList • 474
CloseSecurityAreaList • 204
CloseSecurityClassList • 185
CloseSecurityProfileList • 182
CloseSoftwareJobContainerList • 212
CloseSoftwareJobContainerList2 • 213
CloseSoftwareJobContainerListRecursive • 214
CloseSoftwareJobContainerListRecursive2 • 429
CloseSoftwareJobContainerSoftwareJobList • 214
CloseSoftwareJobList • 242
CloseSoftwareJobUnitSoftwareJobL • 243
CloseSoftwareJobUnitSoftwareJobList • 243
CloseSoftwarePackageGroupList • 289
CloseSoftwarePackageGroupListRecursive • 290
CloseSoftwarePackageGroupSoftwarePackageList • 290
CloseSoftwarePackageGroupSoftwarePackageListRecursive • 291
CloseSoftwarePackageInstallationList • 378
CloseSoftwarePackageList • 378
CloseSoftwarePackageProcedureDependencyList • 311
CloseSoftwarePackageProcedureGroupList • 352
CloseSoftwarePackageProcedureGroupListRecursive • 290

CloseSoftwarePackageProcedureGroupSoftwarePackageProcedureList • 246, 316, 353
CloseSoftwarePackageProcedureList • 317
CloseSoftwarePackageVolumeElementList • 368
CloseSoftwarePackageVolumeList • 369
CloseSoftwarePolicyList • 399
CloseSoftwarePolicyList2 • 429
CloseSoftwarePolicyListRecursive • 399
CloseSoftwarePolicySoftwareJobList • 400
CloseSoftwarePolicySoftwarePolicyExceptionList • 400
CloseUnitDiscoveredSoftwareList • 544
CloseUnitGroupComputerMembersList • 500
CloseUnitGroupUnitGroupMembersList • 500
CloseUnitGroupUnitMembersList • 501
CloseUnitGroupUserAccountMembersList • 501
CloseUnitGroupUserProfileMembersList • 502
CloseUnitInventoryComponentGroupList • 534
CloseUnitInventoryComponentList • 535
CloseUnitInventoryGroupList • 535
CloseUnitInventoryItemList • 536
CloseUnitQueryList • 544
CloseUnitUnitGroupList • 545
CloseUnitUnitSoftwareInstallationJobList • 545
CloseUnitUnitSoftwareJobList • 546
Closing the Session • 24, 41
CO_TRIGGER_UNIT_TYPE • 110, 1093
CommonSoftwarePackageProcedureProperties3 • 651
Complex Attribute Types • 26
COMPONENTS • 126
COMPONENT-STATUS • 126
CompRequiresAuthentication • 68
computerClassID • 50, 655, 678, 894
ComputerFilterCondition • 118
ComputerProperties • 669, 671
ComputerProperties2 • 531, 626, 809, 953
ComputerPropertiesRequired • 670
ComputerPropertiesRequired2 • 531, 806
ComputerProperty • 66
ComputerProperty2 • 115
ComputerPropertyFilter • 671, 980
ComputerPropertyFilter2 • 803
ComputerStatusInfo • 67
ComputerStatusInfoRequired • 673
Conclusions • 42
Conditions • 790
Configuration Parameters • 997
ConfigurationParams • 126

Considerations for Using OSIM Web Services • 449
Contact CA Technologies • 4
CopyElementsToSoftwarePackageVolume • 369
CopyElementsToSoftwarePackageVolume2 • 370
CreateActivateSoftwareJob • 243
CreateActivateSoftwareJob3 • 245
CreateActivateSoftwareJobInSoftwarePolicy • 247
CreateActivateSoftwareJobInSoftwarePolicy3 • 248
CreateActivateSoftwareJobInSoftwarePolicyByName • 249
CreateActivateSoftwareJobInSoftwarePolicyByName3 • 250
CreateAddedSoftwarePackageProcedure • 317
CreateAddedSoftwarePackageProcedure2 • 318
CreateAddedSoftwarePackageProcedure3 • 319
CreateAgentComponents • 673, 674
createAssessmentTask • 922
createAssessmentTemplate • 918
CreateAssetScriptJob • 611
CreateAssetScriptJobResponse • 788
createAutomationProcess • 933
CreateCatalogSoftwarePackageGroup • 291
CreateCatalogSoftwarePackageGroupProperties • 674
CreateComputer • 546
CreateComputer2 • 547
CreateComputerProperties • 675
CreateComputerProperties2 • 678
CreateConfigureSoftwareJob • 251
CreateConfigureSoftwareJob3 • 253
CreateConfigureSoftwareJobInSoftwarePolicy • 254
CreateConfigureSoftwareJobInSoftwarePolicy3 • 255
CreateConfigureSoftwareJobInSoftwarePolicyByName • 256
CreateConfigureSoftwareJobInSoftwarePolicyByName3 • 258
CreateDistributionContainer • 583
CreateDistributionContainerProperties • 631
CreateDomainGroup • 599
CreateDomainGroupProperties • 632
CreateDomainGroupProperties • 632
CreateInstallSoftwareJob • 259
CreateInstallSoftwareJob2 • 430
CreateInstallSoftwareJob3 • 260
CreateInstallSoftwareJobInSoftwarePolicy • 261
CreateInstallSoftwareJobInSoftwarePolicy2 • 431
CreateInstallSoftwareJobInSoftwarePolicy3 • 262

CreateInstallSoftwareJobInSoftwarePolicyByName • 432
CreateInstallSoftwareJobInSoftwarePolicyByName3 • 433
CreateInstallSoftwareJobOrderPolicyProperties • 632
CreateInstallSoftwareJobOrderPolicyProperties3 • 634
CreateInstallSoftwareJobOrderProperties • 636
CreateInstallSoftwareJobOrderProperties3 • 637
CreateOSIMComputer • 449
CreateOSIMComputerProperties • 449, 812
createOSIMTemplate • 926
CreateQuery • 167
CreateQuery2 • 167
CreateQueryProperties • 640, 641
CreateQueryProperties2 • 641
CreateRecoverSoftwareJob • 265
CreateRecoverSoftwareJob3 • 266
CreateRenewSoftwareJob • 263
CreateRenewSoftwareJob3 • 264
CreateSecurityArea • 202
CreateSecurityAreaProperties • 641
CreateSecurityProfile • 179
CreateSecurityProfileProperties • 642
createSoftwareDeploymentTemplate • 929
CreateSoftwareGroup • 491
CreateSoftwareGroupByName • 492
CreateSoftwareGroupProperties • 643
CreateSoftwareJobContainer • 215
CreateSoftwareJobContainer2 • 434
CreateSoftwareJobContainer3 • 216
CreateSoftwareJobContainer4 • 239
CreateSoftwareJobContainerProperties • 643
CreateSoftwareJobContainerProperties2 • 216, 679
CreateSoftwareJobContainerProperties3 • 680
CreateSoftwareJobContainerProperties4 • 949
CreateSoftwareJobLinkage • 746
CreateSoftwareJobOrderPolicyProperties • 682
CreateSoftwareJobOrderPolicyProperties3 • 683
CreateSoftwareJobOrderProperties • 685
CreateSoftwareJobOrderProperties3 • 687
CreateSoftwarePackage • 379
CreateSoftwarePackage3 • 393
CreateSoftwarePackageGroup • 292
CreateSoftwarePackageGroupProperties • 689
CreateSoftwarePackageProcedure • 319
CreateSoftwarePackageProcedure2 • 321

CreateSoftwarePackageProcedure3 • 320
CreateSoftwarePackageProcedureDependency • 311
CreateSoftwarePackageProcedureDependencyProperties • 644
CreateSoftwarePackageProcedureGroup • 354
CreateSoftwarePackageProcedureGroupProperties • 644
CreateSoftwarePackageProcedureProperties • 644
CreateSoftwarePackageProcedureProperties2 • 646
CreateSoftwarePackageProcedureProperties3 • 650
CreateSoftwarePackageProperties • 690
CreateSoftwarePackageProperties3 • 900
CreateSoftwarePackageVolume • 372
CreateSoftwarePackageVolumeProperties • 690
CreateSoftwarePolicy • 401
CreateSoftwarePolicy2 • 401
CreateSoftwarePolicy3 • 402
CreateSoftwarePolicyByName • 403
CreateSoftwarePolicyByName3 • 404
CreateSoftwarePolicyProperties • 779
CreateSoftwarePolicyProperties2 • 693
CreateSoftwarePolicyProperties3 • 781
CreateUninstallSoftwareJob • 267
CreateUninstallSoftwareJob3 • 268
CreateUninstallSoftwareJobInSoftwarePolicy • 269
CreateUninstallSoftwareJobInSoftwarePolicy3 • 271
CreateUnInstallSoftwareJobInSoftwarePolicyByName • 272
CreateUnInstallSoftwareJobInSoftwarePolicyByName3 • 273
CreateUnitGroup • 502
CreateUnitGroup2 • 503
CreateUnitGroup3 • 527
CreateUnitGroupByName • 504
CreateUnitGroupProperties • 783
CreateUnitGroupProperties2 • 746
CreateUnitGroupProperties3 • 917
CreateUnitGroupPropertiesByName • 695
CreateUserProfile • 571
CreateUserProfileProperties • 696
Creating a Software Container • 38
Creating a Software Job • 39
Creating a Software Package • 32
Creating a Software Package Procedure • 35
Creating a Software Package Volume • 34
Creating and Maintaining a Connection • 24, 27
CustomizationParameter • 964

D

DateTime • 697
DateTime2 • 809
DateTimeRestrictions • 792, 1093
DayOfTheWeekDependency • 797
DeleteAlerts • 156
DeleteAllAlerts • 158
deleteAssessmentTasks • 925
deleteAssessmentTemplates • 919
DeleteAssetJobs • 615
deleteAutomationProcesses • 934
DeleteComputers • 547
DeleteComputersByName • 548
DeleteDistributionContainers • 584
DeleteDomainGroups • 599
DeleteElementsFromSoftwarePackageVolume • 372
DeleteItems • 529
deleteOSIMTemplates • 926
DeleteOSInstallationRequest • 465
DeleteOSInstallationRequestByName • 465
DeleteQueries • 168
DeleteSecurityAreas • 203
DeleteSecurityAreasByName • 206
deleteSecurityProfiles • 180
DeleteSecurityProfiles • 180
DeleteSecurityProfilesByName • 208
deleteSoftwareDeploymentTemplates • 930
DeleteSoftwareGroups • 493
DeleteSoftwareGroupsByName • 493
DeleteSoftwareJobContainers • 217
DeleteSoftwareJobInComputerByName • 275
DeleteSoftwareJobsInComputer • 274
DeleteSoftwareJobsInComputerByName • 275
DeleteSoftwarePackageGroups • 293
DeleteSoftwarePackageProcedureDependency • 312
DeleteSoftwarePackageProcedureGroups • 322, 354
DeleteSoftwarePackageProcedures • 322
DeleteSoftwarePackages • 379
DeleteSoftwarePackagesByName • 380
DeleteSoftwarePackageVolumes • 373
DeleteSoftwarePolicies • 405
DeleteSoftwarePoliciesByName • 406
DeleteSoftwarePoliciesInUnitGroup • 407
DeleteSoftwarePoliciesInUnitGroupByName • 407
DeleteUnitGroups • 506
DeleteUnitGroups2 • 525
DeleteUnitGroupsByName • 506
DeleteUnitGroupsByName2 • 526

DeleteUserProfiles • 572
DeleteUserProfilesByName • 572
Dependency • 790
Deployment • 158
DeploymentInfo • 699
DeploymentJobProperties • 673
DeploymentPackageInfo • 700
DeploymentPackageProperties • 697, 981
DeploymentPackagePropertiesRequired • 698
DeploymentStatusCode • 68
DeployPackage • 163
DeployPackageOnTargetComputers • 163
DeployProperties • 699
DeRegisterBootImage • 479
DeRegisterOSImage • 479
DeregisterOSIMComputer • 458
DeRegisterOSIMComputerByName • 459
DiscardDeployment • 161
DiscardDeploymentJob • 161
DiscoveredSoftwareProperties • 700, 701, 980, 981
DiscoveredSoftwarePropertiesRequired • 701
DiscoveredSoftwareSourceType • 70
DiscoveredSoftwareType • 70
DistributionContainerOrderProperties • 701, 981
DistributionContainerOrderProperties3 • 702
DistributionContainerOrderPropertiesRequired • 692
DistributionContainerOrderType • 70
DistributionContainerProperties • 691, 981
DistributionContainerPropertiesRequired • 692
DistributionDomainProperties • 702, 981
DistributionDomainPropertiesRequired • 703
DistributionOrderStatus • 71
DistributionPermissionMask • 692
DistributionStatus • 71
DomainGroupProperties • 704, 981
DomainGroupPropertiesRequired • 704
DomainProperties • 705, 981
DomainPropertiesRequired • 748
DomainProperty • 72
DomainPropertyFilter • 747, 981
domainType • 71
DomainType • 71
DownloadMethod • 74
Dsm
 OpenAssetScriptJobListResponse • 796
DSM Web Services End Point for UTF-8 Strings • 45

E

ElementType • 74
ElementTypeMask • 750
EmptySoftwarePackageGroups • 293
EmptySoftwarePackageProcedureGroups • 323, 355
Enterprise Manager Distribution Functionalities • 575
Enterprise Manager Domain and Domain Group Functionalities • 597
Enumeration • 70
Enumerations • 46, 48, 66, 70, 74, 75, 84, 85, 90, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 104, 106, 107, 108
Enumerations for Automation Services • 128
EvaluateSoftwarePolicy • 406
EvaluateUnitGroupQuery • 521, 548
EvaluationTypeMask • 750
executeAssessmentTask • 922
executeAutomationProcess • 934
ExportSoftwarePackage • 381
ExportSoftwarePackageByName • 382
ExportSoftwarePkgProperties • 709

F

FetchItemJobOrderProperties • 710, 981
FetchItemOrderProperties • 710
FileDependency • 793
FileSystemElementPath • 710, 981
FilterCondition • 130
FILTERCONDITION • 113
FindComputer • 548
FindDomain • 600
FindDomainGroup • 601
Finding a DSM Object • 30, 37
FindSoftwareJob • 275
FindSoftwareJobContainer • 218
FindSoftwarePackage • 382
FindSoftwarePackageGroup • 294
FindSoftwarePackageProcedure • 323, 832
FindSoftwarePackageProcedureGroup • 324, 355
FindSoftwarePolicy • 408
FindUserAccount • 549
FindUserProfile • 550

G

Generating Client Stubs • 24
getActivityLog • 942

getActivityLog2 • 943
GetAllSecurityClassPermissionsForSecurityProfile • 192
getAssessmenTaskList • 924
getAssessmentResult • 923
getAssessmentStatus • 923
getAssessmentTask • 924
GetAssessmentTaskListResponse • 963
getAssessmentTemplate • 919
getAssessmentTemplateList • 920
getAssessmentTemplateListDetailed • 921
GetAssessmentTemplateListDetailedResponse • 961
GetAssetScriptJob • 611
GetAssetScriptJobResponse • 797
GetAssetScriptJobs • 613, 1092
GetAssetScriptJobsProperties • 797
getAutomationProcessList • 935
getAutomationProcessListDetailed • 936
GetAutomationProcessListDetailedResponse • 970
GetAutomationProcessListResponse • 969
getAutomationProcessStatus • 939
getAutomationProcessTargetStatus • 938
getAutomationResult • 937
getAutomationResult2 • 937
GetBootImageDeployedBootServers • 478
GetBootImageList • 471
GetBootImageList2 • 489
GetBootServerList • 476
GetBootServerList2 • 490
GetComputer • 551
GetComputerByName • 568
GetComputerHWModel • 570
GetComputerHWModelByName • 570
GetComputerOSName • 568
GetComputerSDProperties • 570
GetComputerSDPropertiesByName • 571
GetDeploymentJobStatus • 162
GetDeploymentPackagesOnScalabilityServer • 163
GetDistributionContainer • 585
GetDistributionContainerDistributionDomains • 585
GetDistributionContainerOrder • 586
GetDistributionContainerOrder3 • 587
GetDistributionContainerOrders • 588
GetDistributionContainerOrders3 • 589
GetDistributionContainers • 590
GetDomain • 602
GetDomainGroup • 602
GetDomainGroupDomainGroupMembers • 603
GetDomainGroupDomainMembers • 603

GetHardwareModelList • 569
GetInstallationRequestList • 463
GetInstallationRequestListByName • 464
GetInstalledComponents • 135
GetInstalledComponents2 • 136
GetInstalledComponents3 • 137
GetManagerConfigurationParameterInt • 156
GetManagerProperties • 155
GetOSImageDeployedBootServers • 477
GetOSImageParameters • 475
GetOSImages • 470
GetOSImages2 • 484
GetOSIMParameterDefaults • 488
GetOSIMParameterDefaultsByName • 489
GetOSIMTarget • 482
GetOSIMTargetByName • 483
GetOSIMTargetList • 486
GetOSIMTargets • 481
GetOSIMTargets2 • 485
getOSIMTemplate • 927
getOSIMTemplateList • 927
getOsimTemplateListDetailed • 928
GetOsimTemplateListDetailedResponse • 965
GetOSInstallationParameters • 467
GetOSInstallationParametersByName • 468
GetOSInstallationRequest • 461
GetOSInstallationRequestByName • 462
GetOSTypesXML • 172
GetParentSoftwareJobContainers • 219
GetParentSoftwareJobContainers2 • 435
GetParentSoftwareJobContainers3 • 220
GetParentSoftwarePackageGroups • 295
GetParentSoftwarePackageProcedureGroups • 325, 356
GetQueries • 168
GetQueries2 • 176
GetQuery • 169
GetQuery2 • 170
GetQueryFolderId • 175
GetQueryFolders • 177
GetQueryResults • 170
GetScalabilityServers • 473
GetSecurityArea • 205
GetSecurityAreaByName • 205
GetSecurityAreaConfiguration • 185, 710
GetSecurityAreaConfigurationByName • 207
GetSecurityAreas • 204
GetSecurityClasses • 184
GetSecurityClasses X • 184

GetSecurityClassPermissions • 191
GetSecurityClassPermissionsForAllSecurityProfiles • 191
GetSecurityGroupPermissions • 195, 711
GetSecurityGroupPermissionsForAllSecurityProfiles • 196
GetSecurityGroupsPermissions • 195
GetSecurityObjectOwnership • 186
GetSecurityObjectPermissions • 192
GetSecurityObjectPermissionsForAllSecurityProfiles • 193
GetSecurityObjectPermissionsString • 188
GetSecurityObjectSecurityAreaLinkages • 189
GetSecurityObjectsPermissions • 711
GetSecurityObjectsPermissions • 711
GetSecurityObjectsSecurityAreaLinkages • 189
GetSecurityProfile • 182
GetSecurityProfileByName • 209
GetSecurityProfiles • 181
GetServiceDeskRequestURLForContainer • 240
GetServiceDeskTemplates • 155
getSoftwareDeploymentTemplate • 931
getSoftwareDeploymentTemplateList • 931
getSoftwareDeploymentTemplateListDetailed • 932
GetSoftwareDeploymentTemplateListDetailedResponse • 968
GetSoftwareJob • 276
GetSoftwareJob2 • 436
GetSoftwareJob3 • 277
GetSoftwareJobContainer • 221
GetSoftwareJobContainer2 • 436
GetSoftwareJobContainer3 • 222
GetSoftwareJobContainers • 222
GetSoftwareJobContainers2 • 437
GetSoftwareJobContainers3 • 223
GetSoftwareJobContainerSoftwareJobs • 225
GetSoftwareJobContainerSoftwareJobs2 • 438
GetSoftwareJobContainerSoftwareJobs3 • 226
GetSoftwareJobContainersRecursive • 227
GetSoftwareJobContainersRecursive2 • 439
GetSoftwareJobContainersRecursive3 • 228
GetSoftwareJobOutputFile • 286
GetSoftwareJobOutputFile Location • 286, 287
GetSoftwareJobOutputFile2 • 287
GetSoftwareJobs • 278
GetSoftwareJobs2 • 440
GetSoftwareJobs3 • 279
GetSoftwareJobUnitSoftwareJobs • 280
GetSoftwareJobUnitSoftwareJobs2 • 280

GetSoftwarePackage • 383
GetSoftwarePackage2 • 384
GetSoftwarePackage3 • 394
GetSoftwarePackageGroup • 295
GetSoftwarePackageGroups • 296
GetSoftwarePackageGroupSoftwarePackages • 297
GetSoftwarePackageGroupSoftwarePackages2 • 298
GetSoftwarePackageGroupSoftwarePackagesRecursive • 299
GetSoftwarePackageGroupSoftwarePackagesRecursive2 • 299
GetSoftwarePackageGroupsRecursive • 300
GetSoftwarePackageInstallationList • 396
GetSoftwarePackageInstallations • 384
GetSoftwarePackageInstallations2 • 385
GetSoftwarePackageList • 397
GetSoftwarePackageList3 • 395
GetSoftwarePackageProcedure • 326
GetSoftwarePackageProcedure2 • 326
GetSoftwarePackageProcedure3 • 347
GetSoftwarePackageProcedure4 • 348
GetSoftwarePackageProcedureDependencies • 312
GetSoftwarePackageProcedureGroup • 328
GetSoftwarePackageProcedureGroup • 327, 357
GetSoftwarePackageProcedureGroupProcedureList • 367
GetSoftwarePackageProcedureGroups • 328, 357
GetSoftwarePackageProcedureGroupSoftwarePackageProcedures • 328, 358
GetSoftwarePackageProcedureGroupSoftwarePackageProcedures2 • 329
GetSoftwarePackageProcedureGroupSoftwarePackageProcedures3 • 360
GetSoftwarePackageProcedureGroupsRecursive • 330, 360
GetSoftwarePackageProcedureList • 350
GetSoftwarePackageProcedures • 331
GetSoftwarePackageProcedures2 • 332
GetSoftwarePackageProcedures3 • 349
GetSoftwarePackages • 386
GetSoftwarePackages2 • 387
GetSoftwarePackageVolume • 373, 374
GetSoftwarePackageVolumeElements • 374
GetSoftwarePackageVolumes • 375
GetSoftwarePolicies • 409
GetSoftwarePolicies2 • 441
GetSoftwarePolicies3 • 410
GetSoftwarePoliciesRecursive • 411
GetSoftwarePolicy • 411

GetSoftwarePolicy2 • 441
GetSoftwarePolicy3 • 412
GetSoftwarePolicySoftwareJobs • 413
GetSoftwarePolicySoftwareJobs2 • 442
GetSoftwarePolicySoftwareJobs3 • 414
GetSoftwarePolicySoftwarePolicyExceptions • 415
getSupportedOSForAssessment • 944
GetSystemGroupUUID • 507
GetTemplateListResponse • 960
GetUnitDiscoveredSoftware • 553
GetUnitGroup • 507
GetUnitGroup2 • 528
GetUnitGroupName • 532
GetUnitGroupComputerMembers • 508
GetUnitGroupComputerMembers2 • 530
GetUnitGroupComputerMembers3 • 532
GetUnitGroupsXML • 533
GetUnitGroupUnitGroupMembers • 508
GetUnitGroupUnitMembers • 509
GetUnitGroupUserAccountMembers • 510
GetUnitGroupUserProfileMembers • 511
GetUnitInventoryComponentGroups • 536
GetUnitInventoryComponents • 537
GetUnitInventoryGroups • 538
GetUnitInventoryItems • 538
GetUnitQueries • 554
GetUnitSoftwareJobList • 567
GetUnitUnitGroups • 555
GetUnitUnitSoftwareInstallationJobs • 556
GetUnitUnitSoftwareInstallationJobs2 • 556
GetUnitUnitSoftwareJobs • 557
GetUnitUnitSoftwareJobs2 • 558
GetUnmanagedOSIMRegisteredComputers • 460
GetUnnamedOSIMRegisteredComputers • 459
GetUserAccount • 559
GetUserProfile • 560
GlobalType • 975
GroupFlag • 74
GroupInheritance • 75
GroupSystem • 75
GroupType • 975

H

HaltDeliveryDistributionContainers • 591
HaltDeliverySoftwareJobContainer • 229
Handling Errors • 26
Handling UUIDs • 24
HardWareModelType • 975

Health Monitoring • 156
Health Monitoring Sequences • 958
HIPStatus • 123
HLA000008 • 1055
HLA000201 • 1055
HLA000202 • 1055
HLA000204 • 1056
HLA000205 • 1056
HLA000206 • 1056
HLA000207 • 1057
HLA000208 • 1057
HLA000209 • 1057
HLA000210 • 1057
HLA000211 • 1058
HLA000212 • 1058
HLA000213 • 1058
HLA000214 • 1059
HLA000215 • 1059
HLA000216 • 1059
HLA000217 • 1059
HLA000218 • 1060
HLA000219 • 1060
HLA000220 • 1060
HLA000221 • 1060
HLA000222 • 1061
HLA000223 • 1061
HLA000224 • 1061
HLA000225 • 1061
HLA000226 • 1062
HLA000227 • 1062
HLA000228 • 1062
HLA000229 • 1063
HLA000230 • 1063
HLA000231 • 1063
HLA000232 • 1064
HLA000233 • 1064
HLA000234 • 1064
HLA000235 • 1064
HLA000236 • 1065
HLA000237 • 1065
HLA000238 • 1065
HLA000239 • 1065
HLA000240 • 1066
HLA000241 • 1066
HLA000242 • 1066
HLA000243 • 1067
HLA000244 • 1067
HLA000245 • 1067
HLA000246 • 1067

HLA000247 • 1068
HLA000259 • 1090
HLA000260 • 1090
HWMModelFilter • 630, 957
HWMODEL-FILTER-CONDITION • 125

I

ImageType • 963, 964
ImportSoftwarePackage • 388
ImportSoftwarePackage2 • 388
ImportSoftwarePackageByName • 389
ImportSoftwarePackageProperties • 711
ImportSoftwarePackageProperties2 • 712
ImportSoftwarePackagePropertiesByName • 713
Inheritance Level • 76
IsServiceDeskIntegrationEnabled • 154
Item • 915

J

JobDependency • 793
JobExecProperties • 794

L

Limitations • 997
LinkAssetJobsToUnit • 615
LinkAssetJobsToUnitGroup • 616
LinkComputersToUnitGroup • 511
LinkComputersToUnitGroupByName • 512
LinkDomainGroupsToDomainGroup • 604
LinkDomainsToDomainGroup • 605
LinkedObjectTypeRequired • 76
LinkItemsToUnitGroup • 528
LinkSecurityAreasToSecurityProfile • 198
LinkSecurityAreasToSecurityProfileByName • 199
LinkSecurityProfilesToSecurityArea • 201
LinkSecurityProfilesToSecurityAreaByName • 200
LinkSoftwareGroupsToSoftwarePackageGroup • 494
LinkSoftwareGroupsToSoftwarePackageGroupByName • 495
LinkSoftwarePackageGroupsToSoftwarePackageGroup • 301
LinkSoftwarePackageProcedureGroupsToSoftwarePackageGroup • 333
LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroup • 334, 361
LinkSoftwarePackageProceduresToSoftwarePackageProcedureGroupByName • 333

LinkSoftwarePackagesToSoftwarePackageGroup • 301, 335
LinkSoftwarePackagesToSoftwarePackageGroupByName • 302
LinkSoftwarePoliciesToUnitGroup • 415
LinkSoftwarePoliciesToUnitGroupByName • 416
LinkUnitGroupsToUnitGroup • 513
LinkUnitGroupsToUnitGroupByName • 513
LinkUserAccountsToUnitGroup • 514
LinkUserProfilesToUnitGroup • 515
Login • 44, 143
Login2 • 145
Login3 • 150
Login4 • 152
LogLevel • 134
Logout • 153

M

MachineCredentials • 706
ManagerProperties • 948
ManagerVersion • 948
Method name • 134
Methods • 134
Miscellaneous • 792
MoveItemsToUnitGroup • 530

N

NetWorkDrive • 795
NoOfDistributionDomains • 707

O

ObjectPreference • 124
OpenAssetScriptJobList • 612, 1092
OpenDistributionContainerDistributionDomainList • 591
OpenDistributionContainerList • 592
OpenDistributionContainerOrderList • 593
OpenDomainGroupDomainGroupMembersList • 605
OpenDomainGroupDomainGroupsList • 605, 606
OpenDomainGroupDomainMembersList • 607
OpenOSImageList • 469
OpenOSIMTargetList • 480
OpenParentSoftwareJobContainerList • 229
OpenParentSoftwareJobContainerList2 • 443
OpenParentSoftwareJobContainerList3 • 230
OpenParentSoftwarePackageGroupList • 303
OpenParentSoftwarePackageProcedureGroupList • 335, 362, 517

OpenQueryList • 173
OpenQueryResultList • 173
OpenScalabilityServerList • 473
OpenSecurityAreaList • 203, 204
OpenSecurityClassList • 183
OpenSecurityProfileList • 180
OpenSoftwareJobContainerList • 231
OpenSoftwareJobContainerList2 • 444
OpenSoftwareJobContainerList3 • 232
OpenSoftwareJobContainerListRecursive • 233
OpenSoftwareJobContainerListRecursive2 • 445
OpenSoftwareJobContainerListRecursive3 • 234
OpenSoftwareJobContainerSoftwareJobList • 235
OpenSoftwareJobList • 281
OpenSoftwareJobUnitSoftwareJobL • 282
OpenSoftwareJobUnitSoftwareJobList • 282
OpenSoftwarePackageGroupList • 304
OpenSoftwarePackageGroupListRecursive • 305
OpenSoftwarePackageGroupSoftwarePackageList • 306
OpenSoftwarePackageGroupSoftwarePackageListRecursive • 306
OpenSoftwarePackageInstallationList • 389
OpenSoftwarePackageList • 390
OpenSoftwarePackageProcedureDependencyList • 313
OpenSoftwarePackageProcedureGroupList • 336, 363
OpenSoftwarePackageProcedureGroupListRecursive • 337, 345
OpenSoftwarePackageProcedureGroupSoftwarePackageProcedureList • 338, 364
OpenSoftwarePackageProcedureList • 339
OpenSoftwarePackageVolumeElementList • 376
OpenSoftwarePackageVolumeList • 377
OpenSoftwarePolicyList • 416
OpenSoftwarePolicyList2 • 446
OpenSoftwarePolicyList3 • 417
OpenSoftwarePolicyListRecursive • 418
OpenSoftwarePolicySoftwareJobList • 419
OpenSoftwarePolicySoftwarePolicyExceptionList • 420
OpenUnitDiscoveredSoftwareList • 560
OpenUnitGroupComputerMembersList • 515
OpenUnitGroupComputerMembersList2 • 531
OpenUnitGroupUnitGroupMembersList • 516
OpenUnitGroupUnitMembersList • 517
OpenUnitGroupUserAccountMembersList • 518
OpenUnitGroupUserProfileMembersList • 519

- OpenUnitInventoryComponentGroupList • 539
- OpenUnitInventoryComponentList • 541
- OpenUnitInventoryGroupList • 542
- OpenUnitInventoryItemList • 561
- OpenUnitQueryList • 563
- OpenUnitUnitGroupList • 563
- OpenUnitUnitSoftwareInstallationJobList • 564
- OpenUnitUnitSoftwareJobList • 565
- operatingSystemType • 76
- OperatingSystemType • 977
- OSIM • 448
- OSImageFilter • 801
- OSImageProperties • 469, 480, 802, 815
- OSIMAGEPROPERTY • 114
- OSIMparameterDesc • 799, 994
- OSIMProperties • 798, 992
- OSIMPropertiesWithDesc • 799, 994
- OSIMTargetFilter • 802
- OSIMTargetProperties • 803, 952
- OSIMTargetProperties2 • 952
- OSIMTARGETPROPERTY • 115
- OsimTemplate • 963
- OsimTemplateFilter • 965
- OSIMTEMPLATEPROPERTY • 131
- OsimTemplateType • 130

P

- PackageExportType • 82
- PackageFormat • 64
- PackageParameters • 707
- Parameter • 960
- ParameterInfo • 707, 981, 996
- ParameterValue • 708
- Passing System Defined Groups as Parameters • 998
- PeriodicDependency • 794, 1093
- PERIODTYPE • 109
- PostJob • 792
- PostJobProperties • 796
- PreJob • 791
- Priority, Reinstall, and Policy Enhancement • 428
- Properties Architecture Concepts • 25
- Properties Architecture Implementation • 25
- Property_Type • 112, 474
- ProtectionAgentStatus • 121

Q

- Queries • 165
- QueryFilter • 629, 956

- QueryFilterCondition • 125
- QueryFolderProperties • 630, 957
- QueryFragment • 82, 708, 981
- QueryFragment2 • 82, 714, 981
- QueryFragmentType • 84
- QueryFragmentType2 • 85
- QueryProperties • 640, 716, 718, 734, 981
- QueryProperties2 • 629, 956
- QueryProperties3 • 717
- QueryPropertiesRequired • 718
- QueryPropertiesRequired3 • 718
- QueryProperty • 630, 957
- QueryResultProperties • 708, 709, 981
- QueryResultPropertiesRequired • 709
- QueryTargetType • 124
- QueryType • 85

R

- RACPolicy • 86
- RCAgentStatus • 120
- ReActivateOSInstallation • 453
- ReActivateOSInstallationByName • 455
- RegisterComputerWithOSIM • 449
- RegisterComputerWithOSIMByName • 450
- ReInstallIOS • 456
- ReInstallIOSByName • 457
- RenewDeliveryDistributionContainers • 594
- RepositionSoftwarePackageProcedure • 340
- RepositionSoftwarePackageProcedureByName • 341
- ResetTimeout • 153
- ResultGroup • 130
- ResumeDeliveryDistributionContainerProperties • 709
- ResumeDeliveryDistributionContainers • 594
- ResumeDeliverySoftwareJobContainer • 236
- ResumeDeployment • 160
- ResumeDeploymentJob • 160
- ResumeSoftwareJobs • 283
- Retrieving Attributes • 29
- RetryDeployment • 161
- RetryDeploymentJob • 161
- runAutomationGenericJobContainer • 941
- runAutomationOSInstallJob • 941
- runAutomationSoftwareJobContainer • 940

S

- ScalabilityServerProperties • 799, 995
- ScalabilityServerPropertiesRequired • 473, 800

ScanDeploymentPackageOnTargetComputers • 158
ScanDeploymentPackagesOnScalabilityServer • 163
ScanPackageInfo • 719
ScanPackageOnTargetComputers • 158
ScanParameters • 719
Scheduling • 789
SchedulingOptions • 789
SDAgentProperties • 721
SDAgentStatus • 119
SDProperties • 947
SDTemplateProperties • 945
SealAndBuildSoftwareJobContainer • 236
Sealing a Software Package • 37
Sealing and Activating a Job • 40
SealSoftwareJobContainers • 50, 237
SealSoftwarePackages • 391
SealSoftwarePolicies • 421
SearchComputerProperty • 474
Security • 178
SecurityArea • 750, 987
SecurityAreaByName • 725
SecurityAreaConfiguration • 726
SecurityAreaConfigurationByName • 726
SecurityAreaLinkage • 86
SecurityAreaLinkage • 86
SecurityAreaProperties • 722, 981
SecurityAreaPropertiesRequired • 722
SecurityClassPermissionProperties • 723
SecurityClassPermissions • 191, 722, 981, 989
SecurityClassPermissionsForSecurityProfile • 192, 723
SecurityClassPermissionsForSecurityProfile • 723
SecurityClassProperties • 723, 981
SecurityClassPropertiesRequired • 723
SecurityClassType • 86, 981
SecurityGroup • 90
SecurityGroupPermissionsForAllSecurityProfiles • 724, 981
SecurityGroupProfilePermission • 711, 724, 981
SecurityObjectLinkageProperties • 724, 982
SecurityObjectOwner • 725, 982
SecurityObjectPermissions • 725
SecurityObjectPermissionsForAllSecurityProfiles • 193, 726, 982
SecurityObjectProfilePermission • 724, 982
SecurityObjectSecurityAreaLinkages • 725, 982
SecurityObjectSecurityAreaLinkages • 725
SecurityObjectsSecurityAreaProperties • 726
SecurityProfilePermission • 90, 722, 723

SecurityProfilePermission • 90
SecurityProfileProperties • 726, 982
SecurityProfilePropertiesRequired • 727
SecurityProfileSpecialPermissionFlags • 728
SecurityProfileType • 91
SendDistributionContainer • 595
Sequences • 619
Sequences for Automation Web Services • 958
Sessions • 24
setActivationStatus • 944
SetAssetScriptJob • 614
SetCatalogSoftwarePackageGroup • 308
SetCatalogSoftwarePackageGroupProperties • 729
SetComputer • 552
SetComputerByName • 552
SetComputerProperties • 730
SetDistributionContainer • 596
SetDistributionContainerProperties • 751
SetDomainGroup • 608
SetDomainGroupProperties • 731
SetOSIMComputerMACAddressByName • 466
SetOSInstallationParameter • 468
SetOSInstallationParameter2 • 487
SetOSInstallationParameterByName • 469
SetOSInstallationParameterByName2 • 488
SetOSInstallationParameters • 466
SetOSInstallationParametersByName • 467
SetQuery • 174
SetQuery2 • 175
SetQueryProperties • 734
SetQueryProperties3 • 735
SetSecurityArea • 206
SetSecurityAreaByName • 207, 981
SetSecurityAreaConfiguration • 186
SetSecurityAreaConfigurationByName • 208
SetSecurityAreaProperties • 751
SetSecurityClassesPermissions • 190
SetSecurityClassPermissions • 190, 982
SetSecurityClassProperties • 731
SetSecurityClassProperties • 731
SetSecurityGroupPermissions • 728, 982
SetSecurityGroupsPermissions • 197
SetSecurityObjectOwnership • 187
SetSecurityObjectPermissions • 728, 982
SetSecurityObjectsPermissions • 194
SetSecurityObjectsSecurityAreaLinkages • 189, 982
SetSecurityProfile • 183
SetSecurityProfileByName • 209
SetSecurityProfileProperties • 727

SetSoftwareGroup • 496
SetSoftwareGroupByName • 496
SetSoftwareGroupProperties • 734
SetSoftwareJob • 283
SetSoftwareJob2 • 284, 447
SetSoftwareJob3 • 285
SetSoftwareJobContainer • 237
SetSoftwareJobContainer2 • 447
SetSoftwareJobContainer3 • 238
SetSoftwareJobContainerPriority • 448
SetSoftwareJobContainerProperties • 739
SetSoftwareJobContainerProperties2 • 731, 739
SetSoftwareJobContainerProperties3 • 740
SetSoftwareJobProperties • 742
SetSoftwareJobProperties2 • 743
SetSoftwareJobProperties3 • 744
SetSoftwarePackage • 392
SetSoftwarePackage3 • 393
SetSoftwarePackageGroup • 308
SetSoftwarePackageGroupProperties • 751
SetSoftwarePackageProcedure • 342
SetSoftwarePackageProcedure2 • 343
SetSoftwarePackageProcedure3 • 344
SetSoftwarePackageProcedureGroup • 344, 365
SetSoftwarePackageProcedureGroupProperties • 735
SetSoftwarePackageProcedureProperties • 752
SetSoftwarePackageProcedureProperties2 • 754
SetSoftwarePackageProcedureProperties3 • 736
SetSoftwarePackageProperties • 757
SetSoftwarePackageProperties3 • 901
SetSoftwarePackageVolume • 377
SetSoftwarePackageVolumeProperties • 757
SetSoftwarePolicy • 422
SetSoftwarePolicy2 • 422
SetSoftwarePolicy3 • 423
SetSoftwarePolicyByName • 424
SetSoftwarePolicyByName3 • 425
SetSoftwarePolicyProperties • 758
SetSoftwarePolicyProperties2 • 732
SetSoftwarePolicyProperties3 • 759
SetSoftwarePolicyPropertiesByName • 761
SetSoftwarePolicyPropertiesByName3 • 763
SetTimeoutProperties • 766
SetTimeoutValue • 154
SetUnitGroup • 520
SetUnitGroup2 • 521
SetUnitGroupProperties • 767, 874
SetUnitGroupProperties2 • 874

SetUserProfile • 573
SetUserProfileByName • 573
SetUserProfileProperties • 767
Software • 210
Software Groups • 491
Software Job Containers • 211
Software Jobs • 241
Software Package Groups • 288
Software Package Procedure Dependencies • 310
Software Package Procedure Groups • 351
Software Package Procedures • 315
Software Package Volumes • 368
Software Packages • 378
Software Policies • 398
Software PolicyScope • 100
SoftwareByName • 864
SoftwareContainerData • 971
SoftwareDeploymentTemplate • 965
SoftwareDeploymentTemplateFilter • 968
SOFTWAREDEPLOYMENTTEMPLATEPROPERTY • 131
SoftwareGroupType • 91
SoftwareJobBootMask • 785
SoftwareJobBootMask (bootAfterMask) • 784
SoftwareJobBootMask (bootBeforeMask) • 785
SoftwareJobBootMask3 (bootAfterMask) • 784
SoftwareJobBootMask3 (bootBeforeMask) • 786
SoftwareJobContainerOrder • 863
SoftwareJobContainerParameter • 974
SoftwareJobContainerProperties • 643, 739, 786, 818, 982
SoftwareJobContainerProperties2 • 860
SoftwareJobContainerProperties3 • 816
SoftwareJobContainerPropertiesRequired • 818
SoftwareJobContainerPropertiesRequired2 • 741, 862
SoftwareJobContainerPropertiesRequired3 • 819
SoftwareJobContainerPropertyMask • 820
SoftwareJobContainerScope • 92
SoftwareJobContainerState • 92
SoftwareJobContainerState • 92
SoftwareJobContainerType • 92, 966
SoftwareJobData • 972
SoftwareJobLinkage • 864
SoftwareJobLinkageType • 109
SoftwareJobOperationMask • 821
SoftwareJobOperationMask2 • 822
SoftwareJobOrderProperties • 686, 823
SoftwareJobOrderProperties3 • 825
SoftwareJobPermissionMask • 768

SoftwareJobProperties • 742, 768, 777, 882, 885, 982
SoftwareJobProperties2 • 771
SoftwareJobProperties3 • 774
SoftwareJobPropertiesRequired • 777
SoftwareJobRenewState • 93
SoftwareJobStartupTime • 93
SoftwareJobState • 93
SoftwareJobStatusCounters • 778
SoftwareJobTask • 94
SoftwareJobType • 98
SoftwarePackage • 865
SoftwarePackageClass • 95
SoftwarePackageData • 974
SoftwarePackageFiledBy • 98
SoftwarePackageFilter • 945
SoftwarePackageGroupProperties • 690, 751, 778, 828, 982
SoftwarePackageGroupPropertiesRequired • 778
SoftwarePackageGroupProperty • 98, 107
SoftwarePackageGroupScope • 95
SoftwarePackageGroupType • 95
SoftwarePackageMappingType • 967
SoftwarePackageMask • 779
SoftwarePackageMaskType • 779
SoftwarePackageOrder • 865
SoftwarePackageProcedure • 865
SoftwarePackageProcedureBootMask • 829, 830
SoftwarePackageProcedureBootMask (bootAfterMask) • 829
SoftwarePackageProcedureBootMask2 (bootAfterMask) • 830
SoftwarePackageProcedureBootMask2 (bootBeforeMask) • 831
SoftwarePackageProcedureDependencyProperties • 644, 837, 838, 982
SoftwarePackageProcedureDependencyPropertiesRequired • 838
SoftwarePackageProcedureFiledBy • 95
SoftwarePackageProcedureGroupProperties • 644, 735, 838, 839, 982
SoftwarePackageProcedureGroupPropertiesRequired • 838, 839
SoftwarePackageProcedureGroupScope • 96
SoftwarePackageProcedureGroupType • 96
SoftwarePackageProcedureManagementTypeMask • 840
SoftwarePackageProcedureOperationMask • 841
SoftwarePackageProcedureOperationMask2 • 842

SoftwarePackageProcedureOperationMask3 • 654
SoftwarePackageProcedureOrder • 863
SoftwarePackageProcedureProperties • 646, 753, 756, 833, 841, 843, 882, 885, 982
SoftwarePackageProcedureProperties2 • 845
SoftwarePackageProcedureProperties3 • 737
SoftwarePackageProcedureProperties4 • 907
SoftwarePackageProcedurePropertiesRequired • 833
SoftwarePackageProcedurePropertiesRequired2 • 910
SoftwarePackageProcedureRelation • 96
SoftwarePackageProcedureTask • 97
SoftwarePackageProcedureType • 97
SoftwarePackageProcedureTypeMask • 834
SoftwarePackageProperties • 690, 712, 757, 835, 847, 882, 885, 982
SoftwarePackageProperties2 • 849, 946
SoftwarePackageProperties3 • 902
SoftwarePackagePropertiesRequired • 835
SoftwarePackagePropertiesRequired3 • 906
SoftwarePackageProperty • 65
SoftwarePackageType • 99, 967
SoftwarePackageVolumeElementProperties • 836, 850, 982
SoftwarePackageVolumeElementPropertiesRequired • 836
SoftwarePackageVolumeProperties • 691, 757, 836, 837, 982
SoftwarePackageVolumePropertiesRequired • 837
SoftwarePackagingType • 99
SoftwarePolicyEvaluationPeriodType • 98
SoftwarePolicyEvaluationType • 100
SoftwarePolicyExceptionProperties • 851, 982
SoftwarePolicyOrder • 863
SoftwarePolicyProperties • 758, 780, 851, 856, 982
SoftwarePolicyProperties2 • 865
SoftwarePolicyProperties3 • 854
SoftwarePolicyPropertiesRequired • 856
SoftwarePolicyPropertiesRequired2 • 857
SoftwarePolicyPropertiesRequired3 • 858
SoftwarePolicyScope • 100
SoftwarePolicyState • 862
SourceMedium • 101
SSL Configuration • 998
SSPackageInfo • 867
SSPackageInfoRequired • 867
StagePackage • 164
StagingCredentials • 868
StringValueType • 101

Structure CreateUnitGroupProperties2
CreateUnitGroupProperties2 • 746
SuspendDeployment • 160
SuspendDeploymentJob • 160
SuspendSoftwareJobs • 286
SystemGroup • 103

T

TemplateProperties • 961
The Base 64 Alphabet • 148
TriggerInfo • 798

U

Unit Groups • 499
Unit Inventory • 534
UnitGroupProperties • 767, 783, 870, 876
UnitGroupProperties2 • 915
UnitGroupPropertiesForSoftwarePolicy • 874
UnitGroupPropertiesRequired • 876
UnitInventoryComponentProperties • 868, 983
UnitInventoryComponentPropertiesRequired • 868
UnitInventoryGroupProperties • 869
UnitInventoryGroupPropertiesRequired • 869
UnitInventoryItemProperties • 869, 870, 983
UnitInventoryItemPropertiesRequired • 870
UnitObjectType • 110
UnitProperties • 877, 983
UnitPropertiesRequired • 877
UnitProperty • 104
UnitPropertyFilter • 877, 983
UnitQueryProperties • 880
UnitQueryPropertiesRequired • 879
Units • 543
Units and Groups • 498
UnitSoftwareJobProperties • 880, 983
UnitSoftwareJobProperties2 • 883
UnitSoftwareJobProperties3 • 911
UnitSoftwareJobPropertiesRequired • 886, 891
UnitSoftwareJobState • 106
UnitSoftwareJobTaskMask • 887
UnitType • 107
UnLinkAssetJobsFromUnit • 616
UnLinkAssetJobsFromUnitGroup • 617
UnlinkComputersFromUnitGroup • 522
UnlinkComputersFromUnitGroupByName • 522
UnlinkDomainGroupsFromDomainGroup • 609
UnlinkDomainsFromDomainGroup • 610
UnLinkItemsFromUnitGroup • 529

UnLinkSecurityAreasFromSecurityProfile • 198
UnLinkSecurityAreasFromSecurityProfileByName • 200
UnlinkSecurityProfilesFromSecurityArea • 202
UnlinkSecurityProfilesFromSecurityAreaByName • 201
UnlinkSoftwareGroupsFromSoftwarePackageGroup • 497
UnlinkSoftwareGroupsFromSoftwarePackageGroupByName • 498
UnlinkSoftwarePackageGroupsFromSoftwarePackageGroup • 309
UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroup • 346, 365
UnlinkSoftwarePackageProceduresFromSoftwarePackageProcedureGroupByName • 887
UnlinkSoftwarePackagesFromSoftwarePackageGroup • 366
UnlinkSoftwarePackagesFromSoftwarePackageGroupByName • 309
UnlinkSoftwarePoliciesFromUnitGroup • 426
UnlinkSoftwarePoliciesFromUnitGroupByName • 426
UnlinkUnitGroupsFromUnitGroup • 523
UnlinkUnitGroupsFromUnitGroupByName • 523
UnlinkUserAccountsFromUnitGroup • 524
UnlinkUserProfilesFromUnitGroup • 525
UnsealSoftwareJobContainers • 239
UnsealSoftwarePackages • 392
UnsealSoftwarePolicies • 427
UnsealSoftwarePoliciesByName • 427
UpdateAlerts • 157
UpdateAlertsProperties • 958
updateAssessmentTemplate • 918
updateOSIMTemplate • 926
updateSoftwareDeploymentTemplate • 930
UsageListMask • 889
User Profiles • 571
UserAccountProperties • 890, 891, 983
UserAccountPropertiesRequired • 891
UserAccountProperty • 107
UserAccountPropertyFilter • 891, 983
UserProfileProperties • 892, 897, 983
UserProfilePropertiesRequired • 896
UserProfileProperty • 108
UserProfilePropertyFilter • 897, 983

V

Versions • 134

VolumeElementAttributes • 899

W

Web Services Messages • 999

Web Services Reference • 43

Web Services Tutorial Writing Client Applications •
23

WeekDay • 109

Writing DSM Web Service Clients • 24

WS_OSIM_CONF_TYPE • 111, 461, 462

WS_OSIM_CONFIG_STATE • 112, 127

WS_OSIM_DEPLOY • 111, 451, 452, 454, 455, 456,
458

WS_OSIM_IR_STATUS • 111, 465

WSA000053 • 999

WSA000101 • 999

WSA000102 • 999

WSA000103 • 1000

WSA000104 • 1000

WSA000105 • 1000

WSA000106 • 1000

WSA000109 • 1001

WSA000110 • 1001

WSA000111 • 1001

WSA000112 • 1001

WSA000114 • 1002

WSA000201 • 1002

WSA000202 • 1002

WSA000203 • 1003

WSA000204 • 1003

WSA000205 • 1003

WSA000206 • 1003

WSA000207 • 1004

WSA000208 • 1004

WSA000209 • 1004

WSA000210 • 1004

WSA000211 • 1005

WSA000212 • 1005

WSA000213 • 1005

WSA000214 • 1005

WSA000215 • 1006

WSA000216 • 1006

WSA000217 • 1006

WSA000218 • 1006

WSA000219 • 1007

WSA000220 • 1007

WSA000221 • 1007

WSA000222 • 1008

WSA000223 • 1008

WSA000224 • 1008

WSA000225 • 1008

WSA000226 • 1009

WSA000227 • 1009

WSA000228 • 1009

WSA000229 • 1010

WSA000230 • 1010

WSA000231 • 1010

WSA000232 • 1011

WSA000233 • 1011

WSA000234 • 1011

WSA000235 • 1011

WSA000236 • 1012

WSA000237 • 1012

WSA000238 • 1012

WSA000239 • 1012

WSA000240 • 1013

WSA000241 • 1013

WSA000242 • 1013

WSA000243 • 1014

WSA000244 • 1014

WSA000245 • 1014

WSA000246 • 1015

WSA000247 • 1015

WSA000248 • 1015

WSA000249 • 1016

WSA000250 • 1016

WSA000251 • 1016

WSA000252 • 1017

WSA000253 • 1017

WSA000254 • 1017

WSA000255 • 1017

WSA000256 • 1018

WSA000257 • 1018

WSA000258 • 1018

WSA000259 • 1019

WSA000260 • 1019

WSA000261 • 1019

WSA000262 • 1020

WSA000263 • 1020

WSA000264 • 1020

WSA000265 • 1021

WSA000266 • 1021

WSA000267 • 1021

WSA000268 • 1022

WSA000269 • 1022

WSA000270 • 1022
WSA000271 • 1022
WSA000272 • 1023
WSA000273 • 1023
WSA000274 • 1023
WSA000275 • 1024
WSA000276 • 1024
WSA000277 • 1024
WSA000278 • 1025
WSA000279 • 1025
WSA000280 • 1025
WSA000281 • 1025
WSA000282 • 1026
WSA000284 • 1026
WSA000285 • 1026
WSA000286 • 1027
WSA000287 • 1027
WSA000288 • 1027
WSA000289 • 1027
WSA000290 • 1028
WSA000291 • 1028
WSA000292 • 1028
WSA000293 • 1028
WSA000294 • 1029
WSA000295 • 1029
WSA000296 • 1029
WSA000297 • 1029
WSA000298 • 1030
WSA000299 • 1030
WSA000300 • 1030
WSA000301 • 1030
WSA000302 • 1031
WSA000303 • 1031
WSA000304 • 1031
WSA000305 • 1031
WSA000306 • 1032
WSA000307 • 1032
WSA000308 • 1032
WSA000309 • 1032
WSA000310 • 1033
WSA000311 • 1033
WSA000312 • 1033
WSA000313 • 1033
WSA000314 • 1034
WSA000315 • 1034
WSA000316 • 1034
WSA000317 • 1035
WSA000318 • 1035
WSA000319 • 1035

WSA000320 • 1035
WSA000321 • 1036
WSA000322 • 1036
WSA000323 • 1036
WSA000324 • 1036
WSA000325 • 1037
WSA000326 • 1037
WSA000327 • 1037
WSA000328 • 1037
WSA000329 • 1038
WSA000330 • 1038
WSA000331 • 1038
WSA000332 • 1038
WSA000333 • 1039
WSA000334 • 1039
WSA000336 • 1039
WSA000337 • 1039
WSA000338 • 1040
WSA000339 • 1040
WSA000340 • 1040
WSA000341 • 1040
WSA000342 • 1041
WSA000343 • 1041
WSA000344 • 1041
WSA000347 • 1041
WSA000348 • 1042
WSA000349 • 1042
WSA000350 • 1042
WSA000351 • 1043
WSA000352 • 1043
WSA000353 • 1043
WSA000354 • 1043
WSA000355 • 1044
WSA000356 • 1044
WSA000357 • 1044
WSA000358 • 1045
WSA000359 • 1045
WSA000360 • 1045
WSA000361 • 1045
WSA000362 • 1046
WSA000363 • 1046
WSA000364 • 1046
WSA000365 • 1046
WSA000366 • 1047
WSA000367 • 1047
WSA000368 • 1047
WSA000369 • 1047
WSA000370 • 1048
WSA000371 • 1048

WSA000372 • 1048
WSA000373 • 1049
WSA000376 • 1050
WSA000377 • 1050
WSA000378 • 1050
WSA000379 • 1051
WSA000380 • 1051
WSA000381 • 1051
WSA000382 • 1052
WSA000383 • 1052
WSA000384 • 1052
WSA000385 • 1052
WSA000386 • 1053
WSA000387 • 1053
WSA000388 • 1053
WSA000389 • 1054
WSA000390 • 1054
WSA000392 • 1049
WSA000393 • 1049
WSA000394 • 1088
WSA000395 • 1089
WSA000396 • 1089
WSA000397 • 1089
WSA000398 • 1089
WSA000399 • 1090
WSA000400 • 1091
WSA000401 • 1092
WSA000402 • 1092
WSA000403 • 1092
WSA000404 • 1093
WSA000405 • 1093
WSA000406 • 1093
WSA000408 • 1090
WSA000409 • 1091
WSA000410 • 1091
WSA001001 • 1054
WSA001002 • 1055
WSAuthenticate • 142
WSAuthenticateResponse • 914
WSChallenge • 138
WSChallenge2 • 140
WSChallengeResponse • 914