Data Access Web Service API Guide

Version 2024.3

Table of Contents

Disclaimer

Copyright © 2025, Ivanti, Inc. All rights reserved.

Introduction

This document details the web service interfaces exposed by the UWM Management Server.

Warning

Ivanti reserves the right to modify any API classes or method signatures without warning.

Users are advised that changes to the API will occur over the span of releases and updates, and that any scripts that use these APIs should be carefully tested with new versions of UWM products before being deployed in a production environment.

Since the Management Console uses the same API to communicate with the Management Server, anything possible within the Management Console is also possible through the API.

Management Console

Proxy DLL

Third party tools may access the APIs directly via the web services as described in this document.

Authorization & Security

Only users with the authorization to connect and use the Management Server can access the web services API.

Limitations

Some types are not described in this document, as they are based on standard data types defined in the .NET framework. Users looking for documentation on DataSet types should refer to MSDN for further details: http://msdn.microsoft.com/en-us/library/system.data.datatable.aspx.

How to use the Data Access Web Service - Examples

DataAccess Endpoint Tasks

```
# Management Server URL
$url = "http://localhost:7751/ManagementServer/DataAccess"

# Create connection to the Management Server
try
{
    # Connect to Required Web Services
    $groupsWebService = New-WebServiceProxy -Uri $url/Groups.asmx -UseDefaultCredential -
ErrorAction Stop

    Write-Host  "Successfully Connected to Web Service Proxy for Management Server"
}
catch
{
    Write-Host "Failed to Connect to Management Server, Exiting Script"
    Break
}

# Get list of deployment groups
$groupsDataSet = $groupsWebService.GetGroups($True)
$groups = $groupsDataSet.Groups
```

Managing Machines

The Management Server splits machines into two categories:

- DiscoveredMachines (machines which have been discovered through membership rules or been manually added by a user – they may not have completed the CCA installation process).
- Machines (machines which have completed the CCA installation process – these machines will also have a DiscoveredMachines instance).

Machines are managed using the DiscoveredMachinesWebService and the MachineWebService.

```
# Management Server URL
$url = "http://localhost:7751/ManagementServer/DataAccess"

# Create connection to the Management Server
try
{
    # Connect to Required Web Services
    $machinesWebService = New-WebServiceProxy -Uri $url/machines.asmx -UseDefaultCredential -
ErrorAction Stop
    $discoveredMachinesWebService = New-WebServiceProxy -Uri $url/discoveredmachines.asmx -
UseDefaultCredential -ErrorAction Stop
    $groupsWebService = New-WebServiceProxy -Uri $url/Groups.asmx -UseDefaultCredential -
ErrorAction Stop

    Write-Host  "Successfully Connected to Management Server"
}
catch
{
    Write-Host "Failed to Connect to Management Server, Exiting Script"
    Break
```

```
}

$withSummary = $true

# Get list of machines
$machinesDataSet = $machinesWebService.GetMachines($withSummary)
$machines = $machinesDataSet.Machines

# Get default group
$firstGroup = $groupsWebService.GetGroups($withSummary).Groups[0]

# Get list of user specified machines in the first g
$discoveredMachinesDataSet =
$discoveredMachinesWebService.GetUserSpecifiedMachinesFromGroupKey($firstGroup.GroupKey)
$userSpecifiedMachines = $discoveredMachinesDataSet.DiscoveredMachines
```

Retrieving Alerts

Alerts are managed using the AlertWebService.

```
# Management Server URL
$url = "http://localhost:7751/ManagementServer/DataAccess"

# Create connection to the Management Server
try
{
    # Connect to Required Web Services
    $alertsWebService = New-WebServiceProxy -Uri $url/alerts.asmx -UseDefaultCredential -
ErrorAction Stop

    Write-Host  "Successfully Connected to Management Server"
}
catch
{
    Write-Host "Failed to Connect to Management Server, Exiting Script"
    Break
}

$namespace = $alertsWebService.GetType().Namespace
$alertRuleSeverity = new-object -TypeName "${n}.AlertRuleSeverity"

$alertsWebService.DeleteAlertsFromSeverity($alertRuleSeverity::Medium)
```

DataRows - Alerts

Alerts

AlertsRow

When an event is correctly matched against an alert rule, the event is added to an existing alert if one exists with the correct conditions, or to a new alert. The conditions for matching an existing alert require that the status is new, and that the alert was generated for the same group and machine. The status column can be 0 for new, 1 for acknowledged and 2 for resolved.

Column	DataType	Description
AlertKey	Int32	Unique identifier relating to alert
AlertRuleKey	Guid	Unique identifier relating to alert rule
AlertRuleName	String	Name of alert rule
GroupKey	Guid	Unique identifier relating to group
GroupName	String	Name of group
MachineKey	Guid	Unique identifier relating to machine
MachineName	String	Name of machine
Status	Int32	
Time	DateTime	
MillisecondsOfDay	Int32	
Severity	Int32	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified

AlertRulesRow

Represents the definition of an alert rule as seen within the Management Console. It consists of a name and description, as well as a boolean enabled column and a severity. The severity can be 0 for critical, 1 for high, 2 for medium and 3 for low. EventQuery, MachineQuery, UserQuery and GroupQuery columns contain a regular expression for matching against the respective columns within an event.

Column	DataType	Description
AlertRuleKey	Guid	Unique identifier relating to alert rule
Name	String	Name of alert rule
Description	String	Description of alert rule
Severity	Int32	
Enabled	Boolean	
EventQuery	String	
MachineQuery	String	
UserQuery	String	
GroupQuery	String	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
PolicyKey	Guid	Unique identifier relating to policy
OwnerSid	String	
SecurityDescriptor	String	
Version	Int32	

ActionsRow

Represents the SMTP and SNMP actions of an alert rule which are triggered when an alert is generated.

Column	DataType	Description
ActionKey	Guid	Unique identifier relating to action
AlertRuleKey	Guid	Unique identifier relating to alert rule
Name	String	Name of action
Description	String	Description of action
Type	String	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified

ConfigurationsRow

Stores the settings as name, value pairs for each action within an alert rule.

Column	DataType	Description
ActionKey	Guid	Unique identifier relating to action
Name	String	Name of configuration
Value	String	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
ActionsRow	ActionsRow	

DataRows - Conditions

Conditions

ConditionsRow

Represents a condition on the server.

Column	DataType	Description
ConditionPK	Int32	
GroupKey	Guid	Unique identifier relating to group
ConditionType	Byte	The values of which are: 0 – NetBios 1 – Container 2 – ComputerName 3 – Domain 4 – All; used for the default group only
IsInclude	Boolean	
IncludeChildren	Boolean	
ADObjectGUID	Guid	
ADObjectDistinguishedName	String	Name of ad object distinguished
ModifiedTime	DateTime	Time modified
Filter	String	
Domain	String	

DataRows - Database

Database

NamedValuesRow

Used to store general values that the Management Center uses such as the database schema version number.

Column	DataType	Description
Name	String	Name of named value
Value	String	

DataRows - Deployment

Deployment

CredentialsRow

Represents a credential on the server.

Column	DataType	Description
UserName	String	Name of user
Password	String	
Index	Int32	
CredentialsKey	Guid	Unique identifier relating to credentials.
GroupKey	Guid	Unique identifier relating to group.
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified

InstructionsRow

The commandID column can represent either a install CCA instruction (128) or a poll now instruction (129).

Column	DataType	Description
CommandID	Int32	
GroupKey	Guid	Unique identifier relating to group.
PlugIn	String	
Status	String	
MachineKey	Guid	Unique identifier relating to machine.
InstructionKey	Guid	Unique identifier relating to instruction.
Settings	String	

StatusHistoryRow

Represents a status history on the server.

Column	DataType	Description
InstructionKey	Guid	Unique identifier relating to instruction.
Status	String	
CreationTime	DateTime	Time created
InstructionsRow	InstructionsRow	

DataRows - DiscoveredMachines

DiscoveredMachines

DiscoveredMachinesRow

Represents a discovered machine on the server.

Column	DataType	Description
DiscoveredMachineKey	Guid	Unique identifier relating to discovered machine.
ExpectedGroupKey	Guid	Unique identifier relating to expected group.
NetBiosName	String	Name of net bios
ADObjectGuid	Guid	
Description	String	Description of discovered machine.
OperatingSystem	String	
DNS	String	
UserSpecified	Bit	
CreationTime	DateTime	Time created
ModifiedDiscoveryTime	DateTime	
ModifiedDeploymentTime	DateTime	

SchedulerConfigurationDto

Contains the configuration for a service running under the Scheduler Service.

Column	DataType	Description
TaskId	Guid	Unique identifier of a scheduler service.
AssemblyName	String	
TypeName	String	
DefaultInterval	Int32	
ScheduledInterval	Int32	
EventName	String	
StartMode	Int32	1 = automatic, 0 = manual
LastStartMode	Int32	
Timeout	Int32	
ActiveServer	String	
LastStarted	DateTime	
LastUpdated	DateTime	
LastCompleted	DateTime	

DataRows - Events

Events

EventRow

Stores the events that have been raised by the product agents on the managed machines.

Column	DataType	Description
EventKey	Int32	Unique identifier relating to event.
EventDefinitionKey	Int32	Unique identifier relating to event definition.
MachineKey	Guid	Unique identifier relating to machine.
MachineName	String	Name of machine.
GroupKey	Guid	Unique identifier relating to group.
GroupName	String	Name of group.
UserName	String	Name of user.
Time	DateTime	
MillisecondsOfDay	Int32	

ParamRow

Represents a param on the server.

Column	DataType	Description
EventDefinitionParamKey	Int32	Unique identifier relating to event definition param.
EventKey	Int32	Unique identifier relating to event.
Type	Int32	

Name	String	Name of param.
Value	String	
EventRow	EventRow	

EventDefinitionRow

Stores the actual event definition which is associated to a product. An event definition can also have one or more parameters associated to the event. The primary key of the event definition is displayed within the Management Console as an Event ID.

Column	DataType	Description
EventDefinitionKey	Int32	Unique identifier relating to event definition.
Name	String	Name of event definition.
Description	String	Description of event definition.
EventDescription	String	
ProductKey	Guid	Unique identifier relating to product.
ProductName	String	Name of product.
Type	Int32	
HighVolume	Boolean	
DefaultEnabledState	Boolean	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified

EventDefinitionParamRow

Each event definition contains a list of parameters, which are defined by name, type and description columns. The type column represents the data type, and can be 0 for a string, 1 for an integer or 2 for a date or time.

Column	DataType	Description
EventDefinitionParamKey	Int32	Unique identifier relating to event definition param.
EventDefinitionKey	Int32	Unique identifier relating to event definition.
Name	String	Name of event definition param.
Description	String	Description of event definition param.
Type	Int32	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
EventDefinitionRow	EventDefinitionRow	

EventParameterValuesRow

Stores an event's string/integer/date/time/etc parameters, as defined by the EventDefinitionParam table.

Column	DataType	Description
StringValue	String	
IntegerValue	Int32	
DateTimeValue	DateTime	

DataRows - Groups

Groups

ScheduleRow

Stores the installation schedule for machines which belong to this group. The flags column can be 0 for disabled, 1 to use the daily time slots, 2 for immediate and 3 for computer startup. Each day of the week gets a <DayOfWeek>Flags column, and there are separate columns for agent/configuration schedules (configurations are stored in <DayOfWeek>ConfigFlags columns). A single bit represents if deployment is enabled for the specified day:

Flags	Binary
9007199254740992	10000000000000000000000000000000000000000000000000000000

Schedules are split into 30-minute intervals, and each bit represents if agents/configs can be deployed in these half hour intervals. Bit 53 represents whether the schedule is enabled. The console only allows one continuous period to be specified (using a start/end time). The least significant bit of the flag represents the first half hour of the day:

Start	End	Flags	Binary
00:00	00:30	9007199254740993	10000000000000000000000000000000000000000000000000000001

Additional bits starting from the least significant end add enabled additional hours for deployments:

Start	End	Flags	Binary
00:00	00:30	9007199254740993	10000000000000000000000000000000000000000000000000000001
00:00	01:30	9007199254740999	10000000000000000000000000000000000000000000000000000111

00:00	02:00	9007199254741007	10000000000000000000000000000000000000000000000001111
00:00	02:00	15 (disabled)	00000000000000000000000000000000000000000000000001111
00:30	02:00	9007199254741006	10000000000000000000000000000000000000000000000001110

End times can be before start times to allow deployment schedules from the start of the day until the end time followed by a second period from the start time until the end of the day.

Start	End	Flags	Binary
18:00	08:00	9288605512040447	10000011111111111100000000000000000001111111111111111

The set of <DayOfWeek>ConfigFlags columns define the installation schedule for configurations (as opposed to agents). In previous versions, the behavior was always as if the ConfigFlags column was set to "2 - Immediate". The DownloadConfigurationsOnStartup property determines whether startup should be blocked whilst the CCA downloads new configurations and installs them before allowing the user to login.

Column	DataType	Description
GroupKey	Guid	Unique identifier relating to group.
MondayFlags	Int64	
TuesdayFlags	Int64	
WednesdayFlags	Int64	
ThursdayFlags	Int64	
FridayFlags	Int64	

SaturdayFlags	Int64	
SundayFlags	Int64	
Flags	Int32	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
RetryPeriodSeconds	Int32	
MondayConfigFlags	Int64	
TuesdayConfigFlags	Int64	
WednesdayConfigFlags	Int64	
ThursdayConfigFlags	Int64	
FridayConfigFlags	Int64	
SaturdayConfigFlags	Int64	
SundayConfigFlags	Int64	
ConfigFlags	Int32	
PostponeInstallationInSchedule	Boolean	
PostponeInstallationLimit	Int32	
DownloadConfigurationsOnStartup	Boolean	Determines whether startup should be blocked whilst the CCA downloads new configurations and installs them before allowing the user to login.

GroupsRow

Represents a group on the server.

Column	DataType	Description
groupKey	Guid	The key which identifies this group.
name	String	The name of the group.
description	String	Description of the group.
pollPeriodSeconds	Int32	The amount of seconds between polls.
uploadPollPeriodSeconds	Int32	The amount of seconds between uploads.
eventLogEnabled	Boolean	Whether to log events to the event log.
fileLogEnabled	Boolean	Whether to log events to the file log.
fileLogFilename	String	The file name to log events to.
anonymousUserLogging	Boolean	Whether events from machines within this group will be logged with anonymous users.
anonymousMachineLogging	Boolean	Whether events from machines within this group will be logged with anonymous machines.
overrideServerUrls	Boolean	Whether this group overrides server URL's.
selfRegistrationEnabled	Boolean	Whether or not self-registration is allowed.
selfUnregistrationEnabled	Boolean	Whether or not self unregistration is allowed.
selfUpdateEnabled	Boolean	Whether or not self-update of agents and configurations is allowed.
priority	Int32	Order in which this groups membership rules will be evaluated.
pollPeriodVariationSeconds	Int32	The VariationSeconds allowed for each poll
uploadPollPeriodVariationSeconds	Int32	The VariationSeconds allowed for each upload.

nativeConfigurations	Boolean	Set to true whenever the CCA is deploying configurations as the native aemp/aamp/apmp file format.
configurationLocation	String	When nativeConfigurations is true, defines the location that native configurations will be deployed to.

GroupPackagesRow

Stores which packages have been assigned for deployment to the machines within the group.

Column	DataType	Description
GroupKey	Guid	Unique identifier relating to group.
PackageKey	Guid	Unique identifier relating to package.
Name	String	Name of group package.
Type	String	
Company	String	
Platform	Int32	
ProductName	String	Name of product.
Major	Int32	
Minor	Int32	
Build	Int32	
Revision	Int32	
Exists	Boolean	
CertificateKey	Guid	Unique identifier of the associated PFX file (certificate).
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified

EventFilterRow

Stores a list of event id's (the primary key of the event definition) which are enabled for each deployment group. This list represents a deployment group's event filter within the Management Console.

Column	DataType	Description
GroupKey	Guid	Unique identifier relating to group.
EventDefinitionKey	Int32	Unique identifier relating to event definition.
Name	String	Name of event filter.
Description	String	Description of event filter.
ProductName	String	Name of product.
Enabled	Boolean	
HighVolume	Boolean	
DefaultEnabledState	Boolean	

EventFilterRow

Used to store general values that the Management Center uses such as the database schema version number.

Column	DataType	Description
Name	String	Name of named value.
Value	String	

StatisticsDto

Used to present the data on the home page of the Management Console.

Column	DataType	Description
Groups	String	Total number of groups.
GroupsDeployed	String	Number of groups that are fully deployed.
GroupsInError	String	Number of groups containing computers in an error state.
Computers	String	Total number of computers.
ComputersDeployed	String	Number of computers that are fully deployed.
ComputersInError	String	Number of computers that are in an error state.
ComputersOffline	String	Number of computers that are not polling in.
Alerts	String	Number of recorded alerts.
AlertsCritical	String	Number of recorded critical alerts.
AlertsNew	String	Number of new alerts.
AlertsNew24H	String	Number of alerts that have been recorded in the last 24 hours.

DataRows - Licenses

Licenses

LicensesRow

Represents a license on the server.

Column	DataType	Description
LicenseCode	String	
ActivationCode	String	
LicenseType	Int32	
ProductKey	Guid	Unique identifier relating to product.
LicenseCount	Int32	
ExpiryDate	DateTime	
BaseLicense	Boolean	
GroupKey	String	Unique identifier relating to group.
LicenseKey	Guid	Unique identifier relating to license.
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified

DataRows - Machines

Machines

MachinesRow

Provides data on each machine in the Management Center. The Platform column contains either 1 for a 32-bit machine, or 2 for a 64-bit machine. The GroupFK column is a foreign key relating to the group that the machine is a member of. This has the value of null for machines in the unassigned group.

Column	DataType	Description
MachineKey	Guid	Unique identifier relating to machine.
GroupKey	Guid	Unique identifier relating to group.
GroupName	String	Name of group.
Platform	Int32	
NetBiosName	String	Name of net bios.
DistinguishedName	String	Name of distinguished.
OldDistinguishedName	String	Name of old distinguished.
ObjectGuid	Guid	
LastPollTime	DateTime	
LastPollStatus	Int32	
LastUploadTime	DateTime	
LastUploadStatus	Int32	
IsPendingDeletion	Boolean	
AlertCount	Int32	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
ModifiedGroupTime	DateTime	

DiagnosticsError	Boolean	
DiagnosticsState	Int32	
DiagnosticsTime	DateTime	
Deployed	Int32	
DeployError	Boolean	
Offline	Boolean	
DNS	String	
LastResponseSeconds	Int32	

MachinePackagesRow

The CCA detects the installation state of all packages which have been added to the Management Center's database. This information is sent to the Management Server when the CCA polls, and is stored in the MachinePackages table. The Status column indicates the progress through the installation of the package:

- Pending Install
- Checking Prerequisites
- Downloading
- Download Failed
- Installing
- Installed
- Install Failed
- Pending Upgrade
- Upgrade Failed
- Pending Uninstall
- Uninstalling
- Uninstall Failed
- Uninstalled

- Install Prerequisite Failed
- Unmanaged

The StatusMessage column will contain an error message if the Status column is a failure.

Column	DataType	Description
MachineKey	Guid	Unique identifier relating to machine.
PackageKey	Guid	Unique identifier relating to package.
Major	Int32	
Minor	Int32	
Build	Int32	
Revision	Int32	
Name	String	Name of machine package.
Company	String	
Type	String	
Platform	Int32	
ProductName	String	Name of product.
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
Status	Int32	
StatusMessage	String	
ChildStatus	Int32	
ChildMajor	Int32	
ChildMinor	Int32	
ChildBuild	Int32	
ChildRevision	Int32	
CertificateKey	Guid	Unique identifier of the associated PFX file (certificate).

CertificateThumbprints	String	Comma separated list of thumbprints for the certificates in the PFX package (certificate).
CertificatePasssword	String	Encrypted password to open the PFX package.

MatchResultsRow

Represents a match result on the server.

Column	DataType	Description
GroupKey	Guid	Unique identifier relating to group.
GroupName	String	Name of group.
MatchName	String	Name of match.
Difference	Int32	

MachineDetailsRow

Stores a collection of name \ value pairs containing the machines details such as OS, memory, cpu platform 32 or 64 bit etc.

Column	DataType	Description
MachineKey	Guid	Unique identifier relating to machine.
Name	String	Name of machine detail.
Value	String	

MachineDiagnosticsRow

Stores the results of any diagnostic tests that have been performed on the machine.

Column	DataType	Description
MachineKey	Guid	Unique identifier relating to machine.
ServerUrl	String	
Name	String	Name of machine diagnostic.
Error	Boolean	
Message	String	

DataRows - Packages

Packages

PackagesRow

Stores the version independent properties of a package, such as the platform and type. The platform column can be 0 for platform independent, 1 for 32-bit and 2 for 64-bit. The type column can be "msi/configuration" for configurations, and "msi/agent" for agents. If the package has been locked by a user, then the Locked column is set to 1 and the LockedUserName set to the name of the user who owns the lock.

Column	DataType	Description
LatestName	String	Name of latest
PackageKey	Guid	UpgradeCode property defined in associated MSI files.
Company	String	
Type	String	
Platform	Int32	
ProductKey	Guid	Unique identifier relating to product.
ProductName	String	Name of product.
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
OwnerSid	String	
PolicyKey	Guid	Unique identifier relating to policy.
Locked	Boolean	
LockedUserName	String	Name of locked user.
LatestMajor	Int32	
LatestMinor	Int32	
LatestBuild	Int32	
LatestRevision	Int32	

SecurityDescriptor	String	
Dirty	Byte	
CertificateKey	Guid	Unique identifier of the associated PFX certificate package.

PackageVersionsRow

Stores the actual data for each version of a package. The name is stored on a per package basis to accommodate renames of configs and tags of agents (such as beta). The Major, Minor, Build and Revision fields form the unique version number for the package. The InProgress column identifies a version of a package which is currently being modified, and hence should not be deployed. The creator versions store the version number of the console which created a configuration, and the dependent minimum and maximum columns representing the minimum and maximum versions of agents that the configuration is compatible with.

Column	DataType	Description
PackageVersionKey	Guid	ProductCode property defined in associated MSI files.
PackageKey	Guid	UpgradeCode property defined in associated MSI files.
Name	String	Name of package version.
Major	Int32	Major version of package.
Minor	Int32	Minor version of package.
Build	Int32	Build version of package.
Revision	Int32	Revision version of package.
DataLength	Int32	Size of package.
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
InProgress	Boolean	
UserName	String	Name of user.
Description	String	Description of package version.

CreatorMajor	Int32	Major version of package creator (console).
CreatorMinor	Int32	Minor version of package creator (console).
CreatorBuild	Int32	Build version of package creator (console).
CreatorRevision	Int32	Revision version of package creator (console).
DependentMinimumMajor	Int32	Major version of minimum associated agent.
DependentMinimumMinor	Int32	Minor version of minimum associated agent.
DependentMinimumBuild	Int32	Build version of minimum associated agent.
DependentMinimumRevision	Int32	Revision version of minimum associated agent.
DependentMaximumMajor	Int32	Major version of maximum associated agent.
DependentMaximumMinor	Int32	Minor version of maximum associated agent.
DependentMaximumBuild	Int32	Build version of maximum associated agent.
DependentMaximumRevision	Int32	Revision version of maximum associated agent.
PackagesRow	PackagesRow	

PatchesRow

Stores meta-data for a patch, including the package version that the patch applies.

Column	DataType	Description
PatchKey	Guid	The unique identifier of the patch.
PackageVersionKey	Guid	The package version that this patch applies to.
PatchCode	Guid	The Patch Code property of the Windows Installer MSP file.
Name	String	Name of the patch.
Major	Int32	Major version of package once this patch is applied.
Minor	Int32	Minor version of package once this patch is applied.
Build	Int32	Build version of package once this patch is applied.
Revision	Int32	Revision version of package once this patch is applied.
DataLength	Int32	Size of patch.
InProgress	Boolean	True whenever this patch is being updated.
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
TargetMajor	Int32	The version of the patch or package that this patch applies to.
TargetMinor	Int32	The version of the patch or package that this patch applies to.
TargetBuild	Int32	The version of the patch or package that this patch applies to.

TargetRevision	Int32	The version of the patch or package that this patch applies to.
ValidationFlags	Int32	The Validation Flags property of the Windows Installer MSP file.

CertificatesRow

Stores meta-data for a PFX certificate package, including the thumbprints of the contained certificates and the password for the package.

Column	DataType	Description
CertificateKey	Guid	The unique identifier of the PFX certificate package.
PackageKey	Guid	The package key that this certificate is associated with.
Name	String	The name of the PFX certificate file.
DataLength	Int32	The size of the PFX certificate file.
CreationTime	DateTime	The date and time when the PFX certificate file was uploaded.
ModifiedTime	DateTime	The date and time that the certificate row was last modified.
InProgress	Boolean	Set to true when the PFX certificate file is being uploaded.
UserName	String	The name of the user that uploaded the PFX certificate file.
Description	String	Description of the PFX certificate file.
Password	String	The encrypted password for the PFX certificate file.
Thumbprints	String	Comma separated list of thumbprints of the certificates contained in the PFX certificate file.

EarliestExpiry	DateTime	The date and time of the earliest expiry time of the contained certificates.
IsDeployed	Boolean	Set to true when the PFX certificate file is assigned to a group.

PrerequisitesRow

Represents a prerequisite on the server.

Column	DataType	Description
PrerequisitesKey	Guid	Unique identifier relating to prerequisites.
Name	String	Name of prerequisite.
Version	Int32	
PlatformInfo	String	
VersionInfo	String	
ModifiedTime	DateTime	Time modified
CreationTime	DateTime	Time created

PackageVersionPrerequisitesRow

Represents a package version prerequisite on the server.

Column	DataType	Description
PackageVersionKey	Guid	Unique identifier relating to package version.
PrerequisitesKey	Guid	Unique identifier relating to prerequisites.
PrerequisitesRow	PrerequisitesRow	

PatchPrerequisitesRow

Represents a patch prerequisite on the server.

Column	DataType	Description
PatchKey	Guid	Unique identifier relating to patch.
PrerequisitesKey	Guid	Unique identifier relating to prerequisites.
PrerequisitesRow	PrerequisitesRow	

PrerequisiteResourceRow

Represents a prerequisite resource on the server.

Column	DataType	Description
ResourceKey	Guid	Unique identifier relating to resource.
PrerequisiteKey	Guid	Unique identifier relating to prerequisite.
Destination	String	
DataLength	Int32	
HashCode	String	
Valid	Boolean	
ModifiedTime	DateTime	Time modified
CreationTime	DateTime	Time created
PrerequisitesRow	PrerequisitesRow	

PrerequisiteCommandRow

Represents a prerequisite command on the server.

Column	DataType	Description
CommandKey	Guid	Unique identifier relating to command.
PrerequisiteKey	Guid	Unique identifier relating to prerequisite.
Action	Int32	
Type	Int32	
Path	String	
Arguments	String	
DefaultResult	Int32	
ModifiedTime	DateTime	Time modified
CreationTime	DateTime	Time created
PrerequisitesRow	PrerequisitesRow	

PrerequisiteCheckRow

Represents a prerequisite check on the server.

Column	DataType	Description
PrerequisiteCheckKey	Guid	Unique identifier relating to prerequisite check.
PrerequisiteKey	Guid	Unique identifier relating to prerequisite.
CheckType	Int32	
Condition	Int32	
OperatorValue	Int32	
Data	String	

Product_ProductCode	Guid	
Product_UpgradeCode	Guid	
File_Path	String	
Registry_Root	String	
Registry_Key	String	Unique identifier relating to registry_.
Registry_Value	String	
OperatingSystem_Message	String	
ModifiedTime	DateTime	Time modified
CreationTime	DateTime	Time created
PrerequisitesRow	PrerequisitesRow	

PrerequisiteExitCodeRow

Represents a prerequisite exit code on the server.

Column	DataType	Description
ExitCodeKey	Guid	Unique identifier relating to exit code.
Value	Int32	
Result	String	
CommandKey	Guid	Unique identifier relating to command.
ModifiedTime	DateTime	Time modified
CreationTime	DateTime	Time created
PrerequisiteCommandRow	PrerequisiteCommandRow	

DataRows - Products

Products

ProductsRow

Used to identify the product that packages, event definitions and reports belong to.

Column	DataType	Description
ProductKey	Guid	Unique identifier relating to product.
Name	String	Name of product.
Icon	Byte[]	
SupportsAgents	Boolean	
SupportsConfigurations	Boolean	
SupportsSoftware	Boolean	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified

DataRows - Reports

Reports

ReportDefinitionsRow

Represents a report definition on the server.

Column	DataType	Description
ReportDefinitionKey	Guid	Unique identifier relating to report definition.
Name	String	Name of report definition.
Description	String	Description of report definition.
Category	String	
Type	String	
ProductKey	Guid	Unique identifier relating to product.
ProductName	String	Name of product.
Visible	Boolean	
DataLength	Int32	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
PolicyKey	Guid	Unique identifier relating to policy.
OwnerSid	String	
SecurityDescriptor	String	

DataRows - Security

Security

SecurityRolesRow

Stores a list of security roles. A security role defines a grouping of privileges within the Management Center identified by the PermissionsMask. The role type can be 0 for a server role, or 1 for an object role.

Column	DataType	Description
SecurityRoleKey	Guid	Unique identifier relating to security role.
Name	String	Name of security role.
Description	String	Description of security role.
PermissionsMask	Int64	
RoleType	Int32	
ModifiedTime	DateTime	Time modified
ReadOnly	Boolean	

ServerPermissionsRow

Represents a server permission on the server.

Column	DataType	Description
Name	String	Name of server permission.
Mask	Int64	

ObjectPermissionsRow

Represents an object permission on the server.

Column	DataType	Description
Name	String	Name of object permission.
Mask	Int64	

UsersRow

Stores a list of users which have access to the Management Server. If IsGroup is 1, then the user is a user group. The PolicyFK reference to the Policies table is used to store server wide roles for the user.

Column	DataType	Description
UserKey	Guid	Unique identifier relating to user.
Name	String	Name of user.
Sid	String	
IsGroup	Boolean	
IsMember	Boolean	
PolicyKey	Guid	Unique identifier relating to policy.
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
SecurityDescriptor	String	

SecurityElementsRow

Associates an object's policy with an individual user's security roles. ElementType can be either 0 for allow, or 1 for deny.

Column	DataType	Description
SecurityElementKey	Guid	Unique identifier relating to security element.
ElementType	Int32	
PolicyKey	Guid	Unique identifier relating to policy.
SecurityRoleKey	Guid	Unique identifier relating to security role.
UserSid	String	
ModifiedTime	DateTime	Time modified

PoliciesRow

Any object which has overridden permissions contains an entry within the Policies table. This table links the object to one or more security elements. Type represents the object type that the policy is for, and can be 0 for users, 1 for group, 2 for packages, 3 for reports and 4 for alert rules.

Column	DataType	Description
PolicyKey	Guid	Unique identifier relating to policy.
Type	Int32	

DataRows - Servers

Servers

ServersRow

Represents a server on the server.

Column	DataType	Description
ServerKey	Guid	Unique identifier relating to server.
GroupKey	Guid	Unique identifier relating to group.
Url	String	
Index	Int32	
PerformDiagnostics	Boolean	
CreationTime	DateTime	Time created
ModifiedTime	DateTime	Time modified
Disabled	Boolean	

Namespace Documentation

DataAccessServices.WebServices Namespace Reference

Classes

- class Alerts

  The Management Center can be configured to generate Alerts which are based on events generated by the different AppSense product agents on the registered client machines. Alerts provide information as to when the event occurred and the severity of the alert, the alert might also generate and SNMP trap or SMTP (email) action when the alert is triggered.

- class **Auditing**

- class ChangeTracking

  The ChangeTrackingWebService provides access to the Change Tracking information.

- class Conditions

  The Conditions WebService creates conditions for use by Machine Discovery. For more information, refer to the DiscoveredMachine WebService page.

- class DatabaseWebService

  Provides methods for retrieving information about the current Management Server database.

- class Deployment

  Manages deployment of the CCA. When the "Install CCA" instruction is selected from the Management Console each client machine has an associated DeploymentInstruction. This is then used to provide the "CCA Install Log" to the ManagementConsole and any deployment status. The "DeploymentCredentials" is a store of the usernames and passwords provided in the ManagementConsole. The passwords are RSA encrypted.

- class DiscoveredMachines

- *Manages discovered machines in the Management Centre. When discovering computer membership within a group the conditions are evaluated for each group. If a machine from the active directory search matches the condition it is then added to the DiscoveredMachines table.*

- class Events

  Manages events in the Management Centre. Within the AppSense Management Suite the product agents can raise a number of different events to the Management Center. In order to receive and display these events the Management Center database contains a list of all the possible events that can be raised via the product agents.

- class Groups

  Manages deployment groups and assigned packages in the Management Centre. The groups table represents a deployment group, with associated settings being stored within the related tables. The GroupPackages and MachinePackages tables represent the packages which are assigned to a group, and are installed on the group's machines.

- class Housekeeping

  Summary description for Events

- class Licenses

  Manages licenses in the Management Centre. When using the AppSense Management Suite a valid license must be used.

- class Machines

  Manages machines in the Management Centre. The Machines table stores an entry for each machine managed by the Management Center.

- class Maintenance

  Manages maintenance tasks in the Management Centre. Within the AppSense Management Suite the product agents can raise a number of different events/alerts to the Management Center. In order to receive and display these events/alerts the Management Center database contains a list of all the possible evnets/alerts that can be raised via the product agents.

- class **PackageDownloader**

- class Packages

  Packages within the Management Center are stored as MSI files and comprise of either an agent or configuration. A package has one or more associated versions to support software and configuration versioning and concurrency control.

- class Products

  The ProductsWebService creates product entries needed for uploading and managing packages. For more information, refer to the PackagesWebService.

- class Queries

  Provides methods for querying the Management Server database via the reporting engine.

- class Reports

  Manages reports in the Management Centre. Report definitions store the necessary data for generating the reports visible within the Management Console.

- class **Resources**

  A strongly-typed resource class, for looking up localized strings, etc.

- class Security

  Within the Management Center it is possible to configure permissions for the different object types; these object types include groups, packages and alerts rules etc. These objects contain an OwnerSid and PolicyFK columns which control the permissions on the objects.

- class Servers

  Within the Management Center database it is possible to configure a number of fail over servers which are used if the current management server cannot be contacted. The user can configure a number of fail over servers on a per group basis allowing different groups to be serviced by different management servers.

Class Documentation

DataAccessServices.WebServices.Alerts Class Reference

The Management Center can be configured to generate Alerts which are based on events generated by the different AppSense product agents on the registered client machines. Alerts provide information as to when the event occurred and the severity of the alert, the alert might also generate and SNMP trap or SMTP (email) action when the alert is triggered.

Public Member Functions

- void AddActionConfiguration (Guid actionKey, String name, String value, out DateTime modifiedTime)

  Adds a configuration property to an action.

- void AddEventToAlert (Int32 alertKey, Int64 eventKey)

  Adds an event to an existing alert.

- void ApplyActionChanges (ref ActionsDataSet actionsChanges)

  Applies changes made to the actions data set into the database.

- void ApplyAlertChanges (ref AlertsDataSet alertChanges)

  Applies edits from within an alerts data set into the database.

- void ApplyAlertRuleChanges (ref AlertRulesDataSet alertRules)

  Applies edits from within the alert rules data set into the database.

- void ApplyAlertRuleSecurityChanges (ref AlertRulesDataSet alertRules)

  Applies edits from within the alert rules data set into the database.

- void CreateAction (Guid actionKey, Guid alertRuleKey, String name, String description, String type, out DateTime modifiedTime)

  Creates an action associated with an alert rule within the database.

- void CreateAlert (Guid alertRuleKey, Guid? groupKey, Guid? machineKey, AlertStatus status, DateTime time, out DateTime modifiedTime)

  Creates a new alert within the database.

- void CreateAlertRule (Guid alertRuleKey, String name, String description, AlertRuleSeverity severity, Boolean enabled, Int32 version, String eventQuery, String machineQuery, String userQuery, String groupQuery, out DateTime modifiedTime)

  Constructs a new alert rule.

- void DeleteAction (Guid actionKey, DateTime? modifiedTime)

  Deletes an existing action from the database.

- void DeleteAlert (Int32 alertKey, DateTime? modifiedTime)

  Deletes an alert from the database.

- void DeleteAlertRule (Guid alertRuleKey, DateTime? modifiedTime)

  Deletes an existing alert rule.
- void DeleteAlertsFromGroupKey (Guid groupKey)

  Deletes all alerts generated by machines within the specified group.
- void DeleteAlertsFromMachineKey (Guid machineKey)

  Deletes all alerts generated by the specified machine.
- void DeleteAlertsFromSeverity (AlertRuleSeverity severity)

  Deletes all alerts with the given severity.
- void DeleteAlertsFromStatus (AlertStatus status)

  Deletes all alerts with the given status.
- void DeleteAlertsFromTime (DateTime? startTime, DateTime? endTime)

  Deletes all alerts generated at the specified times.
- ActionsDataSet GetActionFromAlertRuleKey (Guid alertRuleKey)

  Returns the actions associated with an alert rule.
- AlertRulesDataSet GetAlertRuleFromKey (Guid alertRuleKey)

  Returns a single alert rule from a given key.
- AlertRulesDataSet GetAlertRules ()

  Returns all alert rules.
- AlertsDataSet GetAlerts ()

  Returns all alerts within the database.
- AlertsDataSet GetAlertsFromAlertRuleKey (Guid alertRuleKey)

  Retrieves alerts that were created from an alert rule key.
- AlertsDataSet GetAlertsFromGroupKey (Guid? groupKey)

  Returns alerts that were generated from a machine within a specific group.
- AlertsDataSet GetAlertsFromMachineKey (Guid? machineKey)

  Returns alerts generated by the specific machine.
- AlertsDataSet GetAlertsFromSeverity (AlertRuleSeverity severity)

  Returns all alerts with the given severity.
- AlertsDataSet GetAlertsFromStatus (AlertStatus status)

  Returns all alerts with the given status.
- AlertsDataSet GetAlertsFromTimePeriod (DateTime startTime, DateTime endTime)

  Returns all alerts within the specified time period.
- void RemoveActionConfiguration (Guid actionKey, String name, DateTime? modifiedTime)

Removes an existing configuration property from an action.

- void RemoveEventFromAlert (Int32 alertKey, Int64 eventKey)

  Removes an event which has previously been added to an alert.

- void UpdateAction (Guid actionKey, String name, String description, String type, ref DateTime modifiedTime)

  Updates properties of an action.

- void UpdateActionConfiguration (Guid actionKey, String name, String value, ref DateTime modifiedTime)

  Updates properties of a configuration.

- void UpdateAlert (Int32 alertKey, Guid? groupKey, Guid? machineKey, AlertStatus status, DateTime time, ref DateTime modifiedTime)

  Updates the properties of an existing alert.

- void UpdateAlertRule (Guid alertRuleKey, String name, String description, AlertRuleSeverity severity, Boolean enabled, Int32 version, String eventQuery, String machineQuery, String userQuery, String groupQuery, Guid? policyKey, String ownerSid, ref DateTime modifiedTime)

  Updates the properties of an existing alert rule.

- void UpdateAlertRuleSecurity (Guid alertRuleKey, Guid? policyKey, String ownerSid, ref DateTime modifiedTime)

  Updates security permissions associated with the alert rule.

Detailed Description

The Management Center can be configured to generate Alerts which are based on events generated by the different AppSense product agents on the registered client machines. Alerts provide information as to when the event occurred and the severity of the alert, the alert might also generate and SNMP trap or SMTP (email) action when the alert is triggered.

Member Function Documentation

void DataAccessServices.WebServices.Alerts.AddActionConfiguration (Guid actionKey, String name, String value, out DateTime modifiedTime) `[inline]`

Adds a configuration property to an action.

Parameters:

actionKey	The action key.
name	The unique name of the configuration.
value	The value of the configuration property.
modifiedTime	The time that the configuration was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.AddEventToAlert (Int32 alertKey, Int64 eventKey) `[inline]`

Adds an event to an existing alert.

Parameters:

alertKey	The key of the alert.
eventKey	The key of the event.

void DataAccessServices.WebServices.Alerts.ApplyActionChanges (ref ActionsDataSet actionsChanges) `[inline]`

Applies changes made to the actions data set into the database.

Parameters:

actionsChanges	The data set consisting of changes.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.ApplyAlertChanges (ref AlertsDataSet alertChanges) `[inline]`

Applies edits from within an alerts data set into the database.

Parameters:

alertChanges	The data set consisting of changes.

Requires group or administrative access.

void DataAccessServices.WebServices.Alerts.ApplyAlertRuleChanges (ref AlertRulesDataSet alertRules) `[inline]`

Applies edits from within the alert rules data set into the database.

Parameters:

alertRules	A data set consisting of changes.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.ApplyAlertRuleSecurityChanges (ref AlertRulesDataSet alertRules)[inline]

Applies edits from within the alert rules data set into the database.

Parameters:

alertRules	A data set consisting of changes.

Requires alert, security, or administrative access.

void DataAccessServices.WebServices.Alerts.CreateAction (Guid actionKey, Guid alertRuleKey, String name, String description, String type, out DateTime modifiedTime) `[inline]`

Creates an action associated with an alert rule within the database.

Parameters:

actionKey	The key that identifies the action.
alertRuleKey	The alert rule key that this action is associated with.
name	The name of the action.
description	A description of the action.
type	The type, such as SNMP or SMTP.
modifiedTime	The time that the action was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.CreateAlert (Guid alertRuleKey, Guid? groupKey, Guid? machineKey, AlertStatus status, DateTime time, out DateTime modifiedTime)[inline]

Creates a new alert within the database.

Parameters:

alertRuleKey	The alert rule which defines this alert.
groupKey	The key of the group containing the machine that generated this alert (Optional).

machineKey	The key of the machine which generated this alert.
status	The status of the alert.
time	The time that the alert occurred.
modifiedTime	The time that the alert was last modified.

Returns:

The key of the created alert.

Requires administrative access.

void DataAccessServices.WebServices.Alerts.CreateAlertRule (Guid alertRuleKey, String name, String description, AlertRuleSeverity severity, Boolean enabled, Int32 version, String eventQuery, String machineQuery, String userQuery, String groupQuery, out DateTime modifiedTime) `[inline]`

Constructs a new alert rule.

Parameters:

alertRuleKey	The key that identifies the new alert rule.
name	The display name of the alert rule.
description	A description of the alert rule.
severity	A measure of how important the alert is.
enabled	Whether the alert rule is enabled.
version	Version number of the alert rule.
eventQuery	A query string which matches event definition keys.
machineQuery	A query string which matches machines.
userQuery	A query string which matches users.
groupQuery	A query string which matches a group name.
modifiedTime	The time that the alert rule was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAction (Guid actionKey, DateTime? modifiedTime) `[inline]`

Deletes an existing action from the database.

Parameters:

actionKey	The key that identifies the action to delete.
modifiedTime	The time that the action was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAlert (Int32 alertKey, DateTime? modifiedTime) `[inline]`

Deletes an alert from the database.

Parameters:

alertKey	The key that identifies the alert.
modifiedTime	The time that the alert was last modified.

Requires administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAlertRule (Guid alertRuleKey, DateTime? modifiedTime) `[inline]`

Deletes an existing alert rule.

Parameters:

alertRuleKey	The key that identifies the alert rule to be deleted.
modifiedTime	The time that the alert rule was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAlertsFromGroupKey (Guid groupKey) `[inline]`

Deletes all alerts generated by machines within the specified group.

Parameters:

groupKey	The key that identifies the group.

Requires administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAlertsFromMachineKey (Guid machineKey) `[inline]`

Deletes all alerts generated by the specified machine.

Parameters:

machineKey	The key that identifies the machine.

Requires administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAlertsFromSeverity (AlertRuleSeverity severity)[inline]

Deletes all alerts with the given severity.

Parameters:

severity	The severity of alerts to delete.

Requires administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAlertsFromStatus (AlertStatus status) `[inline]`

Deletes all alerts with the given status.

Parameters:

status	The status of alerts to delete.

Requires administrative access.

void DataAccessServices.WebServices.Alerts.DeleteAlertsFromTime (DateTime? startTime, DateTime? endTime) `[inline]`

Deletes all alerts generated at the specified times.

Parameters:

startTime	Start time of period of alerts to delete.
endTime	End time of period of alerts to delete.

Requires administrative access.

ActionsDataSet DataAccessServices.WebServices.Alerts.GetActionFromAlertRuleKey (Guid alertRuleKey) `[inline]`

Returns the actions associated with an alert rule.

Parameters:

alertRuleKey	The key of the alert rule.

Returns:

A data set consisting of all actions associated with the alert rule.

Requires alert or administrative access.

AlertRulesDataSet DataAccessServices.WebServices.Alerts.GetAlertRuleFromKey (Guid alertRuleKey) `[inline]`

Returns a single alert rule from a given key.

Parameters:

alertRuleKey	The key that identifies the rule to run.

Returns:

A data set consisting of the alert rule.

Requires alert or administrative access.

AlertRulesDataSet DataAccessServices.WebServices.Alerts.GetAlertRules () `[inline]`

Returns all alert rules.

Returns:

A data set consisting of all alert rules.

Requires alert or administrative access.

AlertsDataSet DataAccessServices.WebServices.Alerts.GetAlerts () `[inline]`

Returns all alerts within the database.

Returns:

A data set consisting of all alerts in the database.

Requires alert, group, or administrative access.

AlertsDataSet DataAccessServices.WebServices.Alerts.GetAlertsFromAlertRuleKey (Guid alertRuleKey) `[inline]`

Retrieves alerts that were created from an alert rule key.

Parameters:

alertRuleKey	The key that represents the alert rules.

Returns:

A data set consisting of the alerts.

Requires alert, group, or administrative access.

AlertsDataSet DataAccessServices.WebServices.Alerts.GetAlertsFromGroupKey (Guid? groupKey) `[inline]`

Returns alerts that were generated from a machine within a specific group.

Parameters:

groupKey	The key of the group.

Returns:

A data set consisting of the alerts.

Requires alert, group, or administrative access.

AlertsDataSet DataAccessServices.WebServices.Alerts.GetAlertsFromMachineKey (Guid? machineKey) `[inline]`

Returns alerts generated by the specific machine.

Parameters:

machineKey	The key of the machine.

Returns:

A data set consisting of the alerts.

Requires alert or administrative access.

AlertsDataSet DataAccessServices.WebServices.Alerts.GetAlertsFromSeverity (AlertRuleSeverity severity) `[inline]`

Returns all alerts with the given severity.

Parameters:

severity	The severity of the alerts to return.

Returns:

All alerts with the specified severity.

Requires alert, group, or administrative access.

AlertsDataSet DataAccessServices.WebServices.Alerts.GetAlertsFromStatus (AlertStatus status)[inline]

Returns all alerts with the given status.

Parameters:

status	The status of the alerts to return.

Returns:

All alerts with the specified status.

Requires alert, group, or administrative access.

AlertsDataSet DataAccessServices.WebServices.Alerts.GetAlertsFromTimePeriod (DateTime startTime, DateTime endTime) `[inline]`

Returns all alerts within the specified time period.

Parameters:

startTime	The earliest time alerts are to be returned.
endTime	The latest time that alerts are to be returned.

Returns:

The alerts that were created within the specified time.

Requires alert, group, or administrative access.

void DataAccessServices.WebServices.Alerts.RemoveActionConfiguration (Guid actionKey, String name, DateTime? modifiedTime) `[inline]`

Removes an existing configuration property from an action.

Parameters:

actionKey	The key that identifies the action.
name	The unique name of the configuration.
modifiedTime	The time that the configuration was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.RemoveEventFromAlert (Int32 alertKey, Int64 eventKey) `[inline]`

Removes an event which has previously been added to an alert.

Parameters:

alertKey	The key of the alert.
eventKey	The key of the event.

void DataAccessServices.WebServices.Alerts.UpdateAction (Guid actionKey, String name, String description, String type, ref DateTime modifiedTime) `[inline]`

Updates properties of an action.

Parameters:

actionKey	The key that identifies the action.
name	The name of the action.
description	A description of the action.
type	The type, such as SNMP or SMTP.
modifiedTime	The time that the action was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.UpdateActionConfiguration (Guid actionKey, String name, String value, ref DateTime modifiedTime)[inline]

Updates properties of a configuration.

Parameters:

actionKey	The action key.
name	The unique name of the configuration.
value	The value of the configuration property.
modifiedTime	The time that the configuration was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.UpdateAlert (Int32 alertKey, Guid? groupKey, Guid? machineKey, AlertStatus status, DateTime time, ref DateTime modifiedTime) `[inline]`

Updates the properties of an existing alert.

Parameters:

alertKey	The key of the alert to update.
groupKey	The key of the group containing the machine that generated this alert (Optional).
machineKey	The key of the machine which generated this alert.
status	The status of the alert.
time	The time that the alert occurred.
modifiedTime	The time that the alert was last modified.

Requires administrative access.

void DataAccessServices.WebServices.Alerts.UpdateAlertRule (Guid alertRuleKey, String name, String description, AlertRuleSeverity severity, Boolean enabled, Int32 version, String eventQuery, String machineQuery, String userQuery, String groupQuery, Guid? policyKey, String ownerSid, ref DateTime modifiedTime) `[inline]`

Updates the properties of an existing alert rule.

Parameters:

alertRuleKey	The key that identifies the new alert rule.
name	The display name of the alert rule.
description	A description of the alert rule.
severity	A measure of how important the alert is.
enabled	Whether the alert rule is enabled.
version	Version of the alert rule.
eventQuery	A query string which matches event definition keys.
machineQuery	A query string which matches machines.
userQuery	A query string which matches users.
groupQuery	A query string which matches a group name.
policyKey	The key with identifies the policy associated with this alert rule.
ownerSid	The security identifier associated with the user that owns this entry.
modifiedTime	The time that the alert rule was last modified.

Requires alert or administrative access.

void DataAccessServices.WebServices.Alerts.UpdateAlertRuleSecurity (Guid alertRuleKey, Guid? policyKey, String ownerSid, ref DateTime modifiedTime) `[inline]`

Updates security permissions associated with the alert rule.

Parameters:

alertRuleKey	The key that identifies the new alert rule.
policyKey	The key with identifies the policy associated with this alert rule.
ownerSid	The security identifier associated with the user that owns this entry.
modifiedTime	The time that the alert rule was last modified.

Requires alert, security, or administrative access.

The documentation for this class was generated from the following file:

- Alerts.cs

DataAccessServices.WebServices.ChangeTracking Class Reference

The ChangeTrackingWebService provides access to the Change Tracking information.

Public Member Functions

- void DeleteOlderThan (DateTime? date)

  Delete change tracking entries older that the supplied date, or all entries if the specified date is null
- void EnableChangeTracking (bool enable)

  Enable or disable Change Tracking
- ChangeTrackingDataSet Get ()

  Returns all of the ChangeTracking data
- ChangeTrackingDataSet GetBefore (DateTime date)

  Get change tracking entries that are earlier than the date provided
- long GetHighestKeyBeforeDate (DateTime? date)

  Get change tracking entries that are earlier than the date provided
- ChangeTrackingDataSet GetPage (int pageSize, long lastId)

  Returns a page of the ChangeTracking data
- ChangeTrackingOperation GetTrackingOperation (int operation)

  Convert an int into a ChangeTrackingOperation.
- bool IsChangeTrackingEnabled ()

  Returns true if change tracking is enabled

Detailed Description

The ChangeTrackingWebService provides access to the Change Tracking information.

Member Function Documentation

void DataAccessServices.WebServices.ChangeTracking.DeleteOlderThan (DateTime? date) `[inline]`

Delete change tracking entries older that the supplied date, or all entries if the specified date is null.

Parameters:

date	The date to delete records older than, or all records if it is null.

void DataAccessServices.WebServices.ChangeTracking.EnableChangeTracking (bool enable) `[inline]`

Enable or disable Change Tracking

Parameters:

enable	Set to true to enable Change Tracking.

ChangeTrackingDataSet DataAccessServices.WebServices.ChangeTracking.Get () `[inline]`

Returns all of the ChangeTracking data.

Returns:

A data set consisting of all products.

ChangeTrackingDataSet DataAccessServices.WebServices.ChangeTracking.GetBefore (DateTime date) `[inline]`

Get change tracking entries that are earlier than the date provided.

Parameters:

date	The cut off date.

long DataAccessServices.WebServices.ChangeTracking.GetHighestKeyBeforeDate (DateTime? date) `[inline]`

Get change tracking entries that are earlier than the date provided.

Parameters:

date	The cut off date.

ChangeTrackingDataSet DataAccessServices.WebServices.ChangeTracking.GetPage (int pageSize, long lastId) `[inline]`

Returns a page of the ChangeTracking data.

Parameters:

pageSize	The number of rows to return.
lastId	data starts 1 row before this one.

Returns:

A data set consisting of all products.

ChangeTrackingOperation DataAccessServices.WebServices.ChangeTracking.GetTrackingOperation (int operation) `[inline]`

Convert an int into a ChangeTrackingOperation.

Parameters:

operation	the int value to be converted.

Returns:

The ChangeTrackingOperation represented by the operation int value.

bool DataAccessServices.WebServices.ChangeTracking.IsChangeTrackingEnabled () `[inline]`

Returns true if change tracking is enabled.

The documentation for this class was generated from the following file:

- ChangeTracking.cs

DataAccessServices.WebServices.Conditions Class Reference

The Conditions WebService creates conditions for use by Machine Discovery. For more information, refer to the DiscoveredMachine WebService page.

Public Member Functions

- void ApplyChanges (ref ConditionsDataSet conditions)

  Apply bulk changes to a dataset. Allows Create, Update & Delete operations.

- void CreateCondition (out int conditionKey, Guid groupKey, ConditionType conditionType, bool isInclude, bool? includeChildren, Guid? adObjectGuid, string adObjectDistinguishedName, String domain, String filter, out DateTime modifiedTime)

  Immediately creates a condition of any type.

- void CreateCondition_ComputerGroup (Guid groupKey, bool isInclude, bool includeChildren, Guid adObjectGuid, string adObjectDistinguishedName, String domain, String filter)

  Creates a condition which includes/excludes a set of computers which are in an active directory computer group.

- void CreateCondition_Container (Guid groupKey, bool isInclude, bool includeChildren, Guid adObjectGuid, string adObjectDistinguishedName, String domain, String filter)

  Creates a condition which includes/excludes a set of computers which are in a container.

- void CreateCondition_Domain (Guid groupKey, Boolean isInclude, Boolean includeChildren, Guid adObjectGuid, String adObjectDistinguishedName, String domain, String filter)

  Creates a condition which includes/excludes a set of computers which are in a domain.

- void CreateCondition_NetBIOS (Guid groupKey, bool isInclude, string domain, string filter)

  Creates a condition which includes/excludes a computer by its NetBIOS name.

- void Delete (int conditionKey, Guid groupKey, DateTime modifiedTime)

  Immediately deletes a condition.

- ConditionsDataSet GetConditions (Guid groupKey)

  Retrieves the conditions associated with a group. Conditions determine which machines should belong to a group.

- void Update (int conditionKey, Guid groupKey, ConditionType conditionType, bool isInclude, bool? includeChildren, Guid? adObjectGuid, string adObjectDistinguishedName, String domain, String filter, ref DateTime modifiedTime)

  Method to directly update a condition.

Detailed Description

The Conditions WebService creates conditions for use by Machine Discovery. For more information, refer to the DiscoveredMachine WebService page.

Member Function Documentation

void DataAccessServices.WebServices.Conditions.ApplyChanges (ref ConditionsDataSet conditions) `[inline]`

Apply bulk changes to a dataset. Allows Create, Update & Delete operations.

Parameters:

conditions	The dataset to update.

void DataAccessServices.WebServices.Conditions.CreateCondition (out int conditionKey, Guid groupKey, ConditionType conditionType, bool isInclude, bool? includeChildren, Guid? adObjectGuid, string adObjectDistinguishedName, String domain, String filter, out DateTime modifiedTime) `[inline]`

Immediately creates a condition of any type.

Parameters:

conditionKey	OUT parameter will contain the key of the condition when the call returns. Passed in value unused.
groupKey	The GUID of the group to attach this condition to.
conditionType	The type of the condition, changes how the condition is interpreted.
isInclude	Whether this condition includes or excludes the affected machines. Excludes take priority.
includeChildren	Whether to include child nodes in this condition (OU/Computer groups).
adObjectGuid	The GUID of the node in question (OU/Computer groups).
adObjectDistinguishedName	The distinguished name of the group in question (OU/Computer groups).

domain	The domain of the computer.
filter	Filter for the computer.
modifiedTime	OUT parameter will contain the latest modified time of the condition on return. Passed in value unused.

void DataAccessServices.WebServices.Conditions.CreateCondition_ComputerGroup (Guid groupKey, bool isInclude, bool includeChildren, Guid adObjectGuid, string adObjectDistinguishedName, String domain, String filter) `[inline]`

Creates a condition which includes/excludes a set of computers which are in an active directory computer group.

Parameters:

groupKey	The GUID of the group to attach this condition to.
isInclude	Whether this condition includes or excludes the affected machines. Excludes take priority.
includeChildren	Whether to include child computer groupes in this condition.
adObjectGuid	The GUID of the group in question.
adObjectDistinguishedName	The distinguished name of the group in question.
domain	The domain of the computer.
filter	Filter for the computer.

void DataAccessServices.WebServices.Conditions.CreateCondition_Container (Guid groupKey, bool isInclude, bool includeChildren, Guid adObjectGuid, string adObjectDistinguishedName, String domain, String filter) `[inline]`

Creates a condition which includes/excludes a set of computers which are in a container.

Parameters:

groupKey	The GUID of the group to attach this condition to.
isInclude	Whether this condition includes or excludes the affected machines. Excludes take priority.
includeChildren	Whether to include child containers in this condition.
adObjectGuid	The GUID of the container in question.
adObjectDistinguishedName	The distinguished name of the container in question.
domain	The domain of the computer.
filter	Filter for the computer.

void DataAccessServices.WebServices.Conditions.CreateCondition_Domain (Guid groupKey, Boolean isInclude, Boolean includeChildren, Guid adObjectGuid, String adObjectDistinguishedName, String domain, String filter) `[inline]`

Creates a condition which includes/excludes a set of computers which are in a domain.

Parameters:

groupKey	The GUID of the group to attach this condition to.
isInclude	Whether this condition includes or excludes the affected machines. Excludes take priority.
includeChildren	Whether to include child containers in this condition.
adObjectGuid	The GUID of the container in question.
adObjectDistinguishedName	The distinguished name of the container in question.

domain	The domain of the computer.
filter	Filter for the computer.

void DataAccessServices.WebServices.Conditions.CreateCondition_NetBIOS (Guid groupKey, bool isInclude, string domain, string filter) `[inline]`

Creates a condition which includes/excludes a computer by its NetBIOS name.

Parameters:

groupKey	The GUID of the group to attach this computer to.
isInclude	Whether this condition includes or excludes the affected machines. Excludes take priority.
domain	The domain of the computer.
filter	Filter for the computer.

void DataAccessServices.WebServices.Conditions.Delete (int conditionKey, Guid groupKey, DateTime modifiedTime) `[inline]`

Immediately deletes a condition.

Parameters:

conditionKey	The primary key of the condition to delete.
groupKey	The group the the condition belongs to. Used for security checks.
modifiedTime	Last modified time of the condition. Passed in value must match the existing record in the database. If the values mismatch, the delete will not take effect and an error will be raised.

ConditionsDataSet DataAccessServices.WebServices.Conditions.GetConditions (Guid groupKey) `[inline]`

Retrieves the conditions associated with a group. Conditions determine which machines should belong to a group.

Parameters:

groupKey	The guid of the group in question.

Must have View or FullControl permissions.

Returns:

Returns a data set containing all the conditions associated with the group.

void DataAccessServices.WebServices.Conditions.Update (int conditionKey, Guid groupKey, ConditionType conditionType, bool isInclude, bool? includeChildren, Guid? adObjectGuid, string adObjectDistinguishedName, String domain, String filter, ref DateTime modifiedTime) `[inline]`

Method to directly update a condition.

Parameters:

conditionKey	The primary key of the condition to update.
groupKey	The group the condition belongs to. Used for security checks. You cannot alter which group a condition is on.
conditionType	The type of condition.
isInclude	Whether this condition includes or excludes the affected machines. Excludes take priority.
includeChildren	Whether to include child nodes in this condition (OU/Computer groups).
adObjectGuid	The GUID of the node in question (OU/Computer groups).
adObjectDistinguishedName	The distinguished name of the group in question (OU/Computer groups).
domain	The domain of the computer.
filter	Filter for the computer.
modifiedTime	REF parameter will contain the latest modified time of the condition on return. Passed in value must match the existing record in the database. If the values mismatch, the update will not take effect and an error will be raised.

The documentation for this class was generated from the following file:

- Conditions.cs

DataAccessServices.WebServices.DatabaseWebService Class Reference

Provides methods for retrieving information about the current Management Server database.

Public Member Functions

- NamedValuesDataSet GetInfo ()

  Returns a data set consisting of name/value pairs of information about the database.

- string GetName ()
- NamedValuesDataSet GetNameSpace (string nameSpace)

  Returns a set of server properties in the specified namespace.

- string GetServerError ()

  Returns any errors associated with the server. Verifies that the connection to the database is valid and that the database schema is the correct version number.

- DateTime GetServerTime ()

  Returns the date and time on the server.

- string GetTransportPublicKey ()

  Returns the public key used for communication with the server.

- string GetVersion ()

  Version of this interface

- void SetNameSpace (string nameSpace, NamedValuesDataSet properties)

  Updates a set of server properties with new values in the specified namespace.

Detailed Description

Provides methods for retrieving information about the current Management Server database.

Member Function Documentation

NamedValuesDataSet DataAccessServices.WebServices.DatabaseWebService.GetInfo () [inline]

Returns a data set consisting of name/value pairs of information about the database.

Returns:

Data set consisting of name/value pairs of information about the database.

string DataAccessServices.WebServices.DatabaseWebService.GetName ()

Returns:

The name of this server.

NamedValuesDataSet DataAccessServices.WebServices.DatabaseWebService.GetNameSpace (string nameSpace) [inline]

Returns a set of server properties in the specified namespace.

Parameters:

nameSpace	Namespace from where the properties will be retrieved from.

Returns:

Data set of server properties in the specified namespace.

string DataAccessServices.WebServices.DatabaseWebService.GetServerError () [inline]

Returns any errors associated with the server. Verifies that the connection to the database is valid and that the database schema is the correct version number.

Returns:

Any errors associated with the server as string.

DateTime DataAccessServices.WebServices.DatabaseWebService.GetServerTime () [inline]

Returns the date and time on the server.

Returns:

Returns a DateTime object in UTC format.

string DataAccessServices.WebServices.DatabaseWebService.GetTransportPublicKey () [inline]

Returns the public key used for communication with the server.

Returns:

Public key used for communication with the server as string.

string DataAccessServices.WebServices.DatabaseWebService.GetVersion ()

Version of this interface.

Returns:

void DataAccessServices.WebServices.DatabaseWebService.SetNameSpace (string nameSpace, NamedValuesDataSet properties) [inline]

Updates a set of server properties with new values in the specified namespace.

Parameters:

nameSpace	Namespace from where the properties will be retrieved from.
properties	Data set containing changes to properties.

The documentation for this class was generated from the following file:

- Database.cs

DataAccessServices.WebServices.Deployment Class Reference

Manages deployment of the CCA. When the "Install CCA" instruction is selected from the Management Console each client machine has an associated DeploymentInstruction. This is then used to provide the "CCA Install Log" to the ManagementConsole and any deployment status. The "DeploymentCredentials" is a store of the usernames and passwords provided in the ManagementConsole. The passwords are RSA encrypted.

Public Member Functions

- string ActivateDeploymentService (int commandID)

  Activate Deployment Service

- void ApplyChanges (ref DeploymentDataSet deploymentChanges)

  Updates the database with the changes in the Credentials table.

- void ApplyInstructionsChanges (ref DeploymentDataSet deploymentChanges, Guid? groupKey)

  Updates the database with the changes in the Instructions table.

- void ClearStatusHistory (Guid instructionKey, Guid? groupKey)

  Clear status history

- void CreateCredentials (Guid credentialsKey, Guid? groupKey, string userName, string password, int index, out DateTime modifiedTime)

  Creates new credentials. To encrypt the password use DatabaseWebService.GetTransportPublicKey to retrieve the public key and encrypt using RSACryptoServiceProvider.Encrypt.

- void CreateInstructions (Guid instructionKey, int commandID, Guid? groupKey, Guid? machineKey, string status, out DateTime modifiedTime)

  Creates a new deployment instruction.

- DeploymentDataSet GetDeploymentCredentials ()

  Returns a data set containing all deployment credentials.

- DeploymentDataSet GetDeploymentCredentialsFromGroupKey (Guid? groupKey)

  Returns a data set consisting of all credentials for a group.

- DeploymentDataSet GetDeploymentInstructions ()

  Return a DataSet of all the deployment instructions

- DeploymentDataSet GetDeploymentInstructionsFromDiscoveredMachineKey (Guid discoveredMachineKey)

  Returns a data set consisting of all instructions for a discovered machine

- DeploymentDataSet GetDeploymentInstructionsFromGroupKey (int commandID)

  Returns a data set consisting of all instructions for a command ID.

- DeploymentDataSet GetStatusHistory (Guid instructionKey)

  Get status history

Detailed Description

Manages deployment of the CCA. When the "Install CCA" instruction is selected from the Management Console each client machine has an associated DeploymentInstruction. This is then used to provide the "CCA Install Log" to the ManagementConsole and any deployment status. The "DeploymentCredentials" is a store of the usernames and passwords provided in the ManagementConsole. The passwords are RSA encrypted.

Member Function Documentation

string DataAccessServices.WebServices.Deployment.ActivateDeploymentService (int commandID) `[inline]`

Activate Deployment Service.

Parameters:

commandID	Command for the deployment service.

Here we will signal either the global PollNow event, the global DeployCcaNow event or the global UploadEvents event. If the events are not present or the events don't have the required privileges then nothing happens

Returns:

Messages from the deployment service.

Command ID is 128 for deployment, 129 for poll now, or 130 for upload events.

void DataAccessServices.WebServices.Deployment.ApplyChanges (ref DeploymentDataSet deploymentChanges) `[inline]`

Updates the database with the changes in the Credentials table.

Parameters:

deploymentChanges	Data set containing changes to be applied.

void DataAccessServices.WebServices.Deployment.ApplyInstructionsChanges (ref DeploymentDataSet* deploymentChanges*, Guid?* groupKey*) `[inline]`

Updates the database with the changes in the Instructions table.

Parameters:

deploymentChanges	Data set containing changes to the instructions table.
groupKey	The group this instruction is associated with.

void DataAccessServices.WebServices.Deployment.ClearStatusHistory (Guid* instructionKey, *Guid?* groupKey*) `[inline]`

Clear status history.

Parameters:

instructionKey	Key for the deployment instruction.
groupKey	The group this instruction is associated with.

void DataAccessServices.WebServices.Deployment.CreateCredentials (Guid* credentialsKey, *Guid? groupKey, string* userName*, string* password*, int* index*, out DateTime* modifiedTime*) `[inline]`

Creates new credentials. To encrypt the password use DatabaseWebService.GetTransportPublicKey to retrieve the public key and encrypt using RSACryptoServiceProvider.Encrypt.

Parameters:

credentialsKey	A unique key for the credentials entry.
groupKey	The group this credentials entry is associated with.
userName	Username for the deployment credentials.
password	Password for the deployment credentials.

index	Order in which access with this credential will be attempted.
modifiedTime	OUT parameter will contain the latest modified time of the credentials o n return. Passed in value unused.

void DataAccessServices.WebServices.Deployment.CreateInstructions (Guid* instructionKey*, int* commandID*, Guid?* groupKey*, Guid?* machineKey*, string* status*, out DateTime* modifiedTime*) ***`[inline]`***

Creates a new deployment instruction.

Parameters:

instructionKey	A unique key for this deployment instruction.
commandID	Command for this instruction.
groupKey	Key identifying which group this instruction is associated with.
machineKey	Key identifying which machine this instruction will be run against.
status	Initial status of this instruction.
modifiedTime	OUT parameter will contain the latest modified time of the instruction on return. Passed in value unused.

Command ID is 128 for deployment and 129 for poll now.

DeploymentDataSet DataAccessServices.WebServices.Deployment.GetDeploymentCredentials () ***`[inline]`***

Returns a data set containing all deployment credentials.

Returns:

Data set containing all deployment credentials.

DeploymentDataSet DataAccessServices.WebServices.Deployment.GetDeploymentCredentialsFromGroupKey (Guid? groupKey) `[inline]`

Returns a data set consisting of all credentials for a group.

Parameters:

groupKey	Key specifying the deployment group to get credentials for.

Returns:

Data set consisting of all credentials for a group.

DeploymentDataSet DataAccessServices.WebServices.Deployment.GetDeploymentInstructions () `[inline]`

Return a DataSet of all the deployment instructions.

Returns:

The DataSet

DeploymentDataSet DataAccessServices.WebServices.Deployment.GetDeploymentInstructionsFromDiscoveredMachineKey (Guid discoveredMachineKey) `[inline]`

Returns a data set consisting of all instructions for a discovered machine.

Parameters:

discoveredMachineKey	Key identifying the machine these instructions are associated with.

Returns:

Data set consisting of all instructions for a discovered machine.

DeploymentDataSet DataAccessServices.WebServices.Deployment.GetDeploymentInstructionsFromGroupKey (int* **commandID*) [inline]

Returns a data set consisting of all instructions for a command ID.

Parameters:

commandID	Command for this instruction.

Returns:

Data set consisting of all instructions for a command ID.

Command ID is 128 for deployment and 129 for poll now.

DeploymentDataSet DataAccessServices.WebServices.Deployment.GetStatusHistory (Guid* **instructionKey*) [inline]

Get status history

Parameters:

instructionKey	Key for the deployment instruction.

Returns:

Data set containing status history for the specified instruction.

The documentation for this class was generated from the following file:

- Deployment.cs

DataAccessServices.WebServices.DiscoveredMachines Class Reference

Manages discovered machines in the Management Centre. When discovering computer membership within a group the conditions are evaluated for each group. If a machine from the active directory search matches the condition it is then added to the DiscoveredMachines table.

Public Member Functions

- void ApplyMachineChanges (ref DiscoveredMachinesDataSet machineChanges)

  Updates the database with all changes within the supplied data set.

- void CheckExpectedGroupPermissions (Guid[] discoveredMachineIds, String userSid, [Parameter(StructuredTypeName="dbo.StringList2")] DataTable groupSids, ObjectPermissions permissions)

  Checks permissions on the expected group for the given user

- DiscoveredMachinesDataSet CreateMachine (Guid discoveredMachinePK, Guid expectedGroupFK, String netBiosName, Guid? adObjectGuid, String description, String operatingSystem, Guid? actualGroupFK, String actualGroupName, String dns, Boolean userSpecified, out DateTime modifiedTime)

  Creates a new discovered machine entry.

- void DeleteAllMachines ()

  Deletes all discovered machines.

- void DeleteMachine (Guid discoveredMachinePK)

  Deletes the specified machine.

- DiscoveredMachinesDataSet FindMachines (String match)

  Returns data set with details of all discovered machines.

- SchedulerConfigurationDto GetDiscoveryServiceSettings ()

  Returns a SchedulerConfigurationDto object containing information about the Discovery Service.

- DiscoveredMachinesDataSet GetMachine (Guid? adObjectGuid, string dns, string netbiosName)

  Returns data set with details of the specified discovered machine.

- DiscoveredMachinesDataSet GetMachineFromDnsAndNetbiosServer (String dns, String netbios)

  Returns data set with details of the specified discovered machine.

- DiscoveredMachinesDataSet GetMachineFromKey (Guid machineKey)

  Returns data set with details of the specified discovered machine

- DiscoveredMachinesDataSet GetMachineFromObjectGuidServer (Guid adObjectGuid)

  Returns data set with details of the specified discovered machine.

- DiscoveredMachinesDataSet GetMachines ()

  Returns a data set with all discovered machines

- DiscoveredMachinesDataSet GetMachinesDelta (Guid consoleId)

  Returns a data set with all discovered machines that have changed since the last call to GetMachines.

- DiscoveredMachinesDataSet GetMachinesFromGroupKey (Guid groupKey)

  Returns a data set with all discovered machines in the specified group.

- DiscoveredMachinesDataSet GetMachinesFromGroupKeyDelta (Guid consoleId, Guid groupKey)

  Returns a data set with all changed discovered machines in the specified group and deployment states since the last call to GetMachinesWithSummary.

- DiscoveredMachinesDataSet GetMachinesFromGroupKeyWithSummary (Guid GroupKey, Boolean withSummary)

  Returns a data set with all discovered machines in the specified group optionally including diagnostics and deployment states.

- DiscoveredMachinesDataSet GetMachinesFromGroupKeyWithSummaryDelta (Guid consoleId, Guid groupKey, Boolean withSummary)

  Returns a data set with all changed discovered machines in the specified group optionally including diagnostics and deployment states.

- DiscoveredMachinesDataSet GetMachinesWithSummary (Boolean withSummary)

  Returns a data set with all discovered machines optionally including diagnostics and deployment states.

- DiscoveredMachinesDataSet GetMachinesWithSummaryDelta (Guid consoleId, Boolean withSummary)

  Returns a data set with all changed discovered machines optionally including diagnostics and deployment states since the last call to GetMachinesWithSummary.

- void GetMisgroupedDataFromGroupKey (Guid? groupKey, ref Int32 misgroupedCount, ref Int32 userSpecifiedCount)

  Returns a data set of all misgrouped discovered machines for the specified group.

- MisgroupedMachinesDataSet GetMisgroupedMachineCounts ()

  Returns a data set with the count of all misgrouped discovered machines.

- DiscoveredMachinesDataSet GetMisgroupedMachinesFromGroupKey (Guid? groupKey)

  Returns a data set with all misgrouped discovered machines in the specified group.

- DiscoveredMachinesDataSet GetPreload (int topRows)

  Returns the first set of DiscoveredMachines.

- DiscoveredMachinesDataSet GetPreloadFromGroupKey (Guid groupKey, int topRows)

  Returns the first set of DiscoveredMachines within a particular group.

- DiscoveredMachinesDataSet GetUserSpecifiedMachinesFromGroupKey (Guid? groupKey)

  Returns a data set with all user-specified discovered machines in the specified group.

- void InvokeDiscovery ()

  Invokes the discovery service.

- Boolean IsDiscoveryActive ()

  Returns a boolean indicating if the discovery is currently taking place.

- void Move (Guid destinationGroupKey, Guid[] discoveredMachineIds)

  Move a list of discovered machines from one group to another.

- DiscoveredMachinesDataSet SetExpectedGroup (Guid machineKey, Guid expectedGroup)

  Set the expected group for the machine, i.e. the group that the machine should be in

- void UpdateDiscoveryInterval (Int32 interval)

  Defines the period to trigger a discovery.

- void UpdateDiscoveryMode (Boolean auto)

  Returns a SchedulerConfigurationDto object containing information about the Discovery Service.

- void UpdateMachineDiscovery (Guid discoveredMachineKey, Guid expectedGroupKey, String netBiosName, String description, String operatingSystem, String dns, ref DateTime modifiedDiscoveryTime)

  Updates the specified discovered machine.

Detailed Description

Manages discovered machines in the Management Centre. When discovering computer membership within a group the conditions are evaluated for each group. If a machine from the active directory search matches the condition it is then added to the DiscoveredMachines table.

Member Function Documentation

void DataAccessServices.WebServices.DiscoveredMachines.ApplyMachineChanges (ref DiscoveredMachinesDataSet* machineChanges*) [inline]

Updates the database with all changes within the supplied data set.

Parameters:

machineChanges	The data set consisting of the changes.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.DiscoveredMachines.CheckExpectedGroupPermissions (Guid []* discoveredMachineIds*, String* userSid*, [Parameter(StructuredTypeName = "dbo.StringList2")] DataTable* groupSids*, ObjectPermissions* permissions*)* `[inline]`**

Checks permissions on the expected group for the given user.

Parameters:

discoveredMachineIds	Array of discovered machine identifiers.
userSid	The sid of the user.
groupSids	The list of group sids.
permissions	The list of permissions to check.

***DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.CreateMachine (Guid* discoveredMachinePK*, Guid* expectedGroupFK*, String* netBiosName*, Guid?* adObjectGuid*, String* description*, String* operatingSystem*, Guid?* actualGroupFK*, String* actualGroupName*, String* dns*, Boolean* userSpecified*, out DateTime* modifiedTime*)* `[inline]`**

Creates a new discovered machine entry.

Parameters:

discoveredMachinePK	A unique key for this discovered machine.
expectedGroupFK	Key identifying the group this machine is expected to be associated with.
netBiosName	The NetBIOS name of the machine.
adObjectGuid	GUID identifier for the machine in Active Directory.
description	Description of this machine.
operatingSystem	Operating system of the machine.
actualGroupFK	Key identifying the group the machine is actually associated with.

actualGroupName	Name of the group the machine is actually associated with.
dns	Fully qualified DNS entry for the machine.
userSpecified	TRUE if specified by the user. FALSE if discovered via membership rules.
modifiedTime	OUT parameter will contain the latest modified time of the machine on return. Passed in value unused.

Returns:

Returns data set containing details of the created machine.

void DataAccessServices.WebServices.DiscoveredMachines.DeleteAllMachines () `[inline]`

Deletes all discovered machines.

void DataAccessServices.WebServices.DiscoveredMachines.DeleteMachine (Guid* **discoveredMachinePK*) `[inline]`

Deletes the specified machine.

Parameters:

discoveredMachinePK	Key for the discovered machine.

DiscoveredMachinesDataSet
DataAccessServices.WebServices.DiscoveredMachines.FindMachines (String **match*)*** `[inline]`

Returns data set with details of all discovered machines.

Parameters:

match	Not currently used.

Returns:

Data set with details of all discovered machines.

SchedulerConfigurationDto DataAccessServices.WebServices.DiscoveredMachines.GetDiscoveryServiceSettings () `[inline]`

Returns a SchedulerConfigurationDto object containing information about the Discovery Service.

Returns:

A SchedulerConfigurationDto object containing details about the Discovery Service.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachine (Guid?* adObjectGuid*, string* dns*, string* netbiosName*) `[inline]`

Returns data set with details of the specified discovered machine.

Parameters:

adObjectGuid	GUID identifier for the machine in Active Directory.
dns	Fully qualified DNS entry for the machine.
netbiosName	The NetBIOS name of the machine.

Returns:

Data set with details of the specified discovered machine.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachineFromDnsAndNetbiosServer (String* dns*, String* netbios*) `[inline]`

Returns data set with details of the specified discovered machine.

Parameters:

dns	The DNS name of the discoveredMachine.
netbios	The NETBIOS name of the discoveredMachine.

Returns:

Data set with details of the specified discovered machine.

***DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachineFromKey (Guid* machineKey*)* `[inline]`**

Returns data set with details of the specified discovered machine.

Parameters:

machineKey	Key for the discovered machine.

Returns:

Data set with details of the specified discovered machine.

***DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachineFromObjectGuidServer (Guid* adObjectGuid*)* `[inline]`**

Returns data set with details of the specified discovered machine.

Parameters:

adObjectGuid	GUID identifier for the machine in Active Directory.

Returns:

Data set with details of the specified discovered machine.

***DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachines ()* `[inline]`**

Returns a data set with all discovered machines.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachinesDelta (Guid* consoleId*) `[inline]`

Returns a data set with all discovered machines that have changed since the last call to GetMachines.

Parameters:

consoleId	Id of the invoking console used to obtain the last refresh date referenced in MachineRefresh table.

Returns:

Data set with all discovered machines that have changed.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachinesFromGroupKey (Guid* groupKey*) `[inline]`

Returns a data set with all discovered machines in the specified group.

Parameters:

groupKey	Key for the group associated with the discovered machines.

Returns:

Data set with all discovered machines in the specified group.

***DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachinesFromGroupKeyDelta (Guid* consoleId, *Guid* groupKey) `[inline]`**

Returns a data set with all changed discovered machines in the specified group and deployment states since the last call to GetMachinesWithSummary.

Parameters:

consoleId	Id of the invoking console used to obtain the last refresh date referenced in MachineRefresh table.
groupKey	Key for the group associated with the discovered machines.

Returns:

Data set with all changed discovered machines in the specified group.

***DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachinesFromGroupKeyWithSummary (Guid* GroupKey, *Boolean* withSummary) `[inline]`**

Returns a data set with all discovered machines in the specified group optionally including diagnostics and deployment states.

Parameters:

GroupKey	Key for the group associated with the discovered machines.
withSummary	Boolean flag indicating if the diagnostics and deployment state should be retrieved.

Returns:

Data set with all discovered machines in the specified group including diagnostics and deployment states.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachinesFromGroupKeyWithSummaryDelta (Guid* consoleId*, Guid* groupKey*, Boolean* withSummary*) [inline]

Returns a data set with all changed discovered machines in the specified group optionally including diagnostics and deployment states.

Parameters:

consoleId	Id of the invoking console used to obtain the last refresh date referenced in MachineRefresh table.
groupKey	Key for the group associated with the discovered machines.
withSummary	Boolean flag indicating if the diagnostics and deployment state should be retrieved.

Returns:

Data set with all changed discovered machines in the specified group including diagnostics and deployment states.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachinesWithSummary (Boolean* withSummary*) [inline]

Returns a data set with all discovered machines optionally including diagnostics and deployment states.

Parameters:

withSummary	Boolean flag indicating if the diagnostics and deployment state should be retrieved.

Returns:

Data set with all discovered machines including diagnostics and deployment states.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMachinesWithSummaryDelta (Guid* consoleId*, Boolean* withSummary*) [inline]

Returns a data set with all changed discovered machines optionally including diagnostics and deployment states since the last call to GetMachinesWithSummary.

Parameters:

consoleId	Id of the invoking console used to obtain the last refresh date referenced in MachineRefresh table.
withSummary	Boolean flag indicating if the diagnostics and deployment state should be retrieved.

Returns:

Data set with all changed discovered machines including diagnostics and deployment states.

void DataAccessServices.WebServices.DiscoveredMachines.GetMisgroupedDataFromGroupKey (Guid?* groupKey*, ref Int32* misgroupedCount*, ref Int32* userSpecifiedCount*) [inline]

Returns a data set of all misgrouped discovered machines for the specified group.

Parameters:

groupKey	Key for the group associated with the discovered machines.
misgroupedCount	The count of misgrouped machines discovered via membership rules.
userSpecifiedCount	The count of user specified machines.

MisgroupedMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMisgroupedMachineCounts () [inline]

Returns a data set with the count of all misgrouped discovered machines.

Returns:

Data set with the count of all misgrouped discovered machines.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetMisgroupedMachinesFromGroupKey (Guid? groupKey) `[inline]`

Returns a data set with all misgrouped discovered machines in the specified group.

Parameters:

groupKey	Key for the group associated with the discovered machines.

Returns:

Data set with all misgrouped discovered machines in the specified group.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetPreload (int topRows) `[inline]`

Returns the first set of DiscoveredMachines.

Parameters:

topRows	The number of rows to return.

Returns:

Data set containing the first set of DiscoveredMachines as specified by the topRows value.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetPreloadFromGroupKey (Guid groupKey, int topRows) `[inline]`

Returns the first set of DiscoveredMachines within a particular group.

Parameters:

groupKey	Id of the group.
topRows	The number of rows to return.

Returns:

Data set containing the first set of DiscoveredMachines in the given group as specified by the topRows value.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.GetUserSpecifiedMachinesFromGroupKey (Guid? groupKey) [inline]

Returns a data set with all user-specified discovered machines in the specified group.

Parameters:

groupKey	Key for the group associated with the discovered machines.

Returns:

Data set with all user-specified discovered machines in the specified group.

void DataAccessServices.WebServices.DiscoveredMachines.InvokeDiscovery () [inline]

Invokes the discovery service.

Boolean DataAccessServices.WebServices.DiscoveredMachines.IsDiscoveryActive () [inline]

Returns a boolean indicating if the discovery is currently taking place.

Returns:

Boolean indicating if the discovery is taking place.

void DataAccessServices.WebServices.DiscoveredMachines.Move (Guid destinationGroupKey, Guid [] discoveredMachineIds) [inline]

Move a list of discovered machines from one group to another.

Parameters:

destinationGroupKey	Move a list of discovered machines from one group to another.
discoveredMachineIds	List of ids for the machines to move.

DiscoveredMachinesDataSet DataAccessServices.WebServices.DiscoveredMachines.SetExpectedGroup (Guid* machineKey, *Guid* expectedGroup*) `[inline]`

Set the expected group for the machine, i.e. the group that the machine should be in.

Parameters:

machineKey	The key for the machine.
expectedGroup	The key for the expected group.

Returns:

The dataset containing the machine record.

void DataAccessServices.WebServices.DiscoveredMachines.UpdateDiscoveryInterval (Int32 interval*)*** `[inline]`

Defines the period to trigger a discovery.

Parameters:

interval	The number of milliseconds between each automatic discovery.

This value is only used when the discovery mode is set to automatic.

void DataAccessServices.WebServices.DiscoveredMachines.UpdateDiscoveryMode (Boolean auto*)*** `[inline]`

Returns a SchedulerConfigurationDto object containing information about the Discovery Service.

Parameters:

auto	True for Automatic discovery. False for Manual discovery.

The method takes one parameter: setting the parameter to true sets the mode to automatic, whereas setting the value to false sets the discovery mode to manual. Automatic discovery triggers a discovery every period set by the call to UpdateDiscoveryInterval. Manual discovery only triggers on a call to InvokeDiscovery.

***void DataAccessServices.WebServices.DiscoveredMachines.UpdateMachineDiscovery (Guid* discoveredMachineKey, *Guid* expectedGroupKey, *String* netBiosName, *String* description, *String* operatingSystem, *String* dns, *ref DateTime* modifiedDiscoveryTime) `[inline]`**

Updates the specified discovered machine.

Parameters:

discoveredMachineKey	Key for the discovered machine.
expectedGroupKey	Key identifying the group that this machine is expected to be associated with.
netBiosName	The NetBIOS name of the machine.
description	Description of this machine.
operatingSystem	Operating system of the machine.
dns	Fully qualified DNS entry for the machine.
modifiedDiscoveryTime	Time this entry was last modified.

The documentation for this class was generated from the following file:

- DiscoveredMachines.cs

DataAccessServices.WebServices.Events Class Reference

Manages events in the Management Centre. Within the AppSense Management Suite the product agents can raise a number of different events to the Management Center. In order to receive and display these events the Management Center database contains a list of all the possible events that can be raised via the product agents.

Public Member Functions

- void AddEventDateTimeParam (long eventKey, string paramName, DateTime paramValue)

  Adds a date/time parameter to an existing event.

- void AddEventDefinitionParam (int eventDefinitionKey, string name, string description, EventParamType type, out DateTime modifiedTime)

  Adds a parameter to an existing event definition.

- void AddEventIntegerParam (long eventKey, string paramName, long paramValue)

  Adds an integer parameter to an existing event.

- void AddEventStringParam (long eventKey, string paramName, string paramValue)

  Adds a string parameter to an existing event.

- void ApplyEventChanges (EventsDataSet eventsChanges)

  Deletes all events that have been deleted within the supplied data set.

- void ApplyEventDefinitionChanges (ref EventDefinitionsDataSet eventDefinitionChanges)

  Applies changes within an event definition data set into the database.

- long CreateEvent (int eventDefinitionKey, Guid? machineKey, Guid? groupKey, string userName, DateTime time)

  Creates a new event.

- void CreateEventDefinition (int eventDefinitionKey, string name, string description, int type, string eventDescription, Guid productKey, bool highVolume, bool defaultEnabledState, string version, out DateTime modifiedTime)

  Creates a new event definition. Supplied key must not exist.

- void DeleteEvent (long eventKey)

  Deletes an existing event.

- void DeleteEventDefinition (int eventDefinitionKey, DateTime? modifiedTime)

  Deletes an existing event definition.

- void DeleteEventsFromAlertKey (int alertKey)

  Deletes events associated with a specific alert.

- void DeleteEventsFromGroupKey (Guid? groupKey, bool unreferencedOnly)

  Deletes events generated by the specified group.

- void DeleteEventsFromMachineKey (Guid? machineKey, bool unreferencedOnly)

  Deletes events generated by the specified machine.

- EventDefinitionsDataSet GetEventDefinitions ()

  Returns all event definitions.

- EventDefinitionsDataSet GetEventDefinitionsFromKey (int eventDefinitionKey)

  Returns a specific event definition.

- EventsDataSet GetEventFromKey (long eventKey, bool withParameters)

  Returns a single event.

- EventParameterValuesDataSet GetEventParameterValues (int eventNumber, string eventParameter)

  Gets the distinct values for an event parameter

- EventsDataSet GetEventsForMultipleIds (List< int > eventTypeIds, Guid? groupKey, string userName, string machineName, bool withParameters, DateTime? startDate, DateTime? endDate)

  Return the first 10K Events based on a list of IDs. It can be further filtered by a group, user, machine and date range

- EventsDataSet GetEventsFromAlert (int alertKey, bool withParameters)

  Returns all events associated with the specific alert.

- EventsDataSet GetEventsFromGroupKey (Guid? groupKey, bool withParameters)

  Returns all events associated with a specific group.

- EventsDataSet GetEventsFromMachineKey (Guid? machineKey, bool withParameters)

  Returns all events associated with a specific machine.

- EventsDataSet GetEventsFromQuery (string queryString)

  Executes the supplied expression returning any results.

- EventsDataSet GetEventsFromRange (long firstEventKey, long maxResults, bool withParameters)

  Returns upto maxResults events, starting with firstEventKey.

- DataSet GetExpandedEventsFromQuery (string queryString)

  Executes the supplied expression returning any events in an expanded data set.

- DataTable GetSummaryByParameterValues (List< int > eventTypeIds, List< string > parameterNames, string additionalParameterName, Guid? groupKey, DateTime? startDate, DateTime? endDate)

  Return grouped summary of events by event id and parameter names, grouped by values in the parameter name

- void RemoveEventDefinitionParam (int eventDefinitionKey, string name, DateTime? modifiedTime)

  Removes an existing parameter from an event definition.

- void UpdateEventDefinition (int eventDefinitionKey, string name, string description, int type, string eventDescription, Guid productKey, bool highVolume, bool defaultEnabledState, string version, ref DateTime modifiedTime)

  Updates an existing event definition.

Detailed Description

Manages events in the Management Centre. Within the AppSense Management Suite the product agents can raise a number of different events to the Management Center. In order to receive and display these events the Management Center database contains a list of all the possible events that can be raised via the product agents.

Each product agent in the AppSense suite can raise an event based on different conditions within the product agent.The Management Center must store all the different event definitions for each product in order to successfully report on any event that is configured and reported back via the CCA.

Member Function Documentation

void DataAccessServices.WebServices.Events.AddEventDateTimeParam (long* eventKey*, string* paramName*, DateTime* paramValue*) `[inline]`

Adds a date/time parameter to an existing event.

Parameters:

eventKey	The key of the event to add the parameter to.
paramName	The name of the parameter.
paramValue	The date/time value of the parameter.

Requires alert or administrative access.

void DataAccessServices.WebServices.Events.AddEventDefinitionParam (int* eventDefinitionKey*, string* name*, string* description*, EventParamType* type*, out DateTime* modifiedTime*) `[inline]`

Adds a parameter to an existing event definition.

Parameters:

eventDefinitionKey	The key of the event definition to add the parameter onto.
name	The name of the parameter.

description	A description of the parameter.
type	The data type of the parameter values.
modifiedTime	OUT parameter will contain the latest modified time of the event definition on return. Passed in value unused.

Requires administrative access.

void DataAccessServices.WebServices.Events.AddEventIntegerParam (long* eventKey*, string* paramName*, long* paramValue*) `[inline]`

Adds an integer parameter to an existing event.

Parameters:

eventKey	The key of the event to add the parameter to.
paramName	The name of the parameter.
paramValue	The integer value of the parameter.

Requires alert or administrative access.

void DataAccessServices.WebServices.Events.AddEventStringParam (long* eventKey*, string* paramName*, string* paramValue*) `[inline]`

Adds a string parameter to an existing event.

Parameters:

eventKey	The key of the event to add the parameter to.
paramName	The name of the parameter.
paramValue	The string value of the parameter.

Requires alert or administrative access.

***void DataAccessServices.WebServices.Events.ApplyEventChanges (EventsDataSet* eventsChanges*)* `[inline]`**

Deletes all events that have been deleted within the supplied data set.

Parameters:

eventsChanges	The data set consisting of the changes.

***void DataAccessServices.WebServices.Events.ApplyEventDefinitionChanges (ref EventDefinitionsDataSet* eventDefinitionChanges*)* `[inline]`**

Applies changes within an event definition data set into the database.

Parameters:

eventDefinitionChanges	A data set consisting of modifications.

Requires administrative access.

***long DataAccessServices.WebServices.Events.CreateEvent (int* eventDefinitionKey*, Guid?* machineKey*, Guid?* groupKey*, string* userName*, DateTime* time*)* `[inline]`**

Creates a new event.

Parameters:

eventDefinitionKey	The event definition of the event to create.
machineKey	The key of the machine which generated the event.
groupKey	The key of the group containing the machine when the events were generated.
userName	The name of the user who generated the event.
time	The time that the event was generated.

Returns:

The event key which identifies the created event.

Requires alert or administrative access.

***void DataAccessServices.WebServices.Events.CreateEventDefinition (int* eventDefinitionKey*, string* name*, string* description*, int* type*, string* eventDescription*, Guid* productKey*, bool* highVolume*, bool* defaultEnabledState*, string* version*, out DateTime* modifiedTime*)* `[inline]`**

Creates a new event definition. Supplied key must not exist.

Parameters:

eventDefinitionKey	The key which identifies this event definition.
name	The name of the event definition.
description	A description of the event definition.
type	The type of the event. 4 for information, 2 for warning, 1 for error.
eventDescription	A format string for displaying event descriptions.
productKey	The key of the product associated with this event.
highVolume	True if large quantities of these events are generated.
defaultEnabledState	The default state of this event definition.
version	The version of the event definition.
modifiedTime	OUT parameter will contain the latest modified time of the event definition on return. Passed in value unused.

Requires administrative access.

***void DataAccessServices.WebServices.Events.DeleteEvent (long* eventKey*)* `[inline]`**

Deletes an existing event.

Parameters:

eventKey	The key of the event to delete.

Requires alert or administrative access.

***void DataAccessServices.WebServices.Events.DeleteEventDefinition (int* eventDefinitionKey, *DateTime?* modifiedTime) `[inline]`**

Deletes an existing event definition.

Parameters:

eventDefinitionKey	The key of the event definition.
modifiedTime	The time that the event definition was modified.

Requires administrative access.

***void DataAccessServices.WebServices.Events.DeleteEventsFromAlertKey (int* alertKey) `[inline]`**

Deletes events associated with a specific alert.

Parameters:

alertKey	The alert key whose events will be deleted.

***void DataAccessServices.WebServices.Events.DeleteEventsFromGroupKey (Guid?* groupKey, *bool* unreferencedOnly) `[inline]`**

Deletes events generated by the specified group.

Parameters:

groupKey	The group that contains the events to delete.
unreferencedOnly	Whether to just delete events that are not referenced by an alert.

***void DataAccessServices.WebServices.Events.DeleteEventsFromMachineKey (Guid?* machineKey, *bool* unreferencedOnly*)* `[inline]`**

Deletes events generated by the specified machine.

Parameters:

machineKey	The machine that contains the events to delete.
unreferencedOnly	Whether to just delete events that are not referenced by an alert.

EventDefinitionsDataSet DataAccessServices.WebServices.Events.GetEventDefinitions () `[inline]`

Returns all event definitions.

Returns:

A data set consisting of all event definitions and their parameters.

Requires alert or administrative access.

EventDefinitionsDataSet DataAccessServices.WebServices.Events.GetEventDefinitionsFromKey (int* eventDefinitionKey*) `[inline]`

Returns a specific event definition.

Parameters:

eventDefinitionKey	The key of the event definition to return.

Returns:

A data set consisting of the event definition and its parameters.

Requires alert or administrative access.

EventsDataSet DataAccessServices.WebServices.Events.GetEventFromKey (long* eventKey*, bool* withParameters*) `[inline]`

Returns a single event.

Parameters:

eventKey	The event to return.
withParameters	Whether to include event parameters.

Returns:

A data set consisting of the event and its parameters.

Requires alert, deployment or administrative access.

EventParameterValuesDataSet DataAccessServices.WebServices.Events.GetEventParameterValues (int* eventNumber*, string* eventParameter*) `[inline]`

Gets the distinct values for an event parameter

Parameters:

eventNumber	The event number.
eventParameter	The event parameter.

Returns:

Data set containing event parameter values.

EventsDataSet DataAccessServices.WebServices.Events.GetEventsForMultipleIds (List< int >* eventTypeIds*, Guid?* groupKey*, string* userName*, string* machineName*, bool* withParameters*, DateTime?* startDate*, DateTime?* endDate*) `[inline]`

Return the first 10K Events based on a list of IDs. It can be further filtered by a group, user, machine and date range.

Parameters:

eventTypeIds	List of event type ids.
groupKey	specify a group by its key, or null for all groups.
userName	Specify a user by name, or null for all users.
machineName	Specify a machine by name, or null for all machines.
withParameters	Set to true if you want the parameters for the events returned.
startDate	Start date for event filtering (optional).
endDate	End date for event filtering (optional).

Returns:

Data Table with appropriate columns based on parameter names.

EventsDataSet DataAccessServices.WebServices.Events.GetEventsFromAlert (int alertKey*, bool* withParameters*)*** `[inline]`

Returns all events associated with the specific alert.

Parameters:

alertKey	The key of the alert.
withParameters	Whether event parameters should be returned.

Returns:

Requires alert, deployment or administrative access.

EventsDataSet DataAccessServices.WebServices.Events.GetEventsFromGroupKey (Guid?* groupKey*, bool* withParameters*) `[inline]`

Returns all events associated with a specific group.

Parameters:

groupKey	The group key of the events to return.
withParameters	Whether to include event parameters.

Returns:

A data set consisting of all events and parameters within a specific group.

Requires alert, deployment or administrative access.

EventsDataSet DataAccessServices.WebServices.Events.GetEventsFromMachineKey (Guid?* machineKey*, bool* withParameters*) `[inline]`

Returns all events associated with a specific machine.

Parameters:

machineKey	The machine key of the events to return.
withParameters	Whether to include event parameters.

Returns:

A data set consisting of all events and parameters associated with a machine.

Requires alert, deployment or administrative access.

***EventsDataSet DataAccessServices.WebServices.Events.GetEventsFromQuery (string* queryString*)* `[inline]`**

Executes the supplied expression returning any results.

Parameters:

queryString	The query string.

Returns:

Data set containing events.

***EventsDataSet DataAccessServices.WebServices.Events.GetEventsFromRange (long* firstEventKey*, long* maxResults*, bool* withParameters*)* `[inline]`**

Returns upto maxResults events, starting with firstEventKey.

Parameters:

firstEventKey	The earliest value of the key to return.
maxResults	The maximum number of events to return.
withParameters	Whether to include event parameters.

Returns:

A data set consisting of events within the supplied range.

Requires alert or administrative access.

***DataSet DataAccessServices.WebServices.Events.GetExpandedEventsFromQuery (string* queryString*)* `[inline]`**

Executes the supplied expression returning any events in an expanded data set.

Parameters:

queryString	The query string.

Returns:

Events in an expanded data set.

DataTable DataAccessServices.WebServices.Events.GetSummaryByParameterValues (List< int >* eventTypeIds*, List< string >* parameterNames*, string* additionalParameterName*, Guid?* groupKey*, DateTime?* startDate*, DateTime?* endDate*) [inline]

Return grouped summary of events by event id and parameter names, grouped by values in the parameter name.

Parameters:

eventTypeIds	List of event type ids.
parameterNames	List of names of event parameters (only string-valued parameters supported).
additionalParameterName	Additional parameter displayed for each row of grouping, representing a random value in that parameter for the group. This parameter should have the same value for all events in the group. (Optional).
groupKey	Guid of deployment group for filtering events (Optional).
startDate	Start date for event filtering (optional).
endDate	End date for event filterig (optional).

Returns:

Data Table with appropriate columns based on parameter names.

void DataAccessServices.WebServices.Events.RemoveEventDefinitionParam (int* eventDefinitionKey*, string* name*, DateTime?* modifiedTime*) [inline]

Removes an existing parameter from an event definition.

Parameters:

eventDefinitionKey	The key of the event definition to remove the event from.
name	The name of the event definition.
modifiedTime	The time that the event definition was modified.

Requires administrative access.

***void DataAccessServices.WebServices.Events.UpdateEventDefinition (int* eventDefinitionKey, *string* name, *string* description, *int* type, *string* eventDescription, *Guid* productKey, *bool* highVolume, *bool* defaultEnabledState, *string* version, *ref DateTime* modifiedTime) `[inline]`**

Updates an existing event definition.

Parameters:

eventDefinitionKey	The key which identifies this event definition.
name	The name of the event definition.
description	A description of the event definition.
type	The type of the event. 4 for information, 2 for warning, 1 for error.
eventDescription	A format string for displaying event descriptions.
productKey	The key of the product associated with this event.
highVolume	True if large quantities of these events are generated.
defaultEnabledState	The default state of this event definition.
version	The product version of the event definition.
modifiedTime	The time that the event definition was modified.

Requires administrative access.

The documentation for this class was generated from the following file:

- Events.cs

DataAccessServices.WebServices.Groups Class Reference

Manages deployment groups and assigned packages in the Management Centre. The groups table represents a deployment group, with associated settings being stored within the related tables. The GroupPackages and MachinePackages tables represent the packages which are assigned to a group, and are installed on the group's machines.

Public Member Functions

- void AddGroupLatestPackage (Guid groupKey, Guid packageKey, out DateTime creationTime, out DateTime modifiedTime)

  Adds a package to a group using the latest version.

- void AddGroupPackage (Guid groupKey, Guid packageKey, Guid? patchKey, int? major, int? minor, int? build, int? revision, Guid? certificateKey, out DateTime creationTime, out DateTime modifiedTime)

  Adds a package to a group.

- void AddGroupPatch (Guid groupKey, Guid packageKey, Guid patchKey, out DateTime creationTime, out DateTime modifiedTime)

  Adds a patch to a group.

- void ApplyEventFilterChanges (ref EventFilterDataSet eventFilterChanges)

  Applies changes made to the event filter data set into the database.

- void ApplyGroupChanges (ref GroupsDataSet groupChanges)

  Applies changes found within the data set to the database.

- void ApplyGroupPackagesChanges (ref GroupPackagesDataSet groupPackagesChanges)

  Applies changes found within the group packages data set to the database.

- void ApplyGroupSecurityChanges (ref GroupsDataSet groupChanges)

  Applies changes found within the data set to the database.

- void ApplyInstallationScheduleChanges (ref ScheduleDataSet scheduleChanges)

  Applies the changes of an installation schedule to the database.

- Guid CloneGroup (Guid groupKey)

  Make a copy of a group, including the assigned packages, but not the machines.

- void CreateGroup (Guid groupKey, string name, string description, int pollPeriodSeconds, int uploadPollPeriodSeconds, byte eventLogEnabled, byte fileLogEnabled, string fileLogFilename, bool anonymousUserLogging, bool anonymousMachineLogging, bool overrideServerUrls, out DateTime modifiedTime, bool selfRegistrationEnabled, bool selfUnregistrationEnabled, bool selfUpdateEnabled, int priority, int pollPeriodVariationSeconds, int uploadPollPeriodVariationSeconds, bool nativeConfigurations, string configurationLocation)

  Constructs a group.

- void DeleteGroup (Guid groupKey, DateTime? modifiedTime)

  Deletes an existing group.
- GroupsDataSet GetDefault ()

  Returns a data set containing the default group in the groups table.
- GroupsDataSet GetDeploymentGroupsLight ()

  Returns a filtered list of group keys/names in the standard data set.
- DeploymentGroupDto [] GetDeploymentGroupsLightDto ()

  Returns an ordered list of all the group names.
- EventFilterDataSet GetEventFilter (Guid groupKey)

  Returns the event filter for a group.
- GroupsDataSet GetForCertificate (Guid packageKey, Guid certificateKey)

  Get the groups that have the certificate, defined by its package key and certificate key
- GroupsDataSet GetForPackageVersion (Guid packageKey, int major, int minor, int build, int revision)

  Get the groups that have the specified package version assigned
- GroupsDataSet GetForPatch (Guid packageKey, Guid patchKey)

  Get the groups that have the specified patch assigned.
- GroupsDataSet GetGroupFromKey (Guid groupKey, bool withSummary)

  Returns a single group based on a group key.
- GroupPackagesDataSet GetGroupPackages (Guid groupKey)

  Returns a list of packages assigned to a group.
- GroupsDataSet GetGroups (bool withSummary)

  Returns all groups known to the database.
- NamedValuesDataSet GetInfo (Guid? groupKey)

  Returns a data set consisting statistical information about the specified group.
- ScheduleDataSet GetInstallationSchedule (Guid groupKey)

  Returns an installation schedule for the supplied group.
- StatisticsDto GetStatistics ()

  Get a statistics summary of all groups, computers and alerts.
- void RemoveGroupPackage (Guid groupKey, Guid packageKey, DateTime? modifiedTime)

  Removes an existing package from a group.
- void RemoveGroupPatch (Guid groupKey, Guid patchKey, DateTime? modifiedTime)

  Removes an existing patch from a group.
- void UpdateEventFilter (Guid groupKey, int eventDefinitionKey, bool enabled)

Adds or removed an event from the event filter of a group, depending upon whether the event is enabled.

- void UpdateGroup (Guid groupKey, string name, string description, int pollPeriodSeconds, int uploadPollPeriodSeconds, byte eventLogEnabled, byte fileLogEnabled, string fileLogFilename, bool anonymousUserLogging, bool anonymousMachineLogging, bool overrideServerUrls, Guid? policyKey, string ownerSid, ref DateTime modifiedTime, bool selfRegistrationEnabled, bool selfUnregistrationEnabled, bool selfUpdateEnabled, int priority, int pollPeriodVariationSeconds, int uploadPollPeriodVariationSeconds, bool nativeConfigurations, string configurationLocation)

  Updates the properties of an existing group.

- void UpdateGroupLatestPackage (Guid groupKey, Guid packageKey, ref DateTime modifiedTime)

  Updates the version of a package within a group.

- void UpdateGroupPackage (Guid groupKey, Guid packageKey, Guid? patchKey, int major, int minor, int build, int revision, ref DateTime modifiedTime)

  Updates the version of a package within a group.

- void UpdateGroupSecurity (Guid groupKey, Guid? policyKey, string ownerSid, ref DateTime modifiedTime)

  Updates security options for a group.

- void UpdateInstallationSchedule (Guid groupKey, long mondayFlags, long tuesdayFlags, long wednesdayFlags, long thursdayFlags, long fridayFlags, long saturdayFlags, long sundayFlags, int flags, bool doNotInstallAgent, ref DateTime modifiedTime, long mondayConfigFlags, long tuesdayConfigFlags, long wednesdayConfigFlags, long thursdayConfigFlags, long fridayConfigFlags, long saturdayConfigFlags, long sundayConfigFlags, int configFlags, bool doNotInstallConfig, bool mirrorAgentSchedule, bool postponeInstallationPromptEnabled, int postponeInstallationLimit, bool downloadConfigurationsOnStartup, bool midSessionUpdate, bool updateAtShutdown)

  Updates the installation schedule for an individual group.

Detailed Description

Manages deployment groups and assigned packages in the Management Centre. The groups table represents a deployment group, with associated settings being stored within the related tables. The GroupPackages and MachinePackages tables represent the packages which are assigned to a group, and are installed on the group's machines.

Member Function Documentation

void DataAccessServices.WebServices.Groups.AddGroupLatestPackage (Guid* groupKey*, Guid* packageKey*, out DateTime* creationTime*, out DateTime* modifiedTime*) `[inline]`

Adds a package to a group using the latest version.

Parameters:

groupKey	The key of the group.
packageKey	The key of the package.
creationTime	Time the group was created.
modifiedTime	Time the group was last modified.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.AddGroupPackage (Guid* groupKey*, Guid* packageKey*, Guid?* patchKey*, int?* major*, int?* minor*, int?* build*, int?* revision*, Guid?* certificateKey*, out DateTime* creationTime*, out DateTime* modifiedTime*) `[inline]`

Adds a package to a group.

Parameters:

groupKey	The key of the group.
packageKey	The key of the package.
patchKey	The patch GUID.
major	Major version of package version to add.

minor	Minor version of package version to add.
build	Build of package version to add.
revision	Revision of package version to add.
certificateKey	The certificate key, or null.
creationTime	Time the group was created.
modifiedTime	Time the group was last modified.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.AddGroupPatch (Guid* groupKey*, Guid* packageKey*, Guid* patchKey*, out DateTime* creationTime*, out DateTime* modifiedTime*) ***`[inline]`***

Adds a patch to a group.

Parameters:

groupKey	The key of the group.
packageKey	The key of the package.
patchKey	The key of the patch.
creationTime	Time the group was created.
modifiedTime	Time the group was last modified.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.ApplyEventFilterChanges (ref EventFilterDataSet* eventFilterChanges*) ***`[inline]`***

Applies changes made to the event filter data set into the database.

Parameters:

eventFilterChanges	A modified data set.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.Groups.ApplyGroupChanges (ref GroupsDataSet* groupChanges*)* `[inline]`**

Applies changes found within the data set to the database.

Parameters:

groupChanges	A data set consisting of changes.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.Groups.ApplyGroupPackagesChanges (ref GroupPackagesDataSet* groupPackagesChanges*)* `[inline]`**

Applies changes found within the group packages data set to the database.

Parameters:

groupPackagesChanges	A data set consisting of changes.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.Groups.ApplyGroupSecurityChanges (ref GroupsDataSet* groupChanges*)* `[inline]`**

Applies changes found within the data set to the database.

Parameters:

groupChanges	A data set consisting of changes.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.Groups.ApplyInstallationScheduleChanges (ref ScheduleDataSet* scheduleChanges*)* `[inline]`**

Applies the changes of an installation schedule to the database.

Parameters:

scheduleChanges	The schedule changes.

Requires deployment or administrative access.

***Guid DataAccessServices.WebServices.Groups.CloneGroup (Guid* groupKey*)* `[inline]`**

Make a copy of a group, including the assigned packages, but not the machines.

Parameters:

groupKey	

Returns:

The guid of the new group.

***void DataAccessServices.WebServices.Groups.CreateGroup (Guid* groupKey*, string* name*, string* description*, int* pollPeriodSeconds*, int* uploadPollPeriodSeconds*, byte* eventLogEnabled*, byte* fileLogEnabled*, string* fileLogFilename*, bool* anonymousUserLogging*, bool* anonymousMachineLogging*, bool* overrideServerUrls*, out DateTime* modifiedTime*, bool* selfRegistrationEnabled*, bool* selfUnregistrationEnabled*, bool* selfUpdateEnabled*, int* priority*, int* pollPeriodVariationSeconds*, int* uploadPollPeriodVariationSeconds*, bool* nativeConfigurations*, string* configurationLocation*)* `[inline]`**

Constructs a group.

Parameters:

groupKey	The key which identifies this group.
name	The name of the group.
description	Description of the group.

pollPeriodSeconds	The amount of seconds between polls.
uploadPollPeriodSeconds	The amount of seconds between uploads.
eventLogEnabled	Whether to log events to the event log.
fileLogEnabled	Whether to log events to the file log.
fileLogFilename	The file name to log events to.
anonymousUserLogging	Whether events from machines within this group will be logged with anonymous users.
anonymousMachineLogging	Whether events from machines within this group will be logged with anonymous machines.
overrideServerUrls	Whether this group overrides server URL's.
modifiedTime	The time that the group was modified.
selfRegistrationEnabled	Whether or not self-registration is allowed.
selfUnregistrationEnabled	Whether or not self unregistration is allowed.
selfUpdateEnabled	Whether or not self-update of agents and configurations is allowed.
priority	Order in which this groups membership rules will be evaluated.
pollPeriodVariationSeconds	The VariationSeconds allowed for each poll.
uploadPollPeriodVariationSeconds	The VariationSeconds allowed for each upload.
nativeConfigurations	Whether configurations be deployed to this group in native format.
configurationLocation	Location on agent machine to store native configurations.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.DeleteGroup (Guid* groupKey, *DateTime?* modifiedTime*) [inline]

Deletes an existing group.

Parameters:

groupKey	The key which identifies the group.
modifiedTime	The time that the group was last modified.

Requires deployment or administrative access.

GroupsDataSet DataAccessServices.WebServices.Groups.GetDefault () [inline]

Returns a data set containing the default group in the groups table.

Returns:

Data set containing the default group in the groups table.

GroupsDataSet DataAccessServices.WebServices.Groups.GetDeploymentGroupsLight () [inline]

Returns a filtered list of group keys/names in the standard data set.

Returns:

Data set containing groups within the database.

Requires deployment or administrative access.

DeploymentGroupDto [] DataAccessServices.WebServices.Groups.GetDeploymentGroupsLightDto () [inline]

Returns an ordered list of all the group names.

Returns:

Array of DeploymentGroupDto's.

***EventFilterDataSet DataAccessServices.WebServices.Groups.GetEventFilter (Guid* groupKey*)* `[inline]`**

Returns the event filter for a group.

Parameters:

groupKey	The key of the group.

Returns:

A data set consisting of all of the events in the groups event filter.

Requires deployment or administrative access.

***GroupsDataSet DataAccessServices.WebServices.Groups.GetForCertificate (Guid* packageKey*, Guid* certificateKey*)* `[inline]`**

Get the groups that have the certificate, defined by its package key and certificate key.

Parameters:

packageKey	
certificateKey	

Returns:

The groups that have the certificate assigned.

GroupsDataSet DataAccessServices.WebServices.Groups.GetForPackageVersion (Guid* packageKey*, int* major*, int* minor*, int* build*, int* revision*) `[inline]`

Get the groups that have the specified package version assigned.

Parameters:

packageKey	
major	
minor	
build	
revision	

Returns:

GroupsDataSet DataAccessServices.WebServices.Groups.GetForPatch (Guid* packageKey*, Guid* patchKey*) `[inline]`

Get the groups that have the specified patch assigned.

Parameters:

packageKey	
patchKey	

Returns:

***GroupsDataSet DataAccessServices.WebServices.Groups.GetGroupFromKey (Guid* groupKey, *bool* withSummary*)* `[inline]`**

Returns a single group based on a group key.

Parameters:

groupKey	The key that identifies the group to return.
withSummary	Whether to include summary information.

Returns:

The specified group if it exists.

Requires deployment or administrative access.

***GroupPackagesDataSet DataAccessServices.WebServices.Groups.GetGroupPackages (Guid* groupKey*)* `[inline]`**

Returns a list of packages assigned to a group.

Parameters:

groupKey	The group key.

Returns:

Requires deployment or administrative access.

***GroupsDataSet DataAccessServices.WebServices.Groups.GetGroups (bool* withSummary*)* `[inline]`**

Returns all groups known to the database.

Parameters:

withSummary	Whether to include summary information.

Returns:

All groups within the database.

Requires deployment or administrative access.

***NamedValuesDataSet DataAccessServices.WebServices.Groups.GetInfo (Guid?* groupKey*)* `[inline]`**

Returns a data set consisting statistical information about the specified group.

Parameters:

groupKey	The key of the group.

Returns:

Data set containing statistical information about the specified group.

***ScheduleDataSet DataAccessServices.WebServices.Groups.GetInstallationSchedule (Guid* groupKey*)* `[inline]`**

Returns an installation schedule for the supplied group.

Parameters:

groupKey	The key which identifies the installation schedule.

Returns:

A data set consisting of the installation schedule for the group.

Requires deployment or administrative access.

***StatisticsDto DataAccessServices.WebServices.Groups.GetStatistics ()* `[inline]`**

Get a statistics summary of all groups, computers and alerts.

Returns:

A StatisticsDto object containing information about the groups, computers and alerts.

This data is used on the home page of the Management Console.

void DataAccessServices.WebServices.Groups.RemoveGroupPackage (Guid* groupKey*, Guid* packageKey*, DateTime?* modifiedTime*) `[inline]`

Removes an existing package from a group.

Parameters:

groupKey	The key of the group.
packageKey	The key of the package.
modifiedTime	The date time the group package row was last modified.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.RemoveGroupPatch (Guid* groupKey*, Guid* patchKey*, DateTime?* modifiedTime*) `[inline]`

Removes an existing patch from a group.

Parameters:

groupKey	The key of the group.
patchKey	The key of the patch.
modifiedTime	The date time the group package row was last modified.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.UpdateEventFilter (Guid* groupKey*, int* eventDefinitionKey*, bool* enabled*) `[inline]`

Adds or removed an event from the event filter of a group, depending upon whether the event is enabled.

Parameters:

groupKey	The group to add/remove the event filter to/from.
eventDefinitionKey	The key of the event definition.
enabled	Whether the event definition is enabled or disabled.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.UpdateGroup (Guid* groupKey*, string* name*, string* description*, int* pollPeriodSeconds*, int* uploadPollPeriodSeconds*, byte* eventLogEnabled*, byte* fileLogEnabled*, string* fileLogFilename*, bool* anonymousUserLogging*, bool* anonymousMachineLogging*, bool* overrideServerUrls*, Guid?* policyKey*, string* ownerSid*, ref DateTime* modifiedTime*, bool* selfRegistrationEnabled*, bool* selfUnregistrationEnabled*, bool* selfUpdateEnabled*, int* priority*, int* pollPeriodVariationSeconds*, int* uploadPollPeriodVariationSeconds*, bool* nativeConfigurations*, string* configurationLocation*) [inline]

Updates the properties of an existing group.

Parameters:

groupKey	The key which identifies this group.
name	The name of the group.
description	Description of the group.
pollPeriodSeconds	The amount of seconds between polls.
uploadPollPeriodSeconds	The amount of seconds between uploads.
eventLogEnabled	Whether to log events to the event log.
fileLogEnabled	Whether to log events to the file log.
fileLogFilename	The file name to log events to.
anonymousUserLogging	Whether events from machines within this group will be logged with anonymous users.
anonymousMachineLogging	Whether events from machines within this group will be logged with anonymous machines.
overrideServerUrls	Whether this group overrides server URL's.
policyKey	Policy asssociated with this group.
ownerSid	SID of user owning this group.
modifiedTime	The time that the group was modified.
selfRegistrationEnabled	Whether or not self-registration is allowed.

selfUnregistrationEnabled	Whether or not self unregistration is allowed.
selfUpdateEnabled	Whether or not self-update of agents and configurations is allowed.
priority	Order in which this groups membership rules will be evaluated.
pollPeriodVariationSeconds	The VariationSeconds allowed for each poll.
uploadPollPeriodVariationSeconds	The VariationSeconds allowed for each upload.
nativeConfigurations	Whether configurations be deployed to this group in native format.
configurationLocation	Location on agent machine to store native configurations.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.UpdateGroupLatestPackage (Guid* groupKey*, Guid* packageKey*, ref DateTime* modifiedTime*) ***`[inline]`***

Updates the version of a package within a group.

Parameters:

groupKey	The key of the group.
packageKey	The key of the package.
modifiedTime	Time the group package was last modified.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.UpdateGroupPackage (Guid* groupKey*, Guid* packageKey*, Guid?* patchKey*, int* major*, int* minor*, int* build*, int* revision*, ref DateTime* modifiedTime*) ***`[inline]`***

Updates the version of a package within a group.

Parameters:

groupKey	The key of the group.
packageKey	The key of the package.

patchKey	The patch GUID.
major	Major version of package version to use.
minor	Minor version of package version to use.
build	Build of package version to use.
revision	Revision of package version to use.
modifiedTime	Time the group package was last modified.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.UpdateGroupSecurity (Guid* groupKey*, Guid?* policyKey*, string* ownerSid*, ref DateTime* modifiedTime*) ***[inline]***

Updates security options for a group.

Parameters:

groupKey	The key which identifies this group.
policyKey	Policy asssociated with this group.
ownerSid	SID of user owning this group.
modifiedTime	The time that the group was last modified.

void DataAccessServices.WebServices.Groups.UpdateInstallationSchedule (Guid* groupKey*, long* mondayFlags*, long* tuesdayFlags*, long* wednesdayFlags*, long* thursdayFlags*, long* fridayFlags*, long* saturdayFlags*, long* sundayFlags*, int* flags*, bool* doNotInstallAgent*, ref DateTime* modifiedTime*, long* mondayConfigFlags*, long* tuesdayConfigFlags*, long* wednesdayConfigFlags*, long* thursdayConfigFlags*, long* fridayConfigFlags*, long* saturdayConfigFlags*, long* sundayConfigFlags*, int* configFlags*, bool* doNotInstallConfig*, bool* mirrorAgentSchedule*, bool* postponeInstallationPromptEnabled*, int* postponeInstallationLimit*, bool* downloadConfigurationsOnStartup*, bool* midSessionUpdate*, bool* updateAtShutdown*) ***[inline]***

Updates the installation schedule for an individual group.

Parameters:

groupKey	The key of the group to update.

mondayFlags	The flags for Monday.
tuesdayFlags	The flags for Tuesday.
wednesdayFlags	The flags for Wednesday.
thursdayFlags	The flags for Thursday.
fridayFlags	The flags for Friday.
saturdayFlags	The flags for Saturday.
sundayFlags	The flags for Sunday.
flags	The global flags.
doNotInstallAgent	Indicates agent installation has been disabled.
modifiedTime	The time that the installation schedule was last modified.
mondayConfigFlags	The flags for Monday Config.
tuesdayConfigFlags	The flags for Tuesday Config.
wednesdayConfigFlags	The flags for Wednesday Config.
thursdayConfigFlags	The flags for Thursday Config.
fridayConfigFlags	The flags for Friday Config.
saturdayConfigFlags	The flags for Saturday Config.
sundayConfigFlags	The flags for Sunday Config.
configFlags	The global config flags.
doNotInstallConfig	Indicates the config installation schedule has been disabled.
mirrorAgentSchedule	Indicates the config schedule should mirror the agent schedule.
postponeInstallationPromptEnabled	Set for postpone prompt for scheduled installs.
postponeInstallationLimit	Max limit for postponement in seconds.
downloadConfigurationsOnStartup	Download and install configurations before signaling agents to start.

midSessionUpdate	Whether non-rebooting packages can be installed or uninstalled on computers mid-session.
updateAtShutdown	Whether packages can be installed or uninstalled on computers at shutdown.

Requires deployment or administrative access.

The documentation for this class was generated from the following file:

- Groups.cs

DataAccessServices.WebServices.Housekeeping Class Reference

Summary description for Events.

Detailed Description

Summary description for Events.

The documentation for this class was generated from the following file:

- Housekeeping.cs

DataAccessServices.WebServices.Licenses Class Reference

Manages licenses in the Management Centre. When using the AppSense Management Suite a valid license must be used.

Public Member Functions

- void AddLicense (Guid licenseKey, String licenseCode, String activationCode, Int32 licenseType, Guid productKey, Int32 licenseCount, DateTime expiryDate, Boolean baseLicense, out DateTime modifiedTime)

  Adds a license.

- void AddLicenseV2 (String licensingKey, out DateTime modifiedTime)

  Adds a license using the new V2 license schema.

- void ApplyChanges (ref LicensesDataSet licenseChanges)

  Updates the database with the changes in the data set.

- void ApplyLicensingSettings (LicensingSettings settings)

  Applies any licensing settings information to the Properties table in the database.

- void DeleteLicense (Guid licenseKey, DateTime? modifiedTime)

  Deletes the specified license.

- void DeleteLicenseV1FromReg (String licenceCode)

  Deletes the specified license from the registry on the server.

- void DeleteLicenseV2 (Guid licenseID, DateTime? modifiedTime)

  Removes the specified license from the database.

- LicensesDataSet GetLicenses ()

  Returns a data set containing all licenses.

- LicensingSettings GetLicensingSettings ()

  Returns any settings related to licensing.

- LicensingV2DataSet GetV2Licenses ()

  Returns a data set containing all V2 licenses.

- void SetLicenses (DataSet ds)

  Set the license dataset

- void UpdateLicenses (Guid licenseKey, String licenseCode, String activationCode, Int32 licenseType, Guid productKey, Int32 licenseCount, DateTime expiryDate, Boolean baseLicense, ref DateTime modifiedTime)

  Updates the specified license.

Detailed Description

Manages licenses in the Management Centre. When using the AppSense Management Suite a valid license must be used.

Member Function Documentation

void DataAccessServices.WebServices.Licenses.AddLicense (Guid* licenseKey*, String* licenseCode*, String* activationCode*, Int32* licenseType*, Guid* productKey*, Int32* licenseCount*, DateTime* expiryDate*, Boolean* baseLicense*, out DateTime* modifiedTime*) `[inline]`

Adds a license.

Parameters:

licenseKey	GUID to identify this license.
licenseCode	License code for this license.
activationCode	Activation code for this license.
licenseType	Type of license.
productKey	Product GUID this license applies to.
licenseCount	Number of machines.
expiryDate	Expiry date for this license.
baseLicense	Whether or not this license is a base license.
modifiedTime	OUT parameter will contain the latest modified time of the license on return. Passed in value unused.

***void DataAccessServices.WebServices.Licenses.AddLicenseV2 (String* licensingKey*, out DateTime* modifiedTime*)* `[inline]`**

Adds a license using the new V2 license schema.

Parameters:

licensingKey	String containing the license encrypted using a base 64 encyption.
modifiedTime	OUT parameter will contain the latest modified time of the license on return. Passed in value unused.

***void DataAccessServices.WebServices.Licenses.ApplyChanges (ref LicensesDataSet* licenseChanges*)* `[inline]`**

Updates the database with the changes in the data set.

Parameters:

licenseChanges	A data set containing changes to licenses.

***void DataAccessServices.WebServices.Licenses.ApplyLicensingSettings (LicensingSettings* settings*)* `[inline]`**

Applies any licensing settings information to the Properties table in the database.

Parameters:

disableFileBased Population	Sets file-based license population to disabled. This prevents license files in a well-known location (typically PROGRAMDATA%/AppSense/Licenses) from being pulled into the database as part of a schedule.

void DataAccessServices.WebServices.Licenses.DeleteLicense (Guid* licenseKey*, DateTime?* modifiedTime*) ***[inline]***

Deletes the specified license.

Parameters:

licenseKey	GUID identifying the license.
modifiedTime	Time this license was last modified.

void DataAccessServices.WebServices.Licenses.DeleteLicenseV1FromReg (String **licenceCode*)* ***[inline]***

Deletes the specified license from the registry on the server.

Parameters:

licenceCode	String identifying the license.

void DataAccessServices.WebServices.Licenses.DeleteLicenseV2 (Guid **licenseID*, DateTime?* modifiedTime*)*** ***[inline]***

Removes the specified license from the database.

Parameters:

licenseID	GUID identifying the V2 license ID.
modifiedTime	Time this license was last modified.

LicensesDataSet DataAccessServices.WebServices.Licenses.GetLicenses () ***[inline]***

Returns a data set containing all licenses.

Returns:

Data set containing all licenses.

LicensingSettings DataAccessServices.WebServices.Licenses.GetLicensingSettings () `[inline]`

Returns any settings related to licensing.

Returns:

Data set containing all V2 licenses.

LicensingV2DataSet DataAccessServices.WebServices.Licenses.GetV2Licenses () `[inline]`

Returns a data set containing all V2 licenses.

Returns:

Data set containing all V2 licenses.

void DataAccessServices.WebServices.Licenses.SetLicenses (DataSet* **ds**) `[inline]`

Set the license dataset

Parameters:

ds	

void DataAccessServices.WebServices.Licenses.UpdateLicenses (Guid* **licenseKey**, *String* **licenseCode**, *String* **activationCode**, *Int32* **licenseType**, *Guid* **productKey**, *Int32* **licenseCount**, *DateTime* **expiryDate**, *Boolean* **baseLicense**, *ref DateTime* **modifiedTime**) `[inline]`

Updates the specified license.

Parameters:

licenseKey	GUID identifying the license.
licenseCode	License code for this license.
activationCode	Activation code for this license.
licenseType	Type of license.
productKey	Product GUID this license applies to.

licenseCount	Number of machines.
expiryDate	Expiry date for this license.
baseLicense	Whether or not this license is a base license.
modifiedTime	Time this license was last modified.

The documentation for this class was generated from the following file:

- Licenses.cs

DataAccessServices.WebServices.Machines Class Reference

Manages machines in the Management Centre. The Machines table stores an entry for each machine managed by the Management Center.

Public Member Functions

- void AddMachineDetails (Guid machineKey, string name, string value)

  Adds some detailed description about a machine.

- void AddMachinePackage (Guid machineKey, Guid packageKey, string name, string company, string type, PackagePlatform platform, Version version, DeploymentStatus status, Version childVersion, DeploymentStatus? childStatus, Version patchVersion, Guid? patchCode, DeploymentStatus? patchStatus, Version patchChildVersion, Guid? patchChildCode, DeploymentStatus? patchChildStatus, string statusMessage, Guid? certificateKey, out DateTime modifiedTime)

  Associates a package with a machine.

- void ApplyAndUpdateMachines (ref MachinesDataSet machineChanges)

  Updates the database with all changes within the supplied data set.

- void ApplyMachineChanges (ref MachinesDataSet machineChanges)

  Updates the database with all changes within the supplied data set.

- void ApplyMachineDetailsChanges (ref MachineDetailsDataSet machineDetailsChanges)

  Applies edits within the machine details data set into the database.

- void ApplyMachineDiagnosticsStateChanges (ref MachinesDataSet machineChanges)

  Updates the database with all changes within the supplied data set.

- void ApplyMachinePackageChanges (ref MachinePackagesDataSet machinePackageChanges)

  Applies changes from a machines packages data set into the database.

- void AutoMove (Guid[] machineIds)
- *Automatically moves the given machines into their expected group.*
- void CheckMachineGroupPermissions (Guid[] machineIds, string userSid, [Parameter(StructuredTypeName="dbo.StringList2")] DataTable groupSids, ObjectPermissions permissions)

  Deprecated.

- int Count (Guid groupKey)

  Returns the number of machines overall, or in a specific group.

- void CreateMachine (Guid machineKey, string netBiosName, string dns, MachinePlatform platform, MachineDiagnosticsState diagnosticsState, string distinguishedName, Guid objectGuid, Guid? groupKey, out DateTime modifiedTime)

  Creates a new machine.

- void DeleteMachine (Guid machineKey, DateTime? modifiedTime)

  Deletes an existing machine.

- MachinesDataSet FindMachines (string match)

  Retrieves all of the machines that match the search string.

- DeploymentStatisticsDto GetDeploymentStatistics (Guid? groupId=null)

  Get a statistics summary of all groups, computers and alerts.

- MachinesDataSet GetFromDns (string dns, bool withSummary)

  Returns a machine from a given fully qualified domain name.

- MachinesDataSet GetFromGroupKey (Guid groupKey, bool withSummary)

  Returns all machines within a specific group.

- MachinesDataSet GetFromGroupKeyDelta (Guid consoleId, Guid groupKey, bool withSummary, ref DateTime? lastRefresh)

  Returns all changed machines within a specific group since the last refresh.

- MachinesDataSet GetFromKey (Guid machineKey, bool withSummary)

  Returns a single machine.

- MachinesDataSet GetFromObjectGuid (Guid objectGuid, bool withSummary)

  Returns a machine from a given objectGUID attribute.

- MachineDetailsDataSet GetMachineDetails (Guid machineKey)

  Retrieves the machine details for a specified machine.

- MachineDiagnosticsDataSet GetMachineDiagnostics (Guid machineKey)

  Retrieves the diagnostic details for a specified machine.

- MachinesDataSet GetMachineFromDnsAndNetbios (string dns, string netbiosName, bool withSummary)

  Returns a machine from a given NetBIOS name and a DNS.

- MachinePackagesDataSet GetMachinePackages (Guid machineKey)

  Returns a set of packages associated with the supplied machine.

- MachinesDataSet GetMachines (bool withSummary)

  Returns all machines

- MachinesDataSet GetMachinesDelta (Guid consoleId, bool withSummary, ref DateTime? lastRefresh)

  Returns all machines changed since the last refresh

- MachinesDataSet GetPendingDeletion (bool withSummary)

  Returns all machines which have the IsPendingDeletion flag set to true.

- MachinesDataSet GetPreloadFromGroupKey (Guid groupKey, int preloadAmount)

Returns the specified number of machines from the start of the Machines table belonging to the specified group

- MachinesDataSet GetPreloadMachines (int preloadAmount)

  Returns the specified number of machines from the start of the Machines table

- MachinesDataSet GetWithObjectGuid (bool withSummary)

  Returns all machines which contain an objectGUID.

- MachinesDataSet GetWithPackage (Guid packageKey)

  Returns the machines that have a particular package installed.

- MachinesDataSet GetWithPackageVersion (Guid packageKey, string version)

  Returns the machines that have a particular package version installed.

- void Move (Guid destinationGroupKey, Guid[] machineIds)

  Move a list of machines from one group to another.

- void RemoveMachineDetails (Guid machineKey, string name)

  Removes descriptive information from a machine.

- void RemoveMachinePackage (Guid machineKey, Guid packageKey, DateTime? modifiedTime)

  Removes a package from a machine.

- void UpdateMachine (Guid machineKey, string netBiosName, string dns, MachinePlatform platform, MachineDiagnosticsState diagnosticsState, string distinguishedName, string oldDistinguishedName, Guid objectGuid, Guid? groupKey, bool isPendingDeletion, bool offline, DateTime modifiedGroupTime, ref DateTime modifiedTime)

  Updates an existing machine.

- void UpdateMachineDetails (Guid machineKey, string name, string value)

  Updates some detailed description about a machine.

- void UpdateMachineDiagnosticsState (Guid machineKey, bool diagnosticsError, MachineDiagnosticsState diagnosticsState, out DateTime modifiedTime)

  Updates an existing machine.

- void UpdateMachinePackage (Guid machineKey, Guid packageKey, string name, string company, string type, PackagePlatform platform, Version version, DeploymentStatus status, Version childVersion, DeploymentStatus? childStatus, Version patchVersion, Guid? patchCode, DeploymentStatus? patchStatus, Version patchChildVersion, Guid? patchChildCode, DeploymentStatus? patchChildStatus, string statusMessage, Guid? certificateKey, ref DateTime modifiedTime)

  Updates a package associated with a machine.

Detailed Description

Manages machines in the Management Centre. The Machines table stores an entry for each machine managed by the Management Center.

Member Function Documentation

void DataAccessServices.WebServices.Machines.AddMachineDetails (Guid* machineKey*, string* name*, string* value*) `[inline]`

Adds some detailed description about a machine.

Parameters:

machineKey	The key that identifies the machine.
name	The name of the descriptive information.
value	The value of the descriptive information.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.AddMachinePackage (Guid* machineKey*, Guid* packageKey*, string* name*, string* company*, string* type*, PackagePlatform* platform*, Version* version*, DeploymentStatus* status*, Version* childVersion*, DeploymentStatus?* childStatus*, Version* patchVersion*, Guid?* patchCode*, DeploymentStatus?* patchStatus*, Version* patchChildVersion*, Guid?* patchChildCode*, DeploymentStatus?* patchChildStatus*, string* statusMessage*, Guid?* certificateKey*, out DateTime* modifiedTime*) `[inline]`

Associates a package with a machine.

Parameters:

machineKey	The key of the machine.
packageKey	The key of the package.
name	The name of the package.
company	The company that supplied the package.
type	The type of the package, such as software or config.
platform	The platform of the package.

version	The version number of the package.
status	The status.
childVersion	The child version.
childStatus	The child status.
patchVersion	The version number of the patch.
patchCode	The patch code.
patchStatus	The patch status.
patchChildVersion	The patch child version.
patchChildCode	The patch child code.
patchChildStatus	The patch child status.
statusMessage	The status message.
certificateKey	The certificate key, or null if the package isn't a certificate.
modifiedTime	OUT parameter will contain the latest modified time of the license on return. Passed in value unused.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.ApplyAndUpdateMachines (ref MachinesDataSet* machineChanges*) `[inline]`

Updates the database with all changes within the supplied data set.

Parameters:

machineChanges	The data set consisting of the changes.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.Machines.ApplyMachineChanges (ref MachinesDataSet* machineChanges*)* `[inline]`**

Updates the database with all changes within the supplied data set.

Parameters:

machineChanges	The data set consisting of the changes.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.Machines.ApplyMachineDetailsChanges (ref MachineDetailsDataSet* machineDetailsChanges*)* `[inline]`**

Applies edits within the machine details data set into the database.

Parameters:

machineDetailsChanges	A data set consisting of the machine details edits.

Requires deployment or administrative access.

***void DataAccessServices.WebServices.Machines.ApplyMachineDiagnosticsStateChanges (ref MachinesDataSet* machineChanges*)* `[inline]`**

Updates the database with all changes within the supplied data set.

Parameters:

machineChanges	The data set consisting of the changes.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.ApplyMachinePackageChanges (ref MachinePackagesDataSet* machinePackageChanges*) [inline]

Applies changes from a machines packages data set into the database.

Parameters:

machinePackageChanges	A machine packages data set consisting of changes.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.AutoMove (Guid []* machineIds*) [inline]

Automatically moves the given machines into their expected group.

Parameters:

machineIds	Array of ids for the machines to move.

void DataAccessServices.WebServices.Machines.CheckMachineGroupPermissions (Guid []* machineIds*, string* userSid*, [Parameter(StructuredTypeName = "dbo.StringList2")] DataTable* groupSids*, ObjectPermissions* permissions*) [inline]

Deprecated.

int DataAccessServices.WebServices.Machines.Count (Guid* groupKey*) [inline]

Returns the number of machines overall, or in a specific group.

Parameters:

groupKey	The key for a group, or Guid.Empty for the overall count.

Returns:

Number of machines overall, or in a specific group.

void DataAccessServices.WebServices.Machines.CreateMachine (Guid* machineKey*, string* netBiosName*, string* dns*, MachinePlatform* platform*, MachineDiagnosticsState* diagnosticsState*, string* distinguishedName*, Guid* objectGuid*, Guid?* groupKey*, out DateTime* modifiedTime*) ***`[inline]`***

Creates a new machine.

Parameters:

machineKey	The key which identifies this machine.
netBiosName	The NetBIOS name of the machine.
dns	Specifies a fully qualified domain name for the machine.
platform	The platform of the machine, 32bit or 64bit.
diagnosticsState	State of the diagnostics.
distinguishedName	The distinguished name of the machine (optional).
objectGuid	The objectGUID attribute (optional).
groupKey	The group that the machine is attached to (optional).
modifiedTime	OUT parameter will contain the latest modified time of the license on return. Passed in value unused.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.DeleteMachine (Guid* machineKey*, DateTime?* modifiedTime*) ***`[inline]`***

Deletes an existing machine.

Parameters:

machineKey	The machine to delete.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.FindMachines (string* match*)* `[inline]`**

Retrieves all of the machines that match the search string.

Parameters:

match	The search string to match.

Returns:

Data set containing all the machines that match the search string.

Requires deployment or administrative access.

***DeploymentStatisticsDto DataAccessServices.WebServices.Machines.GetDeploymentStatistics (Guid?* groupId = `null`*)* `[inline]`**

Get a statistics summary of all groups, computers and alerts.

Parameters:

groupId	The group id to filter the deployment statistics or NULL to return all data.

Returns:

Statistics Dto

***MachinesDataSet DataAccessServices.WebServices.Machines.GetFromDns (string* dns*, bool* withSummary*)* `[inline]`**

Returns a machine from a given fully qualified domain name.

Parameters:

dns	Specifies a fully qualified domain name for the machine.
withSummary	Whether to include summary information.

Returns:

A data set consisting of a machine with a matching NetBIOS name if found.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetFromGroupKey (Guid* groupKey*, bool* withSummary*)* `[inline]`**

Returns all machines within a specific group.

Parameters:

groupKey	The group that the returned machines belong to.
withSummary	Whether to include summary information.

Returns:

A data set consisting of the machines in the specified group.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetFromGroupKeyDelta (Guid* consoleId*, Guid* groupKey*, bool* withSummary*, ref DateTime?* lastRefresh*)* `[inline]`**

Returns all changed machines within a specific group since the last refresh.

Parameters:

consoleId	Guid to identify the console making the request.
groupKey	The group that the returned machines belong to.
withSummary	Whether to include summary information.
lastRefresh	DateTime of the last refresh.

Returns:

A data set consisting of the changed machines since the last refresh in the specified group.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetFromKey (Guid* machineKey*, bool* withSummary*)* `[inline]`**

Returns a single machine.

Parameters:

machineKey	The machine to return.
withSummary	Whether to include summary information.

Returns:

A data set consisting of a single machine.

Requires deployment or administrative access.

MachinesDataSet DataAccessServices.WebServices.Machines.GetFromObjectGuid (Guid* objectGuid*, bool* withSummary*) `[inline]`

Returns a machine from a given objectGUID attribute.

Parameters:

objectGuid	The objectGUID attribute returned from the LDAP server.
withSummary	Whether to include summary information.

Returns:

A data set consisting of a machine with a matching objectGUID attribute if found.

Requires deployment or administrative access.

MachineDetailsDataSet DataAccessServices.WebServices.Machines.GetMachineDetails (Guid* machineKey*) `[inline]`

Retrieves the machine details for a specified machine.

Parameters:

machineKey	The key which identifies the machine.

Returns:

A data set consisting of the machine details.

Requires deployment or administrative access.

MachineDiagnosticsDataSet DataAccessServices.WebServices.Machines.GetMachineDiagnostics (Guid machineKey) `[inline]`

Retrieves the diagnostic details for a specified machine.

Parameters:

machineKey	The key which identifies the machine.

Returns:

A data set consisting of the machine diagnostics.

Requires deployment or administrative access.

MachinesDataSet DataAccessServices.WebServices.Machines.GetMachineFromDnsAndNetbios (string dns, string netbiosName, bool withSummary) `[inline]`

Returns a machine from a given NetBIOS name and a DNS.

Parameters:

dns	The name of the DNS machine to return.
netbiosName	The name of the NetBIOS machine to return.
withSummary	Whether to include summary information.

Returns:

A data set consisting of a machine with a matching NetBIOS name if found.

Requires deployment or administrative access.

***MachinePackagesDataSet DataAccessServices.WebServices.Machines.GetMachinePackages (Guid* machineKey*)* `[inline]`**

Returns a set of packages associated with the supplied machine.

Parameters:

machineKey	The machine key.

Returns:

A data set of all packages applied to the machine.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetMachines (bool* withSummary*)* `[inline]`**

Returns all machines

Parameters:

withSummary	Whether to include summary information.

Returns:

A data set consisting of all the machines.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetMachinesDelta (Guid* consoleId*, bool* withSummary*, ref DateTime?* lastRefresh*)* `[inline]`**

Returns all machines changed since the last refresh.

Parameters:

consoleId	Guid to identify the console making the request.
withSummary	Whether to include summary information.
lastRefresh	DateTime of the last refresh.

Returns:

A data set consisting of the machines in the specified group.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetPendingDeletion (bool* withSummary*)* `[inline]`**

Returns all machines which have the IsPendingDeletion flag set to true.

Parameters:

withSummary	Whether to include summary information.

Returns:

A dataset consisting of all machines pending deletion.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetPreloadFromGroupKey (Guid* groupKey, *int* preloadAmount*)* `[inline]`**

Returns the specified number of machines from the start of the Machines table belonging to the specified group.

Parameters:

groupKey	Guid of the group.
preloadAmount	Number of machines to get.

Returns:

Data set containing the machines belonging to the specified group.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetPreloadMachines (int* preloadAmount*)* `[inline]`**

Returns the specified number of machines from the start of the Machines table.

Parameters:

preloadAmount	Number of machines to get.

Returns:

Data set containing the machines.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetWithObjectGuid (bool* withSummary*)* `[inline]`**

Returns all machines which contain an objectGUID.

Parameters:

withSummary	Whether to include summary information.

Returns:

A data set consisting of all machines with an objectGUID attribute.

Requires deployment or administrative access.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetWithPackage (Guid* packageKey*)* `[inline]`**

Returns the machines that have a particular package installed.

Parameters:

packageKey	GUID specifying a key for the associated package.

Returns:

Data set containing machines that have a particular package installed.

***MachinesDataSet DataAccessServices.WebServices.Machines.GetWithPackageVersion (Guid* packageKey*, string* version*)* `[inline]`**

Returns the machines that have a particular package version installed.

Parameters:

packageKey	GUID specifying a key for the associated package.
version	The version number of the package.

Returns:

Data set containing the machines that have a particular package version installed.

***void DataAccessServices.WebServices.Machines.Move (Guid* destinationGroupKey*, Guid []* machineIds*)* `[inline]`**

Move a list of machines from one group to another.

Parameters:

destinationGroupKey	Guid of the group to move the machines into.
machineIds	Array of ids for the machines to move.

***void DataAccessServices.WebServices.Machines.RemoveMachineDetails (Guid* machineKey*, string* name*)* `[inline]`**

Removes descriptive information from a machine.

Parameters:

machineKey	The key that identifies the machine.
name	The name of the descriptive information to remove.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.RemoveMachinePackage (Guid* machineKey*, Guid* packageKey*, DateTime?* modifiedTime*) `[inline]`

Removes a package from a machine.

Parameters:

machineKey	The key of the machine.
packageKey	The key of the package.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.UpdateMachine (Guid* machineKey*, string* netBiosName*, string* dns*, MachinePlatform* platform*, MachineDiagnosticsState* diagnosticsState*, string* distinguishedName*, string* oldDistinguishedName*, Guid* objectGuid*, Guid?* groupKey*, bool* isPendingDeletion*, bool* offline*, DateTime* modifiedGroupTime*, ref DateTime* modifiedTime*) `[inline]`

Updates an existing machine.

Parameters:

machineKey	The key which identifies this machine.
netBiosName	The NetBIOS name of the machine.
dns	Specifies a fully qualified domain name for the machine.
platform	The platform of the machine, 32bit or 64bit.

diagnosticsState	State of the diagnostics.
distinguishedName	The distinguished name of the machine (optional).
oldDistinguishedName	The old distinguished name, if distinguishedName has changed.
objectGuid	The objectGUID attribute (optional).
groupKey	The group that the machine is attached to (optional).
isPendingDeletion	Whether this machine has been marked for deletion.
offline	If set to true machine is shown as offline.
modifiedGroupTime	The modified group time.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.UpdateMachineDetails (Guid* machineKey*, string* name*, string* value*) `[inline]`

Updates some detailed description about a machine.

Parameters:

machineKey	The key that identifies the machine.
name	The name of the descriptive information.
value	The value of the descriptive information.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.UpdateMachineDiagnosticsState (Guid* machineKey*, bool* diagnosticsError*, MachineDiagnosticsState* diagnosticsState*, out DateTime* modifiedTime*) ***`[inline]`***

Updates an existing machine.

Parameters:

machineKey	The key which identifies this machine.
diagnosticsError	If set to true show a diagnostics error has occurred.
diagnosticsState	State of the diagnostics.
modifiedTime	OUT parameter will contain the latest modified time of the machine on return. Passed in value unused.

Requires deployment or administrative access.

void DataAccessServices.WebServices.Machines.UpdateMachinePackage (Guid* machineKey*, Guid* packageKey*, string* name*, string* company*, string* type*, PackagePlatform* platform*, Version* version*, DeploymentStatus* status*, Version* childVersion*, DeploymentStatus?* childStatus*, Version* patchVersion*, Guid?* patchCode*, DeploymentStatus?* patchStatus*, Version* patchChildVersion*, Guid?* patchChildCode*, DeploymentStatus?* patchChildStatus*, string* statusMessage*, Guid?* certificateKey*, ref DateTime* modifiedTime*) ***`[inline]`***

Updates a package associated with a machine.

Parameters:

machineKey	The key of the machine.
packageKey	The key of the package.
name	The name of the package.
company	The company that supplied the package.
type	The type of the package, such as software or config.
platform	The platform of the package.
version	The version number of the package.
status	The status.

childVersion	The child version.
childStatus	The child status.
patchVersion	The version number of the patch.
patchCode	The patch code.
patchStatus	The patch status.
patchChildVersion	The patch child version.
patchChildCode	The patch child code.
patchChildStatus	The patch child status.
statusMessage	The status message.
certificateKey	The certificate key, or null if the package isn't a certificate.
modifiedTime	Datetime specifying when this item was last modified by the application - used for concurrency purposes.

Requires deployment or administrative access.

The documentation for this class was generated from the following file:

- Machines.cs

DataAccessServices.WebServices.Maintenance Class Reference

Manages maintenance tasks in the Management Centre. Within the AppSense Management Suite the product agents can raise a number of different events/alerts to the Management Center. In order to receive and display these events/alerts the Management Center database contains a list of all the possible evnets/alerts that can be raised via the product agents.

Public Member Functions

- void CancelSchedule (Int64 jobId, DateTime modifiedTime)

  Cancels a running schedule

- MaintenanceSchedulesDataSet GetSchedules ()

  Returns a dataset of all Scheduled maintenance actions

- Int64 Maintenance_AddScheduleRemoveEventsFromDeploymentGroup (string groupName, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool isTemporary=false)

  Add a scheduled maintenance task to remove events associated with a group.

- Int64 Maintenance_AddScheduleRemoveEventsOlderThan (int daysToRetain, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool isTemporary=false)

  Add a scheduled maintenance task to remove events older than the specified time.

- Int64 Maintenance_AddScheduleRemoveEventsWithId (int eventId, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool isTemporary=false)

  Add a scheduled maintenance task to remove events for a specific event ID

- Int64 Maintenance_AddScheduleRemoveEventsWithIdRange (int startEventId, int endEventId, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool isTemporary=false)

  Add a scheduled maintenance task to delete events for event IDs in a given range.

- Int64 Maintenance_AddScheduleRemoveHighVolumeEventsOlderThan (int daysToRetain, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool isTemporary=false)

  Add a scheduled maintenance task to remove high volume events older than a specific date and time

- Int64 Maintenance_AddScheduleRemoveOrphanedAlerts (string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool isTemporary=false)

  Add a scheduled maintenance task to remove orphaned alerts.

- Int64 Maintenance_AddScheduleRemoveUnresponsiveMachines (int daysToRetain, string groupName, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool isTemporary=false)

Add a scheduled maintenance task to remove machines that haven't polled in for a specified number of days.

- void Maintenance_DisableSchedule (Int64 jobId, DateTime modifiedTime)

  Disable the Scheduled Maintenance task specified by the supplied jobId.

- void Maintenance_EnableSchedule (Int64 jobId, DateTime modifiedTime)

  Enable the Scheduled Maintenance task specified by the supplied jobId.

- void Maintenance_LockSchedulingJob (Int64 jobId, String activeServer, out bool lockSuccess)

  This method sets the task ActiveServer name if the task is not running and the table is not locked.

- MaintenanceHelper.PreviewEventsDto Maintenance_RemoveAllEventsFromDeploymentGroup (string deploymentGroup, long batchSize, long batchDelay)

  Removes all product events from the specified depoyment group.

- MaintenanceHelper.PreviewEventsDto Maintenance_RemoveAllEventsOlderThanXDays (int daysToRetain, long batchSize, long batchDelay)

  Removes all product events older than the specified number of days.

- MaintenanceHelper.PreviewEventsDto Maintenance_RemoveAllEventsWithId (int eventId, long batchSize, long batchDelay)

  Removes all product events with the specified identifier.

- MaintenanceHelper.PreviewEventsDto Maintenance_RemoveAllEventsWithinIdRange (int startEventId, int endEventId, long batchSize, long batchDelay)

  Removes all product events within the specified event range

- MaintenanceHelper.PreviewEventsDto Maintenance_RemoveAllHighVolumeEventsOlderThanXDays (int daysToRetain, long batchSize, long batchDelay)

  Removes all high volume product events older than the specified cutoff.

- MaintenanceHelper.PreviewAlertsDto Maintenance_RemoveAllOrphanedAlerts (Int64 batchSize, long batchDelay)

  Removes orphaned alerts from within the database.

- void Maintenance_RemoveSchedule (Int64 jobId, DateTime? modifiedTime)

  Removes the entry in the Scheduled Maintenance table specified by the supplied jobId.

- void Maintenance_ResetSchedule (Int64 jobId, DateTime modifiedTime)

  Removes the entry in the Scheduled Maintenance table specified by the supplied jobId.

- void Maintenance_UnlockSchedulingJob (Int64 jobId, DateTime dtCompleted, out bool lockSuccess)

  This method sets the task back to unlocked by clearing the ActiveServer name.

- void Maintenance_UpdateSchedule (Int64 jobId, string description, string maintenanceType, TimeSpan startTime, DayOfWeek dayOfWeek, int maxRunHours, int batchSize, int batchDelay, string parameters, string activeServer, ScheduleJobStatus jobStatus, DateTime? jobCompleted, bool enabled, bool isTemporary)

Updates the entry in the Scheduled Maintenance table specified by the supplied jobId, using the other supplied parameters. Used by the scheduler service to push updates to the database.

- void Maintenance_UpdateScheduleRemoveEventsFromDeploymentGroup (Int64 jobId, string groupName, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, DateTime modifiedTime)

  Update the scheduled task's attributes specified by the supplied jobId.

- void Maintenance_UpdateScheduleRemoveEventsOlderThan (Int64 jobId, int daysToRetain, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool changeTypeToHighVolumeEventsOlderThan, DateTime modifiedTime)

  Update the scheduled task's attributes specified by the supplied jobId.

- void Maintenance_UpdateScheduleRemoveEventsWithId (Int64 jobId, int eventId, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, DateTime modifiedTime)

  Update the scheduled task's attributes specified by the supplied jobId.

- void Maintenance_UpdateScheduleRemoveEventsWithIdRange (Int64 jobId, int startEventId, int endEventId, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, DateTime modifiedTime)

  Update the scheduled task's attributes specified by the supplied jobId.

- void Maintenance_UpdateScheduleRemoveHighVolumeEventsOlderThan (Int64 jobId, int daysToRetain, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, bool changeTypeToOlderThan, DateTime modifiedTime)

  Update the scheduled task's attributes specified by the supplied jobId.

- void Maintenance_UpdateScheduleRemoveOrphanedAlerts (Int64 jobId, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, DateTime modifiedTime)

  Update the scheduled task's attributes specified by the supplied jobId.

- void Maintenance_UpdateScheduleRemoveUnresponsiveMachines (Int64 jobId, int daysToRetain, string groupName, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, DateTime modifiedTime)

  Update the scheduled task's attributes specified by the supplied jobId.

- EventsMaintenanceDataSet Preview_RemoveAllEventsFromDeploymentGroup (string deploymentGroup, long batchSize)

  Provides a preview of expected results for removing all product events from the specified depoyment group (see Maintenance_RemoveAllEventsFromDeploymentGroup).

- EventsMaintenanceDataSet Preview_RemoveAllEventsOlderThanXDays (int daysToRetain, long batchSize)

  Provides a preview of expected results for removing all product events older than the specified number of days (see Maintenance_RemoveAllEventsOlderThanXDays).

- EventsMaintenanceDataSet Preview_RemoveAllEventsWithId (int eventID, long batchSize)

Provides a preview of expected results for removing all event records that match the specified event ID (see Maintenance_RemoveAllEventsWithId).

- EventsMaintenanceDataSet Preview_RemoveAllEventsWithinIdRange (int startEventID, int endEventID, long batchSize)

  Provides a preview of expected results for removing all product events within the specified event range (see Maintenance_RemoveAllEventsWithinIdRange).

- EventsMaintenanceDataSet Preview_RemoveAllHighVolumeEventsOlderThanXDays (int daysToRetain, long batchSize)

  Provides a preview of expected results for removing all high volume product events older than the specified cutoff (see Maintenance_RemoveAllHighVolumeEventsOlderThan).

- AlertsMaintenanceDataSet Preview_RemoveAllOrphanedAlerts (Int64 batchSize)

  Provides preview results of the 'Maintenance_RemoveAllOphanedAlerts' web method (see above), and similarly calculates results in terms of the specified batch size so that we can aim to avoid long database locks.

Detailed Description

Manages maintenance tasks in the Management Centre. Within the AppSense Management Suite the product agents can raise a number of different events/alerts to the Management Center. In order to receive and display these events/alerts the Management Center database contains a list of all the possible evnets/alerts that can be raised via the product agents.

Each product agent in the AppSense suite can raise an event based on different conditions within the product agent. The Management Center must store all the different event definitions for each product in order to successfully report on any event that is configured and reported back via the CCA.

Member Function Documentation

void DataAccessServices.WebServices.Maintenance.CancelSchedule (Int64* jobId*, DateTime* modifiedTime*) `[inline]`

Cancels a running schedule.

Parameters:

jobId	The id of the job to cancel.
modifiedTime	The timestamp of the last user change for the job.

MaintenanceSchedulesDataSet DataAccessServices.WebServices.Maintenance.GetSchedules () `[inline]`

Returns a dataset of all Scheduled maintenance actions.

Returns:

.

Requires administrative access.

Int64 DataAccessServices.WebServices.Maintenance.Maintenance_AddScheduleRemoveEventsFromDeploymentGroup (string* groupName*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* isTemporary = *false)[inline]

Add a scheduled maintenance task to remove events associated with a group.

Parameters:

groupName	
description	
dayOfWeek	
startTimeTicks	
maxRunHours	
batchSize	
batchDelay	
enabled	
isTemporary	

Returns:

The id for the scheduled task.

Int64 DataAccessServices.WebServices.Maintenance.Maintenance_AddScheduleRemoveEventsOlderThan (int* daysToRetain*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* isTemporary = *false)[inline]

Add a scheduled maintenance task to remove events older than the specified time.

Parameters:

daysToRetain	
description	

dayOfWeek	
startTimeTicks	
maxRunHours	
batchSize	
batchDelay	
enabled	
isTemporary	

Returns:

The id of the scheduled task.

Int64 DataAccessServices.WebServices.Maintenance.Maintenance_AddScheduleRemoveEventsWithId (int* eventId*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* isTemporary = `false`) `[inline]`

Add a scheduled maintenance task to remove events for a specific event ID.

Parameters:

eventId	
description	
dayOfWeek	
startTimeTicks	
maxRunHours	
batchSize	
batchDelay	
enabled	
isTemporary	

Returns:

The task ID.

Int64 DataAccessServices.WebServices.Maintenance.Maintenance_AddScheduleRemoveEventsWithIdRange (int* startEventId*, int* endEventId*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* isTemporary = *false*)[inline]

Add a scheduled maintenance task to delete events for event IDs in a given range.

Parameters:

startEventId	
endEventId	
description	
dayOfWeek	
startTimeTicks	
maxRunHours	
batchSize	
batchDelay	
enabled	
isTemporary	

Returns:

The task ID.

Int64 DataAccessServices.WebServices.Maintenance.Maintenance_AddScheduleRemoveHighVolumeEventsOlderThan (int* daysToRetain*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* isTemporary = *false*)[inline]

Add a scheduled maintenance task to remove *high volume* events older than a specific date and time.

Parameters:

daysToRetain	
description	

dayOfWeek	
startTimeTicks	
maxRunHours	
batchSize	
batchDelay	
enabled	
isTemporary	

Returns:

The task ID.

***Int64 DataAccessServices.WebServices.Maintenance.Maintenance_AddScheduleRemoveOrphanedAlerts (string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* isTemporary = *false)* `[inline]`**

Add a scheduled maintenance task to remove orphaned alerts.

Parameters:

description	
dayOfWeek	
startTimeTicks	
maxRunHours	
batchSize	
batchDelay	
enabled	
isTemporary	

Returns:

The task ID.

Int64 DataAccessServices.WebServices.Maintenance.Maintenance_AddScheduleRemoveUnresponsiveMachines (int* daysToRetain*, string* groupName*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* isTemporary = `false`) `[inline]`

Add a scheduled maintenance task to remove machines that haven't polled in for a specified number of days.

Parameters:

daysToRetain	
groupName	
description	
dayOfWeek	
startTimeTicks	
maxRunHours	
batchSize	
batchDelay	
enabled	
isTemporary	

Returns:

The task ID.

void DataAccessServices.WebServices.Maintenance.Maintenance_DisableSchedule (Int64* jobId*, DateTime* modifiedTime) `[inline]`

Disable the Scheduled Maintenance task specified by the supplied jobId.

Returns:

.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_EnableSchedule (Int64 jobId, DateTime modifiedTime) [inline]

Enable the Scheduled Maintenance task specified by the supplied jobId.

Returns:

.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_LockSchedulingJob (Int64 jobId, String activeServer, out bool lockSuccess) [inline]

This method sets the task ActiveServer name if the task is not running and the table is not locked.

Parameters:

jobId	A unique job id created for the ScheduledMaintenance row.
activeServer	The server that is executing the job.
lockSuccess	This is a return value if the task was successfully locked.

MaintenanceHelper.PreviewEventsDto DataAccessServices.WebServices.Maintenance.Maintenance_RemoveAllEventsFromDeploymentGroup (string deploymentGroup, long batchSize, long batchDelay) [inline]

Removes all product events from the specified depoyment group.

Parameters:

deploymentGroup	The deployment group at which to target deletion action.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).
batchDelay	The delay between batches.

Returns:

PreviewEventsDto

Requires administrative access.

MaintenanceHelper.PreviewEventsDto DataAccessServices.WebServices.Maintenance.Maintenance_RemoveAllEventsOlderThanXDays (int* **daysToRetain, long* **batchSize***, long* **batchDelay***)* `[inline]`

Removes all product events older than the specified number of days.

Parameters:

daysToRetain	The number of days worth of events to keep after this operation completes.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).
batchDelay	The delay between batches.

Returns:

PreviewEventsDto

Requires administrative access.

MaintenanceHelper.PreviewEventsDto DataAccessServices.WebServices.Maintenance.Maintenance_RemoveAllEventsWithId (int* **eventId, long* **batchSize***, long* **batchDelay***)* `[inline]`

Removes all product events with the specified identifier.

Parameters:

eventId	The product event ID for which all records are to be deleted.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).
batchDelay	The delay between batches.

Returns:

PreviewEventsDto

Requires administrative access.

MaintenanceHelper.PreviewEventsDto DataAccessServices.WebServices.Maintenance.Maintenance_RemoveAllEventsWithinIdRange (int* startEventId*, int* endEventId*, long* batchSize*, long* batchDelay*) [inline]

Removes all product events within the specified event range.

Parameters:

startEventId	The lowest product event ID for which all records are to be deleted.
endEventId	The highest product event ID for which all records are to be deleted.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).
batchDelay	The delay between batches.

Returns:

PreviewEventsDto

Requires administrative access.

MaintenanceHelper.PreviewEventsDto DataAccessServices.WebServices.Maintenance.Maintenance_RemoveAllHighVolumeEventsOlderThanXDays (int* daysToRetain*, long* batchSize*, long* batchDelay*) [inline]

Removes all *high volume* product events older than the specified cutoff.

Parameters:

daysToRetain	The number of days worth of high volume events that should remain after this operation.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).
batchDelay	The delay between batches.

Returns:

PreviewEventsDto

Requires administrative access.

MaintenanceHelper.PreviewAlertsDto DataAccessServices.WebServices.Maintenance.Maintenance_RemoveAllOrphanedAlerts (Int64* **batchSize, long* **batchDelay***)* `[inline]`

Removes orphaned alerts from within the database.

Parameters:

batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).
batchDelay	The delay between batches.

Returns:

PreviewAlertsDto

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_RemoveSchedule (Int64 **jobId***, DateTime?* **modifiedTime***)* `[inline]`

Removes the entry in the Scheduled Maintenance table specified by the supplied jobId.

Returns:

.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_ResetSchedule (Int64 **jobId***, DateTime* **modifiedTime***)* `[inline]`

Removes the entry in the Scheduled Maintenance table specified by the supplied jobId.

Returns:

.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_UnlockSchedulingJob (Int64 **jobId***, DateTime* **dtCompleted***, out bool* **lockSuccess***)* `[inline]`

This method sets the task back to unlocked by clearing the ActiveServer name.

Parameters:

jobId	A unique job id created for the ScheduledMaintenance row.
dtCompleted	The time that the task is completed.
lockSuccess	This is a return value if the task was successfully unlocked.

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateSchedule (Int64* jobId*, string* description*, string* maintenanceType*, TimeSpan* startTime*, DayOfWeek* dayOfWeek*, int* maxRunHours*, int* batchSize*, int* batchDelay*, string* parameters*, string* activeServer*, ScheduleJobStatus* jobStatus*, DateTime?* jobCompleted*, bool* enabled*, bool* isTemporary*) [inline]

Updates the entry in the Scheduled Maintenance table specified by the supplied jobId, using the other supplied parameters. Used by the scheduler service to push updates to the database.

Returns:

.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateScheduleRemoveEventsFromDeploymentGroup (Int64* jobId*, string* groupName*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, DateTime* modifiedTime*) [inline]

Update the scheduled task's attributes specified by the supplied jobId.

Parameters:

jobId	Job id of scheduled task to update.
groupName	Name of deployment group for events to be removed by.
description	Description for the scheduled task.
dayOfWeek	Day of week the scheduled task is to run.
startTimeTicks	Time of day for schedule task to start in ticks since midnight (one tick equals 100ns.)

maxRunHours	The maximum number of hours for the task to run.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations.
batchDelay	The delay between batches.
enabled	The enabled state of the task.
modifiedTime	The modified time of the last user change.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateScheduleRemoveEventsOlderThan (Int64* jobId*, int* daysToRetain*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, bool* changeTypeToHighVolumeEventsOlderThan*, DateTime* modifiedTime*) [inline]

Update the scheduled task's attributes specified by the supplied jobId.

Parameters:

jobId	Job id of scheduled task to update.
daysToRetain	The number of days worth of events that should remain after this operation.
description	Description for the scheduled task.
dayOfWeek	Day of week the scheduled task is to run.
startTimeTicks	Time of day for schedule task to start in ticks since midnight (one tick equals 100ns).
maxRunHours	The maximum number of hours for the task to run.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations.
batchDelay	The delay between batches.
enabled	The enabled state of the task.
changeTypeToHighVolumeEventsOlderThan	Change task to be for *high volume* events.

modifiedTime	The modified time of the last user change.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateScheduleRemoveEventsWithId (Int64* jobId*, int* eventId*, string* description*, int* dayOfWeek*, long* startTimeTicks*, long* maxRunHours*, long* batchSize*, long* batchDelay*, bool* enabled*, DateTime* modifiedTime*) [inline]

Update the scheduled task's attributes specified by the supplied jobId.

Parameters:

jobId	Job id of scheduled task to update.
eventId	The product event ID for which all records are to be deleted.
description	Description for the scheduled task.
dayOfWeek	Day of week the scheduled task is to run.
startTimeTicks	Time of day for schedule task to start in ticks since midnight (one tick equals 100ns).
maxRunHours	Maximum run time.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations.
batchDelay	The delay between batches.
enabled	The enabled state of the task.
modifiedTime	The modified time of the last user change.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateScheduleRemoveEventsWithIdRange (Int64 jobId, int startEventId, int endEventId, string description, int dayOfWeek, long startTimeTicks, long maxRunHours, long batchSize, long batchDelay, bool enabled, DateTime modifiedTime) `[inline]`

Update the scheduled task's attributes specified by the supplied jobId.

Parameters:

jobId	Job id of scheduled task to update.
startEventId	The lowest product event ID for which all records are to be deleted.
endEventId	The highest product event ID for which all records are to be deleted.
description	Description for the scheduled task.
dayOfWeek	Day of week the scheduled task is to run.
startTimeTicks	Time of day for schedule task to start in ticks since midnight (one tick equals 100ns).
maxRunHours	Maximum run time.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations.
batchDelay	The delay between batches.
enabled	The enabled state of the task.
modifiedTime	The modified time of the last user change.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateScheduleRemoveHighVolumeEventsOlderThan (Int64* **jobId**, *int* **daysToRetain**, *string* **description**, *int* **dayOfWeek**, *long* **startTimeTicks**, *long* **maxRunHours**, *long* **batchSize**, *long* **batchDelay**, *bool* **enabled**, *bool* **changeTypeToOlderThan**, *DateTime* **modifiedTime**) `[inline]`

Update the scheduled task's attributes specified by the supplied jobId.

Parameters:

jobId	Job id of scheduled task to update.
daysToRetain	The number of days worth of high volume events that should remain after this operation.
description	Description for the scheduled task.
dayOfWeek	Day of week the scheduled task is to run.
startTimeTicks	Time of day for schedule task to start in ticks since midnight (one tick equals 100ns).
maxRunHours	Maximum run time.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations.
batchDelay	The delay between batches.
enabled	The enabled state of the task.
changeTypeToOlderThan	Change task to be not for *high volume* events.
modifiedTime	The modified time of the last user change.

Requires administrative access.

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateScheduleRemoveOrphanedAlerts (Int64* **jobId**, *string* **description**, *int* **dayOfWeek**, *long* **startTimeTicks**, *long* **maxRunHours**, *long* **batchSize**, *long* **batchDelay**, *bool* **enabled**, *DateTime* **modifiedTime**) `[inline]`

Update the scheduled task's attributes specified by the supplied jobId.

Parameters:

jobId	Job id of scheduled task to update.
description	Description for the scheduled task.
dayOfWeek	Day of week the scheduled task is to run.
startTimeTicks	Time of day for schedule task to start.
maxRunHours	Maximum run time.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations.
batchDelay	The delay between batches.
enabled	
modifiedTime	

void DataAccessServices.WebServices.Maintenance.Maintenance_UpdateScheduleRemoveUnresponsiveMachines (Int64* **jobId**, *int* **daysToRetain**, *string* **groupName**, *string* **description**, *int* **dayOfWeek**, *long* **startTimeTicks**, *long* **maxRunHours**, *long* **batchSize**, *long* **batchDelay**, *bool* **enabled**, *DateTime* **modifiedTime**) `[inline]`

Update the scheduled task's attributes specified by the supplied jobId.

Parameters:

jobId	Job id of scheduled task to update.
daysToRetain	The number of days worth of events to keep after this operation completes.
groupName	Name of deployment group for events to be removed by.

description	Description for the scheduled task.
dayOfWeek	Day of week the scheduled task is to run.
startTimeTicks	Time of day for schedule task to start.
maxRunHours	Maximum run time.
batchSize	How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations.
batchDelay	The delay between batches.
enabled	
modifiedTime	

Requires administrator access.

***EventsMaintenanceDataSet DataAccessServices.WebServices.Maintenance.Preview_RemoveAllEventsFromDeploymentGroup (string* deploymentGroup, *long* batchSize) `[inline]`**

Provides a preview of expected results for removing all product events from the specified depoyment group (see Maintenance_RemoveAllEventsFromDeploymentGroup).

Parameters:

deploymentGroup	The deployment group at which to target deletion action.
batchSize	How many rows would be deleted at a time.

Returns:

EventsMaintenanceDataSet.

Requires administrative access.

***EventsMaintenanceDataSet DataAccessServices.WebServices.Maintenance.Preview_RemoveAllEventsOlderThanXDays (int* daysToRetain*, long* batchSize*)* `[inline]`**

Provides a preview of expected results for removing all product events older than the specified number of days (see Maintenance_RemoveAllEventsOlderThanXDays).

Parameters:

daysToRetain	The number of days worth of events to keep after this operation completes.
batchSize	How many rows would be deleted at a time.

Returns:

EventsMaintenanceDataSet.

Requires administrative access.

***EventsMaintenanceDataSet DataAccessServices.WebServices.Maintenance.Preview_RemoveAllEventsWithId (int* eventID*, long* batchSize*)* `[inline]`**

Provides a preview of expected results for removing all event records that match the specified event ID (see Maintenance_RemoveAllEventsWithId).

Parameters:

eventID	The ID of the event that it to be deleted.
batchSize	How many rows would be deleted at a time.

Returns:

EventsMaintenanceDataSet.

Requires administrative access.

EventsMaintenanceDataSet DataAccessServices.WebServices.Maintenance.Preview_RemoveAllEventsWithinIdRange (int* startEventID*, int* endEventID*, long* batchSize*) [inline]

Provides a preview of expected results for removing all product events within the specified event range (see Maintenance_RemoveAllEventsWithinIdRange).

Parameters:

startEventID	The lowest product event ID for which all records are to be deleted.
endEventID	The highest product event ID for which all records are to be deleted.
batchSize	How many rows would be deleted at a time.

Returns:

EventsMaintenanceDataSet.

Requires administrative access.

EventsMaintenanceDataSet DataAccessServices.WebServices.Maintenance.Preview_RemoveAllHighVolumeEventsOlderThanXDays (int* daysToRetain*, long* batchSize*) [inline]

Provides a preview of expected results for removing all *high volume* product events older than the specified cutoff (see Maintenance_RemoveAllHighVolumeEventsOlderThan).

Parameters:

daysToRetain	The number of days worth of events to keep after this operation completes.
batchSize	How many rows would be deleted at a time.

Returns:

EventsMaintenanceDataSet.

Requires administrative access.

***AlertsMaintenanceDataSet DataAccessServices.WebServices.Maintenance.Preview_RemoveAllOrphanedAlerts (Int64* batchSize*)* `[inline]`**

Provides preview results of the 'Maintenance_RemoveAllOphanedAlerts' web method (see above), and similarly calculates results in terms of the specified batch size so that we can aim to avoid long database locks.

Parameters:

batchSize	How many rows would be deleted at a time.

Returns:

The alerts that would be removed.

The documentation for this class was generated from the following file:

- Maintenance.cs

DataAccessServices.WebServices.Packages Class Reference

Packages within the Management Center are stored as MSI files and comprise of either an agent or configuration. A package has one or more associated versions to support software and configuration versioning and concurrency control.

Public Member Functions

- void ApplyPackageChanges (ref PackagesDataSet packageChanges)
  Applies package changes from a packages data set. Note that this method currently only supports deletion of packages and package versions.
- void ApplyPackageSecurityChanges (ref PackagesDataSet packageChanges)
  Applies package changes from a packages data set. Note that this method currently only supports deletion of packages and package versions.
- void ApplyPrerequisiteChanges (ref PrerequisitesDataSet prerequisiteChanges)
  Applies the prerequisite changes.
- Guid BeginCertificateDownload (Guid certificateKey)
  Begins a certificate download.
- Guid BeginCertificateUpload (Guid packageKey, string description, Guid certificateKey, out DateTime modifiedTime, int dataLength)
  Begins the upload of a certificate.
- Guid BeginPackageVersionDownload (Guid packageKey, int major, int minor, int build, int revision)
  Begins a package download.
- Guid BeginPackageVersionUpload (Guid packageKey, string description, Guid packageVersionKey, out DateTime modifiedTime, int dataLength)
  Begins an upload of a package version.
- Guid BeginPatchDownload (Guid patchKey)
  Begins a patch download.
- Guid BeginPatchUpload (Guid patchKey, string description, out DateTime modifiedTime, int dataLength)
  Begins a patch upload.
- Guid BeginPrerequisiteResourceDownload (Guid resourceKey)
  Begins the prerequisite resource download.
- Guid BeginPrerequisiteResourceUpload (Guid resourceKey, int dataLength, out DateTime modifiedTime)
  Begins the prerequisite resource upload.
- void CommitCertificate (Guid certificateKey)
  Completes the upload process and makes the certificate available.
- void CommitPackageVersion (Guid packageVersionKey)
  Commits the package version.
- void CommitPatch (Guid patchKey)
  Commits a patch to the database.
- byte [] ContinueCertificateDownload (Guid downloadKey, int offset, int length)
  Continues a certificate download. The bytes of the package from 'offset' to 'offset + length' are returned.
- void ContinueCertificateUpload (Guid certificateKey, ref DateTime modifiedTime, Guid uploadKey, int offset, byte[] data)
  Continues the upload of a certificate.
- Byte [] ContinuePackageVersionDownload (Guid downloadKey, int offset, int length)

Continues a package download. The bytes of the package from 'offset' to 'offset + length' are returned.

- void ContinuePackageVersionUpload (Guid packageVersionKey, ref DateTime modifiedTime, Guid uploadKey, int offset, Byte [] data)
  Continues an upload of a package version.
- Byte [] ContinuePatchDownload (Guid downloadKey, int offset, int length)
  Continues the patch download.
- void ContinuePatchUpload (Guid patchKey, ref DateTime modifiedTime, Guid uploadKey, int offset, Byte[] data)
  Continues the patch upload.
- Byte [] ContinuePrerequisiteResourceDownload (Guid downloadKey, int offset, int length)
  Continues the prerequisite resource download.
- void ContinuePrerequisiteResourceUpload (Guid resourceKey, ref DateTime modifiedTime, Guid uploadKey, int offset, Byte[] data)
  Continues the prerequisite resource upload.
- void CreateCertificate (Guid packageKey, Guid certificateKey, string name, string description, string password, string thumbprints, DateTime earliestExpiry, out DateTime modifiedTime)
  Creates a new certificate for a package. There can only be one certificate for a package.
- void CreatePackage (Guid key, string company, string type, PackagePlatform platform, Guid productKey, out DateTime modifiedTime)
  Creates a configuration package within the database.
- void CreatePackageVersion (Guid packageKey, Guid packageVersionKey, string name, int major, int minor, int build, int revision, string marketingVersion, int creatorMajor, int creatorMinor, int creatorBuild, int creatorRevision, int dependentMinimumMajor, int dependentMinimumMinor, int dependentMinimumBuild, int dependentMinimumRevision, int dependentMaximumMajor, int dependentMaximumMinor, int dependentMaximumBuild, int dependentMaximumRevision, string description, bool supportsMidSessionUpdate, out DateTime modifiedTime)
  Creates the package version.
- void CreatePatch (Guid patchKey, Guid patchCode, Guid packageVersionKey, string name, int major, int minor, int build, int revision, string marketingVersion, int targetMajor, int targetMinor, int targetBuild, int targetRevision, int validationFlags, string description, bool supportsMidSessionUpdate, out DateTime modifiedTime)
  Creates a patch.
- void DeleteAndUnlockPackage (Guid packageKey)
  Delete a WIP package and unlock the parent as a single operation rolling back the operation if either fails
- void DeletePackage (Guid key, DateTime? modifiedTime)
  Deletes an existing package from the database.
- void DeletePrerequisite (Guid prerequsiteKey, DateTime? modifiedTime)
  Deletes the prerequisite.
- void FinalisePackageVersion (Guid packageVersionKey)
  Finalises the package version.
- void FinalizeCertificate (Guid certificateKey, string userName)
  Finalizes the certificate upload.
- PackagesDataSet GetAllDependentPatchesFromPatchKey (Guid patchKey)
  Returns all dependents for a given patch.
- PackagesDataSet GetCertificatePackages ()
  Gets the Package rows for the Certificates and the Certificates table
- string GetLegacyPrerequisitesXmlV1 (Guid productCode, string name, string version, string platform, Guid upgradeCode, string type, Guid productKey)

Gets the legacy prerequisites XML v1.

- string GetLegacyPrerequisitesXmlV2 (Guid productCode, string name, string version, string platform, Guid upgradeCode, string type, Guid productKey)
  Gets the legacy prerequisites XML v2.
- PackagesDataSet GetPackageFromKey (Guid key)
  Returns an individual package within the database.
- PackagesDataSet GetPackageFromKeyWithInProgress (Guid key)
  Gets the package from key with in progress.

  Parameters:

key	The key that identifies the package to return.

- PackagesDataSet GetPackageFromPackageVersionKey (Guid packageVersionKey)
  Returns a package and package version for a package version with a matching package version key.
- PackagesDataSet GetPackageFromProductKey (Guid productKey)
  Returns all packages and their versions that belong to a specific product.
- PackagesDataSet GetPackageFromProductName (string productName)
  Returns all packages and their versions that belong to a specific product.
- PackagesDataSet GetPackageFromType (string type)
  Returns all packages and their versions that are of a specific type.
- PackagesDataSet GetPackages ()
  Returns all packages and their versions stored within the database.
- PackagesDataSet GetPackagesWithInProgress ()
  Gets the packages with in progress.
- int GetPackageVersionLength (Guid packageKey, int major, int minor, int build, int revision)
  Determines the length of a version of a package.
- PackagesDataSet GetPatchesWithInProgress ()
  Gets the patches with in progress.
- PackagesDataSet GetPatchesWithMissingMetadata ()
  Gets existing patches that need updating with additional metadata.
- PackagesDataSet GetPatchFromPatchKey (Guid patchKey)
  Returns all dependent patches for a given patch.
- PackagesDataSet GetPatchInstallSequenceFromPatchCode (Guid patchCode)
  Returns all dependencies for a given patch code.
- PackagesDataSet GetPatchInstallSequenceFromPatchKey (Guid patchKey)
  Returns all dependencies for a given patch.
- int GetPatchLength (Guid patchKey)
  Determines the length of a patch.
- PrerequisitesDataSet GetPrerequisiteFromPrerequisiteKey (Guid prerequisiteKey)
  Gets the prerequisites from prerequisite key.
- PrerequisitesDataSet GetPrerequisiteFromPrerequisiteName (string prerequisiteName)
  Gets the prerequisites from the prerequisite name.
- PrerequisitesDataSet GetPrerequisites ()
  Gets the prerequisites for package version key.
- PrerequisitesDataSet GetPrerequisitesForPackageVersionKey (Guid packageVersionKey)

Gets the prerequisites from prerequisite key.

- PrerequisitesDataSet GetPrerequisitesForPatchCode (Guid patchCode)
  Gets the prerequisites from the patch code.
- PrerequisitesDataSet GetPrerequisitesFromXml (string xml)
  Gets the prerequisites from XML.
- void LockPackage (Guid packageKey)
  Locks the package.
- void QueryCommitPackageVersionStatus (Guid packageVersionKey, out AgentUploadStatus.UploadState status, out string errorString)
  The CommitPackageVersion launches a seperate thread within ProductAgentInitialize to complete the initialization of events, alerts, and prerequisites. This method allows the client to query whether that process has completed or not.
- void QueryCommitPatchStatus (Guid patchKey, out AgentUploadStatus.UploadState status, out string errorString)
  Checks if the Management Server has finished processing the specified patch.
- void RemovePackageVersion (Guid packageKey, int major, int minor, int build, int revision, DateTime? modifiedTime)
  Deletes a package version from a package.
- void RemovePatch (Guid patchKey, DateTime? modifiedTime)
  Deletes the patch.
- void UnlockPackage (Guid packageKey)
  Unlocks the package.
- void UpdatePackage (Guid key, string company, string type, PackagePlatform platform, Guid productKey, Guid? policyKey, string ownerSid, ref DateTime modifiedTime)
  Updates a package within the database.
- void UpdatePackageSecurity (Guid key, Guid? policyKey, string ownerSid, ref DateTime modifiedTime)
  Updates the package security.
- void **UpdatePrerequisitesForPatch** (Guid patchKey)
- void WIPSaved (Guid packageKey)
  Marks a package as a work-in-progress.

Detailed Description

Packages within the Management Center are stored as MSI files and comprise of either an agent or configuration. A package has one or more associated versions to support software and configuration versioning and concurrency control.

Member Function Documentation

void DataAccessServices.WebServices.Packages.ApplyPackageChanges (ref PackagesDataSet* packageChanges*) [inline]

Applies package changes from a packages data set.

Note

This method currently only supports deletion of packages and package versions.

Parameters:

packageChanges	The package changes.

void DataAccessServices.WebServices.Packages.ApplyPackageSecurityChanges (ref PackagesDataSet* packageChanges*) [inline]

Applies package changes from a packages data set.

Note

This method currently only supports deletion of packages and package versions.

Parameters:

packageChanges	The package changes.

void DataAccessServices.WebServices.Packages.ApplyPrerequisiteChanges (ref PrerequisitesDataSet* prerequisiteChanges*) `[inline]`

Applies the prerequisite changes.

Parameters:

prerequisiteChanges	The prerequisite changes.

***Guid DataAccessServices.WebServices.Packages.BeginCertificateDownload (Guid* certificateKey*)* `[inline]`**

Begins a certificate download.

Parameters:

certificateKey	The key which identifies the certificate.

Returns:

A guid which represents a download key.

***Guid DataAccessServices.WebServices.Packages.BeginCertificateUpload (Guid* packageKey*, string* description*, Guid* certificateKey*, out DateTime* modifiedTime*, int* dataLength*)* `[inline]`**

Begins the upload of a certificate.

Parameters:

packageKey	The key which represents the package.
description	Description of the specified item.
certificateKey	The key which uniquely identifies this certificate.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
dataLength	The length of data that will be uploaded.

Returns:

An upload key used to add data to the upload.

Guid DataAccessServices.WebServices.Packages.BeginPackageVersionDownload (Guid* packageKey*, int* major*, int* minor*, int* build*, int* revision*) `[inline]`

Begins a package download.

Parameters:

packageKey	The key which represents the package.
major	The major version of the package.
minor	The minor version of the package.
build	The build version of the package.
revision	The revision version of the package.

Returns:

A guid which represents a download key.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

Guid DataAccessServices.WebServices.Packages.BeginPackageVersionUpload (Guid* packageKey*, string* description*, Guid* packageVersionKey*, out DateTime* modifiedTime*, int* dataLength*) `[inline]`

Begins an upload of a package version.

Parameters:

packageKey	The key which represents the package.
description	Description of the specified item.
packageVersionKey	The key which uniquely identifies this package.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
dataLength	The length of data that will be uploaded.

Returns:

An upload key used to add data to the upload.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

Guid DataAccessServices.WebServices.Packages.BeginPatchDownload (Guid* patchKey*) `[inline]`

Begins a patch download.

Parameters:

patchKey	The patch key.

Returns:

A download key.

Guid DataAccessServices.WebServices.Packages.BeginPatchUpload (Guid **patchKey*****, string*** **description*****, out DateTime*** **modifiedTime*****, int*** **dataLength*****)*** `[inline]`

Begins a patch upload.

Parameters:

patchKey	The patch key.
description	Description for the patch.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
dataLength	Length of the data.

Returns:

An upload key.

Guid DataAccessServices.WebServices.Packages.BeginPrerequisiteResourceDownload (Guid* resourceKey*) `[inline]`

Begins the prerequisite resource download.

Parameters:

resourceKey	The resource key.

Returns:

A download key.

Guid DataAccessServices.WebServices.Packages.BeginPrerequisiteResourceUpload (Guid* resourceKey*, int* dataLength*, out DateTime* modifiedTime*) `[inline]`

Begins the prerequisite resource upload.

Parameters:

resourceKey	The resource key.
dataLength	Length of the data.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

Returns:

An upload key used to add data to the upload.

void DataAccessServices.WebServices.Packages.CommitCertificate (Guid* certificateKey*) `[inline]`

Completes the upload process and makes the certificate available.

Parameters:

certificateKey	The certificate key.

void DataAccessServices.WebServices.Packages.CommitPackageVersion (Guid* packageVersionKey*) `[inline]`

Commits the package version.

Parameters:

packageVersionKey	The package version key.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

void DataAccessServices.WebServices.Packages.CommitPatch (Guid patchKey*)*** `[inline]`

Commits a patch to the database.

Parameters:

patchKey	The patch key.

byte [] DataAccessServices.WebServices.Packages.ContinueCertificateDownload (Guid downloadKey*, int* offset*, int* length*)*** `[inline]`

Continues a certificate download. The bytes of the package from 'offset' to 'offset + length' are returned.

Parameters:

downloadKey	The key which represents the download.
offset	The offset of the first byte to return.
length	The number of bytes to download.

Returns:

Bytes from the package.

void DataAccessServices.WebServices.Packages.ContinueCertificateUpload (Guid* certificateKey*, ref DateTime* modifiedTime*, Guid* uploadKey*, int* offset*, byte [] * data*) `[inline]`

Continues the upload of a certificate.

Parameters:

certificateKey	The key which uniquely identifies this package.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
uploadKey	The upload key.
offset	The offset.
data	The data.

Byte [] DataAccessServices.WebServices.Packages.ContinuePackageVersionDownload (Guid* downloadKey*, int* offset*, int* length*) `[inline]`

Continues a package download. The bytes of the package from 'offset' to 'offset + length' are returned.

Parameters:

downloadKey	The key which represents the download.
offset	The offset of the first byte to return.
length	The number of bytes to download.

Returns:

Bytes from the package.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

void DataAccessServices.WebServices.Packages.ContinuePackageVersionUpload (Guid* packageVersionKey*, ref DateTime* modifiedTime*, Guid* uploadKey*, int* offset*, Byte []* data*) `[inline]`

Continues an upload of a package version.

Parameters:

packageVersionKey	The key which uniquely identifies this package.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
uploadKey	The upload key.
offset	The offset.
data	The data.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

Byte [] DataAccessServices.WebServices.Packages.ContinuePatchDownload (Guid* downloadKey*, int* offset*, int* length*) `[inline]`

Continues the patch download.

Parameters:

downloadKey	The download key.
offset	The offset.
length	The length.

Returns:

The next block of binary data representing the downloaded patch.

void DataAccessServices.WebServices.Packages.ContinuePatchUpload (Guid* patchKey*, ref DateTime* modifiedTime*, Guid* uploadKey*, int* offset*, Byte []* data*) `[inline]`

Continues the patch upload.

Parameters:

patchKey	The patch key.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
uploadKey	Length of the data.
offset	Length of the data.
data	Length of the data.

Byte [] DataAccessServices.WebServices.Packages.ContinuePrerequisiteResourceDownload (Guid* downloadKey*, int* offset*, int* length*) `[inline]`

Continues the prerequisite resource download.

Parameters:

downloadKey	The download key.
offset	The offset.
length	The length.

Returns:

The next block of binary data representing the downloaded prerequisites.

void DataAccessServices.WebServices.Packages.ContinuePrerequisiteResourceUpload (Guid* resourceKey*, ref DateTime* modifiedTime*, Guid* uploadKey*, int* offset*, Byte [] * data*) `[inline]`

Continues the prerequisite resource upload.

Parameters:

resourceKey	The resource key.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
uploadKey	The upload key.
offset	The offset.
data	The data.

void DataAccessServices.WebServices.Packages.CreateCertificate (Guid* packageKey*, Guid* certificateKey*, string* name*, string* description*, string* password*, string* thumbprints*, DateTime* earliestExpiry*, out DateTime* modifiedTime*) `[inline]`

Creates a new certificate for a package. There can only be one certificate for a package.

Parameters:

packageKey	The package key.
certificateKey	A unique key for the package.
name	A unique name for the certificate.
description	A description for the certificate.
password	The encrypted password to open the certificate.
thumbprints	A comma separated list of thumbprints for the cerificates in the file.
earliestExpiry	DateTine of the first certificate to expire.
modifiedTime	The modified time for the created entry.

void DataAccessServices.WebServices.Packages.CreatePackage (Guid* key*, string* company*, string* type*, PackagePlatform* platform*, Guid* productKey*, out DateTime* modifiedTime*) ***[inline]***

Creates a configuration package within the database.

Parameters:

key	The Guid which identifies the package.
company	The company name.
type	The type of the package, i.e. Software, Configuration.
platform	The platform that the package should be installed on.
productKey	The name of the product associated with this package.
modifiedTime	A date time indicating when the package was created.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

void DataAccessServices.WebServices.Packages.CreatePackageVersion (Guid* packageKey*, Guid* packageVersionKey*, string* name*, int* major*, int* minor*, int* build*, int* revision*, string* marketingVersion*, int* creatorMajor*, int* creatorMinor*, int* creatorBuild*, int* creatorRevision*, int* dependentMinimumMajor*, int* dependentMinimumMinor*, int* dependentMinimumBuild*, int* dependentMinimumRevision*, int* dependentMaximumMajor*, int* dependentMaximumMinor*, int* dependentMaximumBuild*, int* dependentMaximumRevision*, string* description*, bool* supportsMidSessionUpdate*, out DateTime* modifiedTime*) ***[inline]***

Creates the package version.

Parameters:

packageKey	GUID specifying a key for the associated package.
packageVersionKey	The package version key.

name	Name of the specified item.
major	The major.
minor	The minor.
build	The build.
revision	The revision.
marketingVersion	The marketing version for the package version.
creatorMajor	Major version of package creator (console).
creatorMinor	Minor version of package creator (console).
creatorBuild	Build version of package creator (console).
creatorRevision	Revision version of package creator (console).
dependentMinimumMajor	Major version of minimum associated agent.
dependentMinimumMinor	Minor version of minimum associated agent.
dependentMinimumBuild	Build version of minimum associated agent.
dependentMinimumRevision	Revision version of minimum associated agent.
dependentMaximumMajor	Major version of maximum associated agent.
dependentMaximumMinor	Minor version of maximum associated agent.
dependentMaximumBuild	Build version of maximum associated agent.
dependentMaximumRevision	Revision version of maximum associated agent.
description	Description of the specified item.

supportsMidSessionUpdate	Whether package can be installed or uninstalled on computers mid-session.
modifiedTime	A date time indicating when the package version was created.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

void DataAccessServices.WebServices.Packages.CreatePatch (Guid* patchKey*, Guid* patchCode*, Guid* packageVersionKey*, string* name*, int* major*, int* minor*, int* build*, int* revision*, string* marketingVersion*, int* targetMajor*, int* targetMinor*, int* targetBuild*, int* targetRevision*, int* validationFlags*, string* description*, bool* supportsMidSessionUpdate*, out DateTime* modifiedTime*) [inline]

Creates a patch.

Parameters:

patchKey	GUID specifying a key for the associated package.
patchCode	Patch Code property from MSP file.
packageVersionKey	The package version key.
name	Name of the specified item.
major	The major.
minor	The minor.
build	The build.
revision	The revision.
marketingVersion	The marketing version for the patch.
targetMajor	The major version of the patch or package that this patch updates.
targetMinor	The minor version of the patch or package that this patch updates.

targetBuild	The build version of the patch or package that this patch updates.
targetRevision	The revision version of the patch or package that this patch updates.
validationFlags	The Validation Flags property from the Windows Installer MSP file.
description	Description of the specified item.
supportsMidSessionUpdate	This defines whether an agent can be installed without requiring a reboot.
modifiedTime	Time that the package version was created.

void DataAccessServices.WebServices.Packages.DeleteAndUnlockPackage (Guid* packageKey*) ***`[inline]`***

Delete a WIP package and unlock the parent as a single operation rolling back the operation if either fails.

Parameters:

packageKey	The primary key of the package.

void DataAccessServices.WebServices.Packages.DeletePackage (Guid key**, *DateTime?*** modifiedTime***)*** ***`[inline]`***

Deletes an existing package from the database.

Parameters:

key	The Guid which identifies the package.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

***void DataAccessServices.WebServices.Packages.DeletePrerequisite (Guid* prerequsiteKey, *DateTime?* modifiedTime) `[inline]`**

Deletes the prerequisite.

Parameters:

prerequsiteKey	The prerequsite key.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

***void DataAccessServices.WebServices.Packages.FinalisePackageVersion (Guid* packageVersionKey) `[inline]`**

Finalises the package version.

Parameters:

packageVersionKey	The package version key.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

***void DataAccessServices.WebServices.Packages.FinalizeCertificate (Guid* certificateKey, *string* userName) `[inline]`**

Finalizes the certificate upload.

Parameters:

certificateKey	The certificate key.
userName	The user that is uploading the certificate.

***PackagesDataSet DataAccessServices.WebServices.Packages.GetAllDependentPatchesFromPatchKey (Guid* patchKey*)* `[inline]`**

Returns all dependents for a given patch.

Parameters:

patchKey	The key which represents the patch.

Returns:

A dataset consisting of all dependent patches.

***PackagesDataSet DataAccessServices.WebServices.Packages.GetCertificatePackages ()* `[inline]`**

Gets the Package rows for the Certificates and the Certificates table.

***string DataAccessServices.WebServices.Packages.GetLegacyPrerequisitesXmlV1 (Guid* productCode*, string* name*, string* version*, string* platform*, Guid* upgradeCode*, string* type*, Guid* productKey*)* `[inline]`**

Gets the legacy prerequisites XML v1.

Parameters:

productCode	The product code.
name	Name of the specified item.
version	The version number of the package.
platform	The platform.
upgradeCode	The upgrade code.
type	The type.
productKey	GUID specifying a key for the associated product.

Returns:

An XML string.

string DataAccessServices.WebServices.Packages.GetLegacyPrerequisitesXmlV2 (Guid* productCode*, string* name*, string* version*, string* platform*, Guid* upgradeCode*, string* type*, Guid* productKey*) `[inline]`

Gets the legacy prerequisites XML v2.

Parameters:

productCode	The product code.
name	Name of the specified item.
version	The version number of the package.
platform	The platform.
upgradeCode	The upgrade code.
type	The type.
productKey	GUID specifying a key for the associated product.

Returns:

An XML string.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPackageFromKey (Guid* key*) `[inline]`

Returns an individual package within the database.

Parameters:

key	The key that identifies the package to return.

Returns:

A data set describing the package and its versions.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible. Requires deployment or administrative access.

***PackagesDataSet DataAccessServices.WebServices.Packages.GetPackageFromKeyWithInProgress (Guid* key*)* `[inline]`**

Gets the package from key within progress.

Parameters:

key	The key that identifies the package to return.

Returns:

A data set describing the packages and its versions.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

***PackagesDataSet DataAccessServices.WebServices.Packages.GetPackageFromPackageVersionKey (Guid* packageVersionKey*)* `[inline]`**

Returns a package and package version for a package version with a matching package version key.

Parameters:

packageVersionKey	The key which uniquely identifies a package version.

Returns:

A data set populated with a package and a version.

Requires deployment or administrative access.

***PackagesDataSet DataAccessServices.WebServices.Packages.GetPackageFromProductKey (Guid* productKey*)* `[inline]`**

Returns all packages and their versions that belong to a specific product.

Parameters:

productKey	GUID specifying a key for the associated product.

Returns:

A data set consisting of the packages and their versions.

Requires deployment or administrative access.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPackageFromProductName (string* productName*) [inline]

Returns all packages and their versions that belong to a specific product.

Parameters:

productName	The name of the product.

Returns:

A data set consisting of the packages and their versions.

Requires deployment or administrative access.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPackageFromType (string* type*) [inline]

Returns all packages and their versions that are of a specific type.

Parameters:

type	The type of the package.

Returns:

A data set consisting of the packages and their versions.

Requires deployment or administrative access.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPackages () [inline]

Returns all packages and their versions stored within the database.

Returns:

A data set consisting of all packages and versions.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible. Requires deployment or administrative access.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPackagesWithInProgress () [inline]

Gets the packages within progress.

Returns:

A data set describing the packages and its versions.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

***int DataAccessServices.WebServices.Packages.GetPackageVersionLength (Guid* packageKey, *int* major, *int* minor, *int* build, *int* revision) [inline]**

Determines the length of a version of a package.

Parameters:

packageKey	The key which represents the package.
major	The major version of the package.
minor	The minor version of the package.

build	The build version of the package.
revision	The revision version of the package.

Returns:

The length of data in bytes.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPatchesWithInProgress () ***`[inline]`***

Gets the patches with in progress.

Returns:

A dataset consisting of all dependent patches.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPatchesWithMissingMetadata () ***`[inline]`***

Gets existing patches that need updating with additional metadata.

Returns:

A dataset consisting of all dependent patches.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPatchFromPatchKey (Guid **patchKey*)* ***`[inline]`***

Returns all dependent patches for a given patch.

Parameters:

patchKey	The key which represents the patch.

Returns:

A dataset consisting of all dependent patches.

***PackagesDataSet DataAccessServices.WebServices.Packages.GetPatchInstallSequenceFromPatchCode (Guid* patchCode*)* `[inline]`**

Returns all dependencies for a given patch code.

Parameters:

patchCode	The Patch Code property from the Windows Installer MSP file.

Returns:

A dataset consisting of all dependent patches.

***PackagesDataSet DataAccessServices.WebServices.Packages.GetPatchInstallSequenceFromPatchKey (Guid* patchKey*)* `[inline]`**

Returns all dependencies for a given patch.

Parameters:

patchKey	The key which represents the patch.

Returns:

A dataset consisting of all dependent patches.

***int DataAccessServices.WebServices.Packages.GetPatchLength (Guid* patchKey*)* `[inline]`**

Determines the length of a patch.

Parameters:

patchKey	The patch key.

Returns:

The length of data in bytes.

PrerequisitesDataSet DataAccessServices.WebServices.Packages.GetPrerequisiteFromPrerequisiteKey (Guid* prerequisiteKey*) `[inline]`

Gets the prerequisites from prerequisite key.

Parameters:

prerequisiteKey	The prerequisite key.

Returns:

A prerequisite data set.

PrerequisitesDataSet DataAccessServices.WebServices.Packages.GetPrerequisiteFromPrerequisiteName (string* prerequisiteName*) `[inline]`

Gets the prerequisites from the prerequisite name.

Parameters:

prerequisiteName	Name of the prerequisite.

Returns:

A prerequisite data set.

PrerequisitesDataSet DataAccessServices.WebServices.Packages.GetPrerequisites () `[inline]`

Gets the prerequisites for package version key.

Returns:

A prerequisite data set.

PrerequisitesDataSet DataAccessServices.WebServices.Packages.GetPrerequisitesForPackageVersionKey (Guid* packageVersionKey*) ***[inline]***

Gets the prerequisites from prerequisite key.

Parameters:

packageVersionKey	The package version key.

Returns:

A prerequisite data set.

PrerequisitesDataSet DataAccessServices.WebServices.Packages.GetPrerequisitesForPatchCode (Guid* patchCode*) ***[inline]***

Gets the prerequisites from the patch code.

Parameters:

patchCode	The patch code.

Returns:

A prerequisite data set.

PrerequisitesDataSet DataAccessServices.WebServices.Packages.GetPrerequisitesFromXml (string* xml*) ***[inline]***

Gets the prerequisites from XML.

Parameters:

xml	The XML.

Returns:

A prerequisite data set.

***void DataAccessServices.WebServices.Packages.LockPackage (Guid* packageKey*)* `[inline]`**

Locks the package.

Parameters:

packageKey	GUID specifying a key for the associated package.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

***void DataAccessServices.WebServices.Packages.QueryCommitPackageVersionStatus (Guid* packageVersionKey*, out AgentUploadStatus.UploadState* status*, out string* errorString*)* `[inline]`**

The CommitPackageVersion launches a seperate thread within ProductAgentInitialize to complete the initialization of events, alerts, and prerequisites. This method allows the client to query whether that process has completed or not.

Parameters:

packageVersionKey	The package version key.
status	The current upload status.
errorString	The error string.

***void DataAccessServices.WebServices.Packages.QueryCommitPatchStatus (Guid* patchKey*, out AgentUploadStatus.UploadState* status*, out string* errorString*)* `[inline]`**

Checks if the Management Server has finished processing the specified patch.

Parameters:

patchKey	The patch key.
status	The current upload status.
errorString	The error string.

***void DataAccessServices.WebServices.Packages.RemovePackageVersion (Guid* packageKey*, int* major*, int* minor*, int* build*, int* revision*, DateTime?* modifiedTime*)* `[inline]`**

Deletes a package version from a package.

Parameters:

packageKey	The key which identifies the package which contains the package version.
major	The major version number of the package to remove.
minor	The minor version number of the package to remove.
build	The build version number of the package to remove.
revision	The revision number of the package to remove.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

void DataAccessServices.WebServices.Packages.RemovePatch (Guid* patchKey, *DateTime?* modifiedTime*) [inline]

Deletes the patch.

Parameters:

patchKey	The patch key.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

void DataAccessServices.WebServices.Packages.UnlockPackage (Guid* packageKey*) [inline]

Unlocks the package.

Parameters:

packageKey	GUID specifying a key for the associated package.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

void DataAccessServices.WebServices.Packages.UpdatePackage (Guid* key, *string* company, *string* type, *PackagePlatform* platform, *Guid* productKey, *Guid?* policyKey, *string* ownerSid, *ref DateTime* modifiedTime*) [inline]

Updates a package within the database.

Parameters:

key	The Guid which identifies the package.
company	The company name.
type	The type of the package, i.e. Software, Configuration.
platform	The platform that the package should be installed on.

productKey	The name of the product associated with this package.
policyKey	The policy key.
ownerSid	The owner sid.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

***void DataAccessServices.WebServices.Packages.UpdatePackageSecurity (Guid* key, *Guid?* policyKey, *string* ownerSid, *ref DateTime* modifiedTime) `[inline]`**

Updates the package security.

Parameters:

key	The Guid which identifies the package.
policyKey	The policy key.
ownerSid	The owner sid.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

***void DataAccessServices.WebServices.Packages.WIPSaved (Guid* packageKey) `[inline]`**

Marks a package as a work-in-progress.

Parameters:

packageKey	The key which represents the package.

Warning

This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

The documentation for this class was generated from the following file:

- Packages.cs

DataAccessServices.WebServices.Products Class Reference

The ProductsWebService creates product entries needed for uploading and managing packages. For more information, refer to the PackagesWebService.

Public Member Functions

- void ApplyChanges (ref ProductsDataSet productChanges)

  Updates the database with the changes in the data set.

- void CreateProduct (Guid productKey, String name, Byte [] icon, Boolean supportsAgents, Boolean supportsConfigurations, Boolean supportsSoftware, string highestVersionNumber, out DateTime modifiedTime)

  Creates a new product.

- void DeleteProduct (Guid productKey, DateTime? modifiedTime)

  Deletes a product.

- ProductsDataSet GetProducts ()

  Returns a data set consisting of all products.

- void UpdateProduct (Guid productKey, String name, Byte [] icon, Boolean supportsAgents, Boolean supportsConfigurations, Boolean supportsSoftware, string highestVersionNumber, ref DateTime modifiedTime)

  Updates an existing product.

Detailed Description

The ProductsWebService creates product entries needed for uploading and managing packages. For more information, refer to the PackagesWebService.

Member Function Documentation

void DataAccessServices.WebServices.Products.ApplyChanges (ref ProductsDataSet* productChanges*) `[inline]`

Updates the database with the changes in the data set.

Parameters:

productChanges	The product changes.

void DataAccessServices.WebServices.Products.CreateProduct (Guid* productKey*, String* name*, Byte []* icon*, Boolean* supportsAgents*, Boolean* supportsConfigurations*, Boolean* supportsSoftware*, string* highestVersionNumber*, out DateTime* modifiedTime*) `[inline]`

Creates a new product.

Parameters:

productKey	GUID specifying a key for the associated product.
name	Name of the specified item.
icon	Byte array containing a Windows Icon file.
supportsAgents	Whether or not the product can support agents.
supportsConfigurations	Whether or not the product can support configuration files.
supportsSoftware	Whether or not the product can support software such as consoles.
highestVersionNumber	
modifiedTime	A date time indicating when the product was created.

void DataAccessServices.WebServices.Products.DeleteProduct (Guid* productKey*, DateTime?* modifiedTime*) `[inline]`

Deletes a product.

Parameters:

productKey	GUID specifying a key for the associated product.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

ProductsDataSet DataAccessServices.WebServices.Products.GetProducts () `[inline]`

Returns a data set consisting of all products.

Returns:

A data set consisting of all products.

void DataAccessServices.WebServices.Products.UpdateProduct (Guid* productKey*, String* name*, Byte []* icon*, Boolean* supportsAgents*, Boolean* supportsConfigurations*, Boolean* supportsSoftware*, string* highestVersionNumber*, ref DateTime* modifiedTime*) `[inline]`

Updates an existing product.

Parameters:

productKey	GUID specifying a key for the associated product.
name	Name of the specified item.
icon	Byte array containing a Windows Icon file.
supportsAgents	Whether or not the product can support agents.
supportsConfigurations	Whether or not the product can support configuration files.
supportsSoftware	Whether or not the product can support software such as consoles.
highestVersionNumber	
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

The documentation for this class was generated from the following file:

- Products.cs

DataAccessServices.WebServices.Queries Class Reference

Provides methods for querying the Management Server database via the reporting engine.

Public Member Functions

- DataSet Execute (String queryString, Int32 timeout)
- *Returns a data set containing the results of the specified query.*

Detailed Description

Provides methods for querying the Management Server database via the reporting engine.

Member Function Documentation

DataSet DataAccessServices.WebServices.Queries.Execute (String* queryString*, Int32* timeout*) `[inline]`

Returns a data set containing the results of the specified query.

Parameters:

queryString	The query to execute, in serialised XML.
timeout	The timeout before the query is cancelled (in milliseconds).

Returns:

Data set containing the results of the specified query.

The documentation for this class was generated from the following file:

- Queries.cs

DataAccessServices.WebServices.Reports Class Reference

Manages reports in the Management Centre. Report definitions store the necessary data for generating the reports visible within the Management Console.

Public Member Functions

- void ApplyReportDefinitionChanges (ref ReportDefinitionsDataSet reportDefinitionChanges)

  Applies changes within a report definition data set.

- void ApplyReportDefinitionSecurityChanges (ref ReportDefinitionsDataSet reportDefinitionChanges)

  Updates owner sid for report definitions specified in a report definition data set.

- Guid BeginReportDefinitionDownload (Guid reportDefinitionKey)

  Begins downloading a report definition, allowing for chunked streaming of data.

- Guid BeginReportDefinitionUpload (Guid reportDefinitionKey, ref DateTime modifiedTime, Int32 dataLength)

  Begins an upload of some report definition data.

- Byte [] ContinueReportDefinitionDownload (Guid downloadKey, Int32 offset, Int32 length)

  Continues downloading a report definition, after receiving a download key from BeginReportDefinitionDownload.

- void ContinueReportDefinitionUpload (Guid reportDefinitionKey, ref DateTime modifiedTime, Guid uploadKey, Int32 offset, Byte [] data)

  Continues uploading a report definition, after creation of an upload key from BeginReportDefinitionUpload.

- void CreateReportDefinition (Guid reportDefinitionKey, String name, String description, String category, String type, Guid productKey, Boolean visible, DateTime? fileTime, out DateTime modifiedTime)

  Creates a new report definition within the database. The report definition will only be available when data has been added.

- void CreateReportDefinitions (ReportDefinitionList reportDefinitionList, Boolean compress)

  Creates the supplied report definitions within the database.

- void DeleteReportDefinition (Guid reportDefinitionKey, DateTime? modifiedTime)

  Deletes a report definition from the database.

- void DeleteReportDefinitions (ReportDefinitionList reportDefinitionList)

  Deletes the supplied report definitions from the database.

- ReportDefinitionsDataSet GetReportDefinition (Guid reportDefinitionKey)

  Returns a specific report definition from the database.

- ReportDefinitionsDataSet GetReportDefinitions ()

Returns a list of all report definitions stored within the database.

- ReportDefinitionsDataSet GetVisibleReportDefinitions (Boolean visible)

  Returns report definitions filtered on the visible property.

- void UpdateReportDefinition (Guid reportDefinitionKey, String name, String description, String category, String type, Guid productKey, Boolean visible, Guid? policyKey, String ownerSid, DateTime modifiedTime)

  Updates the report definition.

- void UpdateReportDefinitionSecurity (Guid reportDefinitionKey, Guid? policyKey, String ownerSid, ref DateTime modifiedTime)

  Updates the report definition security.

Detailed Description

Manages reports in the Management Centre. Report definitions store the necessary data for generating the reports visible within the Management Console.

Member Function Documentation

void DataAccessServices.WebServices.Reports.ApplyReportDefinitionChanges (ref ReportDefinitionsDataSet* reportDefinitionChanges*) [inline]

Applies changes within a report definition data set.

Parameters:

reportDefinitionChanges	The report definition changes.

void DataAccessServices.WebServices.Reports.ApplyReportDefinitionSecurityChanges (ref ReportDefinitionsDataSet* reportDefinitionChanges*) [inline]

Updates owner sid for report definitions specified in a report definition data set.

Parameters:

reportDefinitionChanges	The report definition changes containing the owner sids to apply.

Guid DataAccessServices.WebServices.Reports.BeginReportDefinitionDownload (Guid* reportDefinitionKey*) `[inline]`

Begins downloading a report definition, allowing for chunked streaming of data.

Parameters:

reportDefinitionKey	The report definition key.

Returns:

Download Key to be passed to ContinueReportDefinitionDownload to continue downloading the report definition

Guid DataAccessServices.WebServices.Reports.BeginReportDefinitionUpload (Guid* reportDefinitionKey*, ref DateTime* modifiedTime*, Int32* dataLength*) `[inline]`

Begins an upload of some report definition data.

Parameters:

reportDefinitionKey	The report definition key.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
dataLength	Length of the data to be written.

Returns:

Upload Key to be passed to ContinueReportDefinitionUpload to continue uploading the report definition.

Byte [] DataAccessServices.WebServices.Reports.ContinueReportDefinitionDownload (Guid* downloadKey*, Int32* offset*, Int32* length*) `[inline]`

Continues downloading a report definition, after receiving a download key from BeginReportDefinitionDownload.

Parameters:

downloadKey	The download key returned by BeginReportDefinitionDownload.
offset	The offset to start reading the data from.
length	Length of the data to read.

Returns:

Array of bytes read from the database.

void DataAccessServices.WebServices.Reports.ContinueReportDefinitionUpload (Guid* reportDefinitionKey*, ref DateTime* modifiedTime*, Guid* uploadKey*, Int32* offset*, Byte []* data*) `[inline]`

Continues uploading a report definition, after creation of an upload key from BeginReportDefinitionUpload.

Parameters:

reportDefinitionKey	The report definition key.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.
uploadKey	The upload key returned by BeginReportDefinitionUpload.
offset	The offset to start writing the data from.
data	Length of the data.

void DataAccessServices.WebServices.Reports.CreateReportDefinition (Guid* reportDefinitionKey*, String* name*, String* description*, String* category*, String* type*, Guid* productKey*, Boolean* visible*, DateTime?* fileTime*, out DateTime* modifiedTime*) `[inline]`

Creates a new report definition within the database. The report definition will only be available when data has been added.

Parameters:

reportDefinitionKey	The report definition key.
name	Name of the specified item.
description	Description of the specified item.
category	The category.
type	The type.
productKey	GUID specifying a key for the associated product.
visible	If set to true report is visible.
fileTime	The modified time for the file.
modifiedTime	OUT parameter will contain the latest modified time of the report definition on return. Passed in value unused.

void DataAccessServices.WebServices.Reports.CreateReportDefinitions (ReportDefinitionList* reportDefinitionList*, Boolean* compress*) `[inline]`

Creates the supplied report definitions within the database.

Parameters:

reportDefinitionList	A list of report definitions to add.
compress	Whether or not the report definitions are compressed.

void DataAccessServices.WebServices.Reports.DeleteReportDefinition (Guid* reportDefinitionKey, *DateTime?* modifiedTime*) `[inline]`

Deletes a report definition from the database.

Parameters:

reportDefinitionKey	Key of the report to delete.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

void DataAccessServices.WebServices.Reports.DeleteReportDefinitions (ReportDefinitionList* reportDefinitionList*) `[inline]`

Deletes the supplied report definitions from the database.

Parameters:

reportDefinitionList	The report definition list.

ReportDefinitionsDataSet DataAccessServices.WebServices.Reports.GetReportDefinition (Guid* reportDefinitionKey*) `[inline]`

Returns a specific report definition from the database.

Parameters:

reportDefinitionKey	The report definition key.

Returns:

Data set containing specific report definition from the database.

ReportDefinitionsDataSet DataAccessServices.WebServices.Reports.GetReportDefinitions () `[inline]`

Returns a list of all report definitions stored within the database.

Returns:

Data set containing all report definitions stored within the database.

ReportDefinitionsDataSet DataAccessServices.WebServices.Reports.GetVisibleReportDefinitions (Boolean* visible*) `[inline]`

Returns report definitions filtered on the visible property.

Parameters:

visible	If set to true returns visible reports.

Returns:

Data set containing report definitions filtered on the visible property.

void DataAccessServices.WebServices.Reports.UpdateReportDefinition (Guid* reportDefinitionKey*, String* name*, String* description*, String* category*, String* type*, Guid* productKey*, Boolean* visible*, Guid?* policyKey*, String* ownerSid*, DateTime* modifiedTime*) `[inline]`

Updates the report definition.

Parameters:

reportDefinitionKey	The report definition key.
name	Name of the specified item.
description	Description of the specified item.
category	The category.
type	The type.
productKey	GUID specifying a key for the associated product.

visible	If set to true report is visible.
policyKey	The policy key.
ownerSid	The owner sid.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

void DataAccessServices.WebServices.Reports.UpdateReportDefinitionSecurity (Guid* reportDefinitionKey*, Guid?* policyKey*, String* ownerSid*, ref DateTime* modifiedTime*) `[inline]`

Updates the report definition security.

Parameters:

reportDefinitionKey	The report definition key.
policyKey	The policy key.
ownerSid	The owner sid.
modifiedTime	DateTime specifying when this item was last modified by the application - used for concurrency purposes.

The documentation for this class was generated from the following file:

- Reports.cs

DataAccessServices.WebServices.Security Class Reference

Within the Management Center it is possible to configure permissions for the different object types; these object types include groups, packages and alerts rules etc. These objects contain an OwnerSid and PolicyFK columns which control the permissions on the objects.

Public Member Functions

- void ApplySecurityElementChanges (ref SecurityElementsDataSet changes)

  Apply changes made to a users data set into the database.

- void ApplySecurityRoleChanges (ref SecurityRolesDataSet changes)

  Apply changes made to a users data set into the database.

- void ApplyUserChanges (ref UsersDataSet userChanges)

  Apply changes made to the users data set.

- bool CanLogin ()

  Determine if the current user can login to the database.

- bool CanLoginEx (out string userName, out string userSid)

  Determine if the current user can login to the database.

- void CreateObjectSecurityRole (Guid securityRoleKey, string name, string description, ObjectPermissions mask, out DateTime modifiedTime, bool readOnly)

  Create an Object security role.

- void CreatePolicy (Guid policyKey, PolicyType type)

  Create a new policy.

- void CreateSecurityElement (Guid securityElementKey, Guid policyKey, ElementType elementType, Guid securityRoleKey, string userSid, out DateTime modifiedTime)

  Create a security element.

- void CreateServerSecurityRole (Guid securityRoleKey, string name, string description, ServerPermissions mask, out DateTime modifiedTime, bool readOnly)

  Create a Server security role.

- void CreateUser (Guid userKey, string name, string sid, bool isGroup, out DateTime modifiedTime)

  Create a new user.

- void DeletePolicy (Guid policyKey)

  Delete a policy.

- void DeleteSecurityElement (Guid securityElementKey, DateTime modifiedTime)

  Delete a security element.

- void DeleteSecurityRole (Guid securityRoleKey, DateTime modifiedTime)

Delete a security role.

- void DeleteUser (Guid userKey, DateTime? modifiedTime)

  Delete an existing user.

- PoliciesDataSet GetPolicies ()

  Get all policy keys.

- SecurityElementsDataSet GetSecurityElementFromKey (Guid securityElementKey)

  Get a specific security element.

- SecurityElementsDataSet GetSecurityElements ()

  Get all security elements.

- SecurityElementsDataSet GetSecurityElementsFromPolicy (Guid policyKey)

  Get security elements with the specific policy key.

- SecurityRolesDataSet GetSecurityRoleFromKey (Guid securityRoleKey)

  Get a specific security role from the database.

- SecurityRolesDataSet GetSecurityRoles (bool withPermissions)

  Get all security roles.

- SecurityRolesDataSet GetSecurityRolesFromType (RoleType roleType)

  Get security roles with a specified type.

- ServerPermissions GetServerPermissions ()

  Get all server permissions for the current user.

- string GetUserName ()

  Get the current user name.

- UsersDataSet GetUsers ()

  Get all users.

- void UpdateSecurityRole (Guid securityRoleKey, string name, string description, ObjectPermissions mask, ref DateTime modifiedTime, bool readOnly)

  Update an existing security role.

- void UpdateUser (Guid userKey, string name, string sid, ref DateTime modifiedTime)

  Update information about a user.

Detailed Description

Within the Management Center it is possible to configure permissions for the different object types; these object types include groups, packages and alerts rules etc. These objects contain an OwnerSid and PolicyFK columns which control the permissions on the objects.

Member Function Documentation

***void DataAccessServices.WebServices.Security.ApplySecurityElementChanges (ref SecurityElementsDataSet* changes)** `[inline]`*

Apply changes made to a users data set into the database.

Parameters:

changes	The data set consisting of the changes.

***void DataAccessServices.WebServices.Security.ApplySecurityRoleChanges (ref SecurityRolesDataSet* changes)** `[inline]`*

Apply changes made to a users data set into the database.

Parameters:

changes	The data set consisting of the changes.

***void DataAccessServices.WebServices.Security.ApplyUserChanges (ref UsersDataSet* userChanges)** `[inline]`*

Apply changes made to the users data set.

Parameters:

userChanges	The data set consisting of the changes.

bool DataAccessServices.WebServices.Security.CanLogin () `[inline]`

Determine if the current user can login to the database.

Returns:

True if the user can login, otherwise False.

bool DataAccessServices.WebServices.Security.CanLoginEx (out string **userName*****, out string*** **userSid*****)*** `[inline]`

Determine if the current user can login to the database.

Parameters:

userSid	Returns the security identifier of the current user.

Returns:

True if the user can login, otherwise False.

void DataAccessServices.WebServices.Security.CreateObjectSecurityRole (Guid **securityRoleKey*****, string*** **name*****, string*** **description*****, ObjectPermissions*** **mask*****, out DateTime*** **modifiedTime*****, bool*** **readOnly*****)*** `[inline]`

Create an Object security role.

Parameters:

securityRoleKey	The key for the role.
name	The name of the role.
description	The description for the role.
mask	The permissions for the role.
modifiedTime	Returns the modified time of the role.
readOnly	Whether the role is read only.

***void DataAccessServices.WebServices.Security.CreatePolicy (Guid* policyKey*, PolicyType* type*)* `[inline]`**

Create a new policy.

Parameters:

policyKey	The key for the new policy.
type	The type of the policy to create.

***void DataAccessServices.WebServices.Security.CreateSecurityElement (Guid* securityElementKey*, Guid* policyKey*, ElementType* elementType*, Guid* securityRoleKey*, string* userSid*, out DateTime* modifiedTime*)* `[inline]`**

Create a security element.

Parameters:

securityElementKey	The key for the new element.
policyKey	The policy key associated with this element.
elementType	The element type.
securityRoleKey	The security role associated with this element.
userSid	The security identifier of the user associated with this element.
modifiedTime	Returns the modified time of the new element.

***void DataAccessServices.WebServices.Security.CreateServerSecurityRole (Guid* securityRoleKey*, string* name*, string* description*, ServerPermissions* mask*, out DateTime* modifiedTime*, bool* readOnly*)* `[inline]`**

Create a Server security role.

Parameters:

securityRoleKey	The key for the role.
name	The name of the role.

description	The description for the role.
mask	The permissions for the role.
modifiedTime	Returns the modified time of the role.
readOnly	Whether the role is read only.

void DataAccessServices.WebServices.Security.CreateUser (Guid* userKey*, string* name*, string* sid*, bool* isGroup*, out DateTime* modifiedTime*) `[inline]`

Create a new user.

Parameters:

userKey	The key which identifies the user.
name	The name of the user.
sid	The security identifier of the user.
isGroup	Whether the user is a Windows group.
modifiedTime	Returns the modified time of the new record.

void DataAccessServices.WebServices.Security.DeletePolicy (Guid* policyKey*) `[inline]`

Delete a policy.

Parameters:

policyKey	The key of the policy to delete.

void DataAccessServices.WebServices.Security.DeleteSecurityElement (Guid* securityElementKey, *DateTime* modifiedTime*) `[inline]`

Delete a security element.

Parameters:

securityElementKey	The key of the element to delete.
modifiedTime	Modified time of the element.

void DataAccessServices.WebServices.Security.DeleteSecurityRole (Guid* securityRoleKey, *DateTime* modifiedTime*) `[inline]`

Delete a security role.

Parameters:

securityRoleKey	The key of the role to delete.
modifiedTime	Modified time of the role.

void DataAccessServices.WebServices.Security.DeleteUser (Guid* userKey, *DateTime?* modifiedTime*) `[inline]`

Delete an existing user.

Parameters:

userKey	The key which identifies the user.
modifiedTime	Modified time of the user to delete, or Null.

PoliciesDataSet DataAccessServices.WebServices.Security.GetPolicies () `[inline]`

Get all policy keys.

Returns:

A data set containing all policy keys.

***SecurityElementsDataSet DataAccessServices.WebServices.Security.GetSecurityElementFromKey (Guid* securityElementKey*)* `[inline]`**

Get a specific security element.

Parameters:

securityElementKey	The key of the element.

Returns:

A data set containing the specific element, or an empty data set if the key does not exist.

***SecurityElementsDataSet DataAccessServices.WebServices.Security.GetSecurityElements ()* `[inline]`**

Get all security elements.

Returns:

A data set containing all security elements.

***SecurityElementsDataSet DataAccessServices.WebServices.Security.GetSecurityElementsFromPolicy (Guid* policyKey*)* `[inline]`**

Get security elements with the specific policy key.

Parameters:

policyKey	The policy key to find.

Returns:

A data set containing matching security elements, or an empty data set if there are no matches.

SecurityRolesDataSet DataAccessServices.WebServices.Security.GetSecurityRoleFromKey (Guid* securityRoleKey*) `[inline]`

Get a specific security role from the database.

Parameters:

securityRoleKey	The key of the security role.

Returns:

A data set containing information for the specified security role, or an empty data set if there are no matches.

SecurityRolesDataSet DataAccessServices.WebServices.Security.GetSecurityRoles (bool* withPermissions*) `[inline]`

Get all security roles.

Parameters:

withPermissions	Whether to include the Server and Object permissions for each role.

Returns:

A data set containing information for the security roles or an empty data set if none exist.

SecurityRolesDataSet DataAccessServices.WebServices.Security.GetSecurityRolesFromType (RoleType* roleType*) `[inline]`

Get security roles with a specified type.

Parameters:

roleType	The role type to retrieve. This is either Server or Object.

Returns:

A data set containing information for the security roles of the specified type or an empty data set if none exist.

ServerPermissions DataAccessServices.WebServices.Security.GetServerPermissions () `[inline]`

Get all server permissions for the current user.

Returns:

The server permissions for the current user.

string DataAccessServices.WebServices.Security.GetUserName () `[inline]`

Get the current user name.

Returns:

The current user name.

UsersDataSet DataAccessServices.WebServices.Security.GetUsers () `[inline]`

Get all users.

Returns:

A data set containing all users.

void DataAccessServices.WebServices.Security.UpdateSecurityRole (Guid* securityRoleKey*, string* name*, string* description*, ObjectPermissions* mask*, ref DateTime* modifiedTime*, bool* readOnly*) ***`[inline]`***

Update an existing security role.

Parameters:

securityRoleKey	The key of the role to update.
name	The name of the role.
description	The description for the role.
mask	The permissions mask for the role.
modifiedTime	The modified time of the record to update. Returns the modified time of the updated record.
readOnly	Whether the role is read only.

void DataAccessServices.WebServices.Security.UpdateUser (Guid* userKey*, string* name*, string* sid*, ref DateTime* modifiedTime*) ***`[inline]`***

Update information about a user.

Parameters:

userKey	The key which identifies the user.
name	The name of the user.
sid	The security identifier of the user.
modifiedTime	The modified time of the record to update. Returns the modified time of the updated record.

The documentation for this class was generated from the following file:

- Security.cs

DataAccessServices.WebServices.Servers Class Reference

Within the Management Center database it is possible to configure a number of fail over servers which are used if the current management server cannot be contacted. The user can configure a number of fail over servers on a per group basis allowing different groups to be serviced by different management servers.

Public Member Functions

- void ApplyChanges (ref ServersDataSet serverChanges)

  Update the database with the changes in the data set.

- void CreateServer (Guid serverKey, Guid? groupKey, String url, Int32 index, Boolean performDiagnostics, Boolean disabled, out DateTime modifiedTime)

  Create a new server.

- void DeleteServer (Guid serverKey, DateTime? modifiedTime)

  Delete a server.

- ServersDataSet GetServers (Boolean includeDisabled)

  Get all servers.

- ServersDataSet GetServersFromGroupKey (Guid? groupKey, Boolean includeDisabled)

  Get all servers for a group.

- void UpdateServer (Guid serverKey, Guid? groupKey, String url, Int32 index, Boolean performDiagnostics, Boolean disabled, ref DateTime modifiedTime)

  Update an existing server.

Detailed Description

Within the Management Center database it is possible to configure a number of fail over servers which are used if the current management server cannot be contacted. The user can configure a number of fail over servers on a per group basis allowing different groups to be serviced by different management servers.

Member Function Documentation

void DataAccessServices.WebServices.Servers.ApplyChanges (ref ServersDataSet* serverChanges*) `[inline]`

Update the database with the changes in the data set.

Parameters:

serverChanges	A data set containing the server changes to apply.

void DataAccessServices.WebServices.Servers.CreateServer (Guid* serverKey*, Guid?* groupKey*, String* url*, Int32* index*, Boolean* performDiagnostics*, Boolean* disabled*, out DateTime* modifiedTime*) ***`[inline]`***

Create a new server.

Parameters:

serverKey	The key of the new server.
groupKey	The group key this server will be added to.
url	The URL of the server.
index	The index of the server.
performDiagnostics	Whether the group can perform diagnostics.
disabled	Whether the group is disabled.
modifiedTime	Returns the modified time of the new server.

void DataAccessServices.WebServices.Servers.DeleteServer (Guid* serverKey*, DateTime?* modifiedTime*) ***`[inline]`***

Delete a server.

Parameters:

serverKey	The key of the server to delete.
modifiedTime	Modified time of the server to delete.

***ServersDataSet DataAccessServices.WebServices.Servers.GetServers (Boolean* includeDisabled*)* `[inline]`**

Get all servers.

Parameters:

includeDisabled	Whether to include disabled servers.

Returns:

A data set containing the servers.

***ServersDataSet DataAccessServices.WebServices.Servers.GetServersFromGroupKey (Guid?* groupKey*, Boolean* includeDisabled*)* `[inline]`**

Get all servers for a group.

Parameters:

groupKey	The group key to retrieve.
includeDisabled	Whether to include disabled servers.

Returns:

A data set containing the servers or an empty data set if none exist.

void DataAccessServices.WebServices.Servers.UpdateServer (Guid* serverKey*, Guid?* groupKey*, String* url*, Int32* index*, Boolean* performDiagnostics*, Boolean* disabled*, ref DateTime* modifiedTime*) `[inline]`

Update an existing server.

Parameters:

serverKey	The key of the server to update.
groupKey	The group key for the server.
url	The URL for the server.
index	The index of the server.
performDiagnostics	Whether the server should be enabled or disabled for diagnostics.
disabled	Whether the server should be enabled or disabled.
modifiedTime	The modified time of the server. Returns the new modified time of the server.

The documentation for this class was generated from the following file:

- Servers.cs