# Table of Contents

Table of contents ........................................................................................................... 2
Disclaimer ............................................................................................................................ 3
Introduction ......................................................................................................................... 4
  Management Console ...................................................................................................... 4
  Proxy DLL ......................................................................................................................... 4
  Authorization & Security .................................................................................................. 4
  Limitations ....................................................................................................................... 4
How to use the Data Access WebServices - Examples ..................................................... 5
DataAccess Endpoint Tasks .............................................................................................. 5
  Connecting to the Management Server ......................................................................... 5
  Creating & Managing Deployment Groups ...................................................................... 6
  Managing Machines ........................................................................................................ 6
  Retrieving Alerts ............................................................................................................ 6
DataRows - Alerts ............................................................................................................. 7
  Alerts ............................................................................................................................... 7
    AlertsRow .................................................................................................................... 7
    AlertRulesRow ............................................................................................................. 7
    ActionsRow .................................................................................................................. 8
    ConfigurationsRow ..................................................................................................... 8
DataRows - Conditions .................................................................................................... 9
  Conditions ...................................................................................................................... 9
    ConditionsRow ............................................................................................................ 9
DataRows - Database ....................................................................................................... 10
  Database ....................................................................................................................... 10
    NamedValuesRow ..................................................................................................... 10
DataRows - Deployment .................................................................................................. 11
  Deployment ................................................................................................................... 11
    CredentialsRow .......................................................................................................... 11
    InstructionsRow ........................................................................................................... 11
    StatusHistoryRow ...................................................................................................... 11
DataRows - DiscoveredMachines .................................................................................... 13
  DiscoveredMachines ..................................................................................................... 13
    DiscoveredMachinesRow .......................................................................................... 13
    SchedulerConfigurationDto ....................................................................................... 13
DataRows - Events .......................................................................................................... 14
  Events ........................................................................................................................... 14
    EventRow .................................................................................................................... 14
    ParamRow .................................................................................................................. 14
    EventDefinitionRow .................................................................................................. 14
    EventDefinitionParamRow ......................................................................................... 15
    EventParameterValuesRow ......................................................................................... 15
DataRows - Groups ......................................................................................................... 16
  Groups ........................................................................................................................... 16
    ScheduleRow .............................................................................................................. 16
    GroupsRow .................................................................................................................. 18
    GroupPackagesRow .................................................................................................... 19
    EventFilterRow ......................................................................................................... 19
    EventFilterRow ......................................................................................................... 20
    StatisticsDto ............................................................................................................... 20
DataRows - Licenses ....................................................................................................... 21
  Licenses ......................................................................................................................... 21
    LicensesRow ............................................................................................................... 21
<table>
<thead>
<tr>
<th>Module</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>174</td>
</tr>
</tbody>
</table>
# Table of contents

1. [Disclaimer](#)
2. [Introduction](#)
3. [Examples - How To](#)
4. [DataRows - Alerts](#)
5. [DataRows - Conditions](#)
6. [DataRows - Database](#)
7. [DataRows - Deployment](#)
8. [DataRows - DiscoveredMachines](#)
9. [DataRows - Events](#)
10. [DataRows - Groups](#)
11. [DataRows - Licenses](#)
12. [DataRows - Machines](#)
13. [DataRows - Packages](#)
14. [DataRows - Products](#)
15. [DataRows - Reports](#)
16. [DataRows - Security](#)
17. [DataRows - Servers](#)
Disclaimer
Copyright © 2020, Ivanti. All rights reserved.
Introduction

This document details the web services interface exposed by the AppSense Management Server. While primarily used by the AppSense Management Console this interface is available for others to use.

Caution

AppSense 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 AppSense 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

The Management Console product ships with a web services proxy DLL used by the console for communication with the server. Third party tools may use this interface for convenience, or may access the APIs directly via the web services described in this document. The following DLLs are provided:

- DataAccess Endpoints: [InstallDir]\Management Center\Console\ManagementConsole.WebServices.dll
- PackageManagement Endpoint: [InstallDir]\Management Center\Console\PackageManager.dll (this also requires CommonDialogs.dll)

Caution

The AppSense Management Console must be installed before using the Proxy DLLs directly to ensure that any dependant DLLs are also available. This step is not required if the APIs are accessed directly via the web services

Authorization & Security

A full set of credentials should be specified when using the Proxy DLL by passing an instance of NetworkCredential. An example of this can be found under “Connecting to the Management Server”

Limitations

Some types defined in the proxy DLL 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 WebServices - Examples

DataAccess Endpoint Tasks

Connecting to the Management Server

Firstly, ensure that the latest version of the Management Console is installed. Before using the API a connection to the Management Server must be established. This involves:

• Loading the proxy DLL
• Instantiating a NetworkCredential instance
• Call the Connect method of ManagementConsole.WebServices

To logon as the current user:

```powershell
# Load proxy DLL
Add-Type -Path "${Env:ProgramFiles}\AppSense\ManagementCenter\Console\ManagementConsole.WebServices.dll"

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

# Get NetworkCredential instance
$credentials = [System.Net.CredentialCache]::DefaultCredentials
$credential = $credentials.GetCredential($url, "Basic")

# Create connection to the Management Server
[ManagementConsole.WebServices]::Connect($url, $credential)
```

To specify a user to logon as:

```powershell
# Load proxy DLL
Add-Type -Path "${Env:ProgramFiles}\AppSense\ManagementCenter\Console\ManagementConsole.WebServices.dll"

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

# Get NetworkCredential instance
$credential = New-Object System.Net.NetworkCredential -Property
Username = "MyUsername"; Password = "MyPassword"; Domain = "MyDomain";

# Create connection to the Management Server
[ManagementConsole.WebServices]::Connect($url, $credential)
```
Creating & Managing Deployment Groups

Groups are managed using the GroupsWebService. For convenience when using the Proxy DLL a reference to this web service is available from ManagementConsole.WebServices.Groups once a connection is established.

```powershell
# Get GroupsWebService reference
$GroupsWebService = [ManagementConsole.WebServices]::Groups

# 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. For convenience when using the Proxy DLL a reference to this web service is available from ManagementConsole.WebServices.DiscoveredMachines and ManagementConsole.WebServices.Machines.

```powershell
# Get DiscoveredMachinesWebService reference
$DiscoveredMachinesWebService = [ManagementConsole.WebServices]::DiscoveredMachines

# Get MachinesWebService reference
$MachinesWebService = [ManagementConsole.WebServices]::Machines

# Set up a Boolean to get the machines with or without summary
# machines with a summary detail the number of alerts counted
$withSummary = $true

# Get list of machines
$MachinesDataSet = $MachinesWebService.GetMachines($withSummary)
$Machines = $MachinesDataSet.Machines
```

Retrieving Alerts

Alerts are managed using the AlertWebService. For convenience when using the Proxy DLL a reference to this web service is available from ManagementConsole.WebServices.Alerts once a connection is established.

```powershell
# Get AlertsWebService reference
$AlertsWebService = [ManagementConsole.WebServices]::Alerts

# Get list of alerts
$AlertsDataSet = $AlertsWebService.GetAlerts()
$Alerts = $AlertsDataSet.Alerts
```
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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlertKey</td>
<td>Int32</td>
<td>Unique identifier relating to alert</td>
</tr>
<tr>
<td>AlertRuleKey</td>
<td>Guid</td>
<td>Unique identifier relating to alert rule</td>
</tr>
<tr>
<td>AlertRuleName</td>
<td>String</td>
<td>Name of alert rule</td>
</tr>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>GroupName</td>
<td>String</td>
<td>Name of group</td>
</tr>
<tr>
<td>MachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to machine</td>
</tr>
<tr>
<td>MachineName</td>
<td>String</td>
<td>Name of machine</td>
</tr>
<tr>
<td>Status</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>MillisecondsOfDay</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Severity</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlertRuleKey</td>
<td>Guid</td>
<td>Unique identifier relating to alert rule</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of alert rule</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of alert rule</td>
</tr>
<tr>
<td>Severity</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Enabled</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>EventQuery</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>MachineQuery</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>UserQuery</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td>DataType</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>GroupQuery</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>PolicyKey</td>
<td>Guid</td>
<td>Unique identifier relating to policy</td>
</tr>
<tr>
<td>OwnerSid</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>SecurityDescriptor</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Int32</td>
<td></td>
</tr>
</tbody>
</table>

**ActionsRow**

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionKey</td>
<td>Guid</td>
<td>Unique identifier relating to action</td>
</tr>
<tr>
<td>AlertRuleKey</td>
<td>Guid</td>
<td>Unique identifier relating to alert rule</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of action</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of action</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>

**ConfigurationsRow**

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionKey</td>
<td>Guid</td>
<td>Unique identifier relating to action</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of configuration</td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>ActionsRow</td>
<td>ActionsRow</td>
<td></td>
</tr>
</tbody>
</table>
DataRows - Conditions

Conditions

ConditionsRow

Represents a condition on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionPK</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>ConditionType</td>
<td>Byte</td>
<td>The values of which are: 0 – NetBios 1 – Container 2 – ComputerName 3 – Domain 4 – All; used for the default group only</td>
</tr>
<tr>
<td>IsInclude</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>IncludeChildren</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>ADOBJECTGUID</td>
<td>Guid</td>
<td></td>
</tr>
<tr>
<td>ADOBJECTDISTINGUISHEDNAME</td>
<td>String</td>
<td>Name of ad object distinguished</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>Filter</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
DataRows - Database

Database

NamedValuesRow

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of named value</td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
DataRows - Deployment

Deployment

CredentialsRow

Represents a credential on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserName</td>
<td>String</td>
<td>Name of user</td>
</tr>
<tr>
<td>Password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>CredentialsKey</td>
<td>Guid</td>
<td>Unique identifier relating to credentials</td>
</tr>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>

InstructionsRow

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommandID</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>Plugin</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>MachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to machine</td>
</tr>
<tr>
<td>InstructionKey</td>
<td>Guid</td>
<td>Unique identifier relating to instruction</td>
</tr>
<tr>
<td>Settings</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

StatusHistoryRow

Represents a status history on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstructionKey</td>
<td>Guid</td>
<td>Unique identifier relating to instruction</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>InstructionsRow</td>
<td>InstructionsRow</td>
<td></td>
</tr>
</tbody>
</table>
DataRows - DiscoveredMachines

DiscoveredMachines

DiscoveredMachinesRow
Represents a discovered machine on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscoveredMachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to discovered machine</td>
</tr>
<tr>
<td>ExpectedGroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to expected group</td>
</tr>
<tr>
<td>NetBiosName</td>
<td>String</td>
<td>Name of net bios</td>
</tr>
<tr>
<td>ADOBJECTGUID</td>
<td>Guid</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of discovered machine</td>
</tr>
<tr>
<td>OperatingSystem</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>DNS</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>UserSpecified</td>
<td>Bit</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedDiscoveryTime</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>ModifiedDeploymentTime</td>
<td>DateTime</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TaskId</td>
<td>Guid</td>
<td>Unique identifier of a scheduler service</td>
</tr>
<tr>
<td>AssemblyName</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>TypeName</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>DefaultInterval</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ScheduledInterval</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>EventName</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>StartMode</td>
<td>Int32</td>
<td>1 = automatic, 0 = manual</td>
</tr>
<tr>
<td>LastStartMode</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Timeout</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ActiveServer</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>LastStarted</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>LastUpdated</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>LastCompleted</td>
<td>DateTime</td>
<td></td>
</tr>
</tbody>
</table>
# DataRows - Events

## Events

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventKey</td>
<td>Int32</td>
<td>Unique identifier relating to event</td>
</tr>
<tr>
<td>EventDefinitionKey</td>
<td>Int32</td>
<td>Unique identifier relating to event definition</td>
</tr>
<tr>
<td>MachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to machine</td>
</tr>
<tr>
<td>MachineName</td>
<td>String</td>
<td>Name of machine</td>
</tr>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>GroupName</td>
<td>String</td>
<td>Name of group</td>
</tr>
<tr>
<td>UserName</td>
<td>String</td>
<td>Name of user</td>
</tr>
<tr>
<td>Time</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>MillisecondsOfDay</td>
<td>Int32</td>
<td></td>
</tr>
</tbody>
</table>

**ParamRow**
Represents a param on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventDefinitionParamKey</td>
<td>Int32</td>
<td>Unique identifier relating to event definition param</td>
</tr>
<tr>
<td>EventKey</td>
<td>Int32</td>
<td>Unique identifier relating to event</td>
</tr>
<tr>
<td>Type</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of param</td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>EventRow</td>
<td>EventRow</td>
<td></td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventDefinitionKey</td>
<td>Int32</td>
<td>Unique identifier relating to event definition</td>
</tr>
<tr>
<td>Column</td>
<td>DataType</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of event definition</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of event definition</td>
</tr>
<tr>
<td>EventDescription</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ProductKey</td>
<td>Guid</td>
<td>Unique identifier relating to product</td>
</tr>
<tr>
<td>ProductName</td>
<td>String</td>
<td>Name of product</td>
</tr>
<tr>
<td>Type</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>HighVolume</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>DefaultEnabledState</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventDefinitionParamKey</td>
<td>Int32</td>
<td>Unique identifier relating to event param</td>
</tr>
<tr>
<td>EventDefinitionKey</td>
<td>Int32</td>
<td>Unique identifier relating to event</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of event param</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of event param</td>
</tr>
<tr>
<td>Type</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>EventDefinitionParamKey</td>
<td>EventDefinitionRow</td>
<td></td>
</tr>
</tbody>
</table>

**EventParameterValuesRow**

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StringValue</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>IntegerValue</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>DateTimeValue</td>
<td>DateTime</td>
<td></td>
</tr>
</tbody>
</table>
## 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:

<table>
<thead>
<tr>
<th>Flags</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>9007199254740992</td>
<td>10000000000000000000000000000000000000000000</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Flags</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00</td>
<td>00:30</td>
<td>9007199254740993</td>
<td>10000000000000000000000000000000000000000000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Flags</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00</td>
<td>00:30</td>
<td>9007199254740993</td>
<td>10000000000000000000000000000000000000000000</td>
</tr>
<tr>
<td>00:00</td>
<td>01:30</td>
<td>9007199254740999</td>
<td>10000000000000000000000000000000000000000000000000001</td>
</tr>
<tr>
<td>00:00</td>
<td>02:00</td>
<td>9007199254741007</td>
<td>1000000000000000000000000000000000000000000000000000111</td>
</tr>
<tr>
<td>00:00</td>
<td>02:00</td>
<td>15 (disabled)</td>
<td>0000000000000000000000000000000000000000000000000000000111</td>
</tr>
</tbody>
</table>

16
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.

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Flags</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>18:00</td>
<td>08:00</td>
<td>9288605512040447</td>
<td>1000000111111111</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>110000000000000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0000001111111111111111</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>MondayFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>TuesdayFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>WednesdayFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>ThursdayFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>FridayFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>SaturdayFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>SundayFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>RetryPeriodSeconds</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>MondayConfigFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>TuesdayConfigFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>WednesdayConfigFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>ThursdayConfigFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>FridayConfigFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>SaturdayConfigFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>SundayConfigFlags</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>ConfigFlags</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>PostponeInstallationInSche</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>dule</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PostponeInstallationLimit</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>DownloadConfigurationsOnSta</td>
<td>Boolean</td>
<td>Determines whether startup should be blocked whi</td>
</tr>
<tr>
<td>rtup</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
the CCA downloads new configurations and installs them before allowing the user to login

**GroupsRow**

Represents a group on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataTypes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>Guid</td>
<td>The key which identifies this group.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the group.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>Description of the group.</td>
</tr>
<tr>
<td>pollPeriodSeconds</td>
<td>Int32</td>
<td>The amount of seconds between polls.</td>
</tr>
<tr>
<td>uploadPollPeriodSeconds</td>
<td>Int32</td>
<td>The amount of seconds between uploads.</td>
</tr>
<tr>
<td>eventLogEnabled</td>
<td>Boolean</td>
<td>Whether to log events to the event log.</td>
</tr>
<tr>
<td>fileLogEnabled</td>
<td>Boolean</td>
<td>Whether to log events to the file log.</td>
</tr>
<tr>
<td>fileLogFilename</td>
<td>String</td>
<td>The file name to log events to.</td>
</tr>
<tr>
<td>anonymousUserLogging</td>
<td>Boolean</td>
<td>Whether events from machines within this group will be logged with anonymous users.</td>
</tr>
<tr>
<td>anonymousMachineLogging</td>
<td>Boolean</td>
<td>Whether events from machines within this group will be logged with anonymous machines.</td>
</tr>
<tr>
<td>overrideServerUrls</td>
<td>Boolean</td>
<td>Whether this group overrides server URL’s.</td>
</tr>
<tr>
<td>selfRegistrationEnabled</td>
<td>Boolean</td>
<td>Whether or not self registration is allowed.</td>
</tr>
<tr>
<td>selfUnregistrationEnabled</td>
<td>Boolean</td>
<td>Whether or not self unregistration is allowed.</td>
</tr>
<tr>
<td>selfUpdateEnabled</td>
<td>Boolean</td>
<td>Whether or not self update of agents and configurations is allowed.</td>
</tr>
<tr>
<td>priority</td>
<td>Int32</td>
<td>Order in which this group’s membership rules will be evaluated.</td>
</tr>
<tr>
<td>pollPeriodVariationSeconds</td>
<td>Int32</td>
<td>The VariationSeconds allowed for each poll</td>
</tr>
<tr>
<td>uploadPollPeriodVariationSeconds</td>
<td>Int32</td>
<td>The VariationSeconds allowed for each upload.</td>
</tr>
<tr>
<td>nativeConfigurations</td>
<td>Boolean</td>
<td>Set to true whenever the CCA is deploying configurations as</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>PackageKey</td>
<td>Guid</td>
<td>Unique identifier relating to package</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of group package</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Company</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ProductName</td>
<td>String</td>
<td>Name of product</td>
</tr>
<tr>
<td>Major</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Minor</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Build</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Revision</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Exists</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>EventDefinitionKey</td>
<td>Int32</td>
<td>Unique identifier relating to event definition</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of event filter</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of event filter</td>
</tr>
<tr>
<td>ProductName</td>
<td>String</td>
<td>Name of product</td>
</tr>
<tr>
<td>Enabled</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>HighVolume</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>DefaultEnabledState</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>
**EventFilterRow**

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of named value</td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

**StatisticsDto**

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>String</td>
<td>Total number of groups</td>
</tr>
<tr>
<td>GroupsDeployed</td>
<td>String</td>
<td>Number of groups that are fully deployed</td>
</tr>
<tr>
<td>GroupsInError</td>
<td>String</td>
<td>Number of groups containing computers in an error state</td>
</tr>
<tr>
<td>Computers</td>
<td>String</td>
<td>Total number of computers</td>
</tr>
<tr>
<td>ComputersDeployed</td>
<td>String</td>
<td>Number of computers that are fully deployed</td>
</tr>
<tr>
<td>ComputersInError</td>
<td>String</td>
<td>Number of computers that are in an error state</td>
</tr>
<tr>
<td>ComputersOffline</td>
<td>String</td>
<td>Number of computers that are not polling in</td>
</tr>
<tr>
<td>Alerts</td>
<td>String</td>
<td>Number of recorded alerts</td>
</tr>
<tr>
<td>AlertsCritical</td>
<td>String</td>
<td>Number of recorded critical alerts</td>
</tr>
<tr>
<td>AlertsNew</td>
<td>String</td>
<td>Number of new alerts</td>
</tr>
<tr>
<td>AlertsNew24H</td>
<td>String</td>
<td>Number of alerts that have been recorded in the last 24 hours</td>
</tr>
</tbody>
</table>
# DataRows - Licenses

## Licenses

**LicensesRow**

Represents a license on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LicenseCode</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ActivationCode</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>LicenseType</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ProductKey</td>
<td>Guid</td>
<td>Unique identifier relating to product</td>
</tr>
<tr>
<td>LicenseCount</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ExpiryDate</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>BaseLicense</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>GroupKey</td>
<td>String</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>LicenseKey</td>
<td>Guid</td>
<td>Unique identifier relating to license</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to machine</td>
</tr>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>GroupName</td>
<td>String</td>
<td>Name of group</td>
</tr>
<tr>
<td>Platform</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>NetBiosName</td>
<td>String</td>
<td>Name of net bios</td>
</tr>
<tr>
<td>DistinguishedName</td>
<td>String</td>
<td>Name of distinguished</td>
</tr>
<tr>
<td>OldDistinguishedName</td>
<td>String</td>
<td>Name of old distinguished</td>
</tr>
<tr>
<td>ObjectGuid</td>
<td>Guid</td>
<td></td>
</tr>
<tr>
<td>LastPollTime</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>LastPollStatus</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>LastUploadTime</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>LastUploadStatus</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>IsPendingDeletion</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>AlertCount</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>ModifiedGroupTime</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>DiagnosticsError</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>DiagnosticsState</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>DiagnosticsTime</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>Deployed</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>DeployError</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>Offline</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>DNS</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>LastResponseSeconds</td>
<td>Int32</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to machine</td>
</tr>
<tr>
<td>PackageKey</td>
<td>Guid</td>
<td>Unique identifier relating to package</td>
</tr>
<tr>
<td>Major</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Minor</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Build</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Revision</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of machine package</td>
</tr>
<tr>
<td>Company</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ProductName</td>
<td>String</td>
<td>Name of product</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>Status</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>StatusMessage</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ChildStatus</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ChildMajor</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ChildMinor</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ChildBuild</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ChildRevision</td>
<td>Int32</td>
<td></td>
</tr>
</tbody>
</table>

**MatchResultsRow**

Represents a match result on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>GroupName</td>
<td>String</td>
<td>Name of group</td>
</tr>
<tr>
<td>MatchName</td>
<td>String</td>
<td>Name of match</td>
</tr>
<tr>
<td>Difference</td>
<td>Int32</td>
<td></td>
</tr>
</tbody>
</table>
### MachineDetailsRow

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to machine</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of machine detail</td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### MachineDiagnosticsRow

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MachineKey</td>
<td>Guid</td>
<td>Unique identifier relating to machine</td>
</tr>
<tr>
<td>ServerUrl</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of machine diagnostic</td>
</tr>
<tr>
<td>Error</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LatestName</td>
<td>String</td>
<td>Name of latest</td>
</tr>
<tr>
<td>PackageKey</td>
<td>Guid</td>
<td>UpgradeCode property defined in associated MSI files</td>
</tr>
<tr>
<td>Company</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ProductKey</td>
<td>Guid</td>
<td>Unique identifier relating to product</td>
</tr>
<tr>
<td>ProductName</td>
<td>String</td>
<td>Name of product</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>OwnerSid</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>PolicyKey</td>
<td>Guid</td>
<td>Unique identifier relating to policy</td>
</tr>
<tr>
<td>Locked</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>LockedUserName</td>
<td>String</td>
<td>Name of locked user</td>
</tr>
<tr>
<td>LatestMajor</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>LatestMinor</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>LatestBuild</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>LatestRevision</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>SecurityDescriptor</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Dirty</td>
<td>Byte</td>
<td></td>
</tr>
</tbody>
</table>

Package Versions Row
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.
<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PackageVersionKey</td>
<td>Guid</td>
<td>ProductCode property defined in associated MSI files</td>
</tr>
<tr>
<td>PackageKey</td>
<td>Guid</td>
<td>UpgradeCode property defined in associated MSI files</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of package version</td>
</tr>
<tr>
<td>Major</td>
<td>Int32</td>
<td>Major version of package</td>
</tr>
<tr>
<td>Minor</td>
<td>Int32</td>
<td>Minor version of package</td>
</tr>
<tr>
<td>Build</td>
<td>Int32</td>
<td>Build version of package</td>
</tr>
<tr>
<td>Revision</td>
<td>Int32</td>
<td>Revision version of package</td>
</tr>
<tr>
<td>DataLength</td>
<td>Int32</td>
<td>Size of package</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>InProgress</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>String</td>
<td>Name of user</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of package version</td>
</tr>
<tr>
<td>CreatorMajor</td>
<td>Int32</td>
<td>Major version of package creator (console)</td>
</tr>
<tr>
<td>CreatorMinor</td>
<td>Int32</td>
<td>Minor version of package creator (console)</td>
</tr>
<tr>
<td>CreatorBuild</td>
<td>Int32</td>
<td>Build version of package creator (console)</td>
</tr>
<tr>
<td>CreatorRevision</td>
<td>Int32</td>
<td>Revision version of package creator (console)</td>
</tr>
<tr>
<td>DependentMinimumMajor</td>
<td>Int32</td>
<td>Major version of minimum associated agent</td>
</tr>
<tr>
<td>DependentMinimumMinor</td>
<td>Int32</td>
<td>Minor version of minimum associated agent</td>
</tr>
<tr>
<td>DependentMinimumBuild</td>
<td>Int32</td>
<td>Build version of minimum associated agent</td>
</tr>
<tr>
<td>DependentMinimumRevision</td>
<td>Int32</td>
<td>Revision version of minimum associated agent</td>
</tr>
<tr>
<td>DependentMaximumMajor</td>
<td>Int32</td>
<td>Major version of maximum associated agent</td>
</tr>
<tr>
<td>DependentMaximumMinor</td>
<td>Int32</td>
<td>Minor version of maximum associated agent</td>
</tr>
<tr>
<td>DependentMaximumBuild</td>
<td>Int32</td>
<td>Build version of maximum associated agent</td>
</tr>
<tr>
<td>DependentMaximumRevision</td>
<td>Int32</td>
<td>Revision version of maximum associated agent</td>
</tr>
<tr>
<td>PackagesRow</td>
<td>PackagesRow</td>
<td></td>
</tr>
</tbody>
</table>

**PatchesRow**

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
</table>

26
<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatchKey</td>
<td>Guid</td>
<td>The unique identifier of the patch</td>
</tr>
<tr>
<td>PackageVersionKey</td>
<td>Guid</td>
<td>The package version that this patch applies to</td>
</tr>
<tr>
<td>PatchCode</td>
<td>Guid</td>
<td>The Patch Code property of the Windows Installer MSP file</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of the patch</td>
</tr>
<tr>
<td>Major</td>
<td>Int32</td>
<td>Major version of package once this patch is applied</td>
</tr>
<tr>
<td>Minor</td>
<td>Int32</td>
<td>Minor version of package once this patch is applied</td>
</tr>
<tr>
<td>Build</td>
<td>Int32</td>
<td>Build version of package once this patch is applied</td>
</tr>
<tr>
<td>Revision</td>
<td>Int32</td>
<td>Revision version of package once this patch is applied</td>
</tr>
<tr>
<td>DataLength</td>
<td>Int32</td>
<td>Size of patch</td>
</tr>
<tr>
<td>InProgress</td>
<td>Boolean</td>
<td>True whenever this patch is being updated</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>TargetMajor</td>
<td>Int32</td>
<td>The version of the patch or package that this patch applies to</td>
</tr>
<tr>
<td>TargetMinor</td>
<td>Int32</td>
<td>The version of the patch or package that this patch applies to</td>
</tr>
<tr>
<td>TargetBuild</td>
<td>Int32</td>
<td>The version of the patch or package that this patch applies to</td>
</tr>
<tr>
<td>TargetRevision</td>
<td>Int32</td>
<td>The version of the patch or package that this patch applies to</td>
</tr>
<tr>
<td>ValidationFlags</td>
<td>Int32</td>
<td>The Validation Flags property of the Windows Installer MSP file</td>
</tr>
</tbody>
</table>

**PrerequisitesRow**

Represents a prerequisite on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrerequisitesKey</td>
<td>Guid</td>
<td>Unique identifier relating to prerequisites</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of prerequisite</td>
</tr>
<tr>
<td>Version</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>PlatformInfo</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>VersionInfo</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>Column</td>
<td>DataType</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>PackageVersionKey</td>
<td>Guid</td>
<td>Unique identifier relating to package version</td>
</tr>
<tr>
<td>PrerequisitesKey</td>
<td>Guid</td>
<td>Unique identifier relating to prerequisites</td>
</tr>
<tr>
<td>PrerequisitesRow</td>
<td>PrerequisitesRow</td>
<td></td>
</tr>
</tbody>
</table>
**PrerequisiteCheckRow**

Represents a prerequisite check on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrerequisiteCheckKey</td>
<td>Guid</td>
<td>Unique identifier relating to prerequisite check</td>
</tr>
<tr>
<td>PrerequisiteKey</td>
<td>Guid</td>
<td>Unique identifier relating to prerequisite</td>
</tr>
<tr>
<td>CheckType</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>OperatorValue</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Product_ProductCode</td>
<td>Guid</td>
<td></td>
</tr>
<tr>
<td>Product_UpgradeCode</td>
<td>Guid</td>
<td></td>
</tr>
<tr>
<td>File_Path</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Registry_Root</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Registry_Key</td>
<td>String</td>
<td>Unique identifier relating to registry registry</td>
</tr>
<tr>
<td>Registry_Value</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>OperatingSystem_Message</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>PrerequisitesRow</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PrerequisiteExitCodeRow**

Represents a prerequisite exit code on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExitCodeKey</td>
<td>Guid</td>
<td>Unique identifier relating to exit code</td>
</tr>
<tr>
<td>Value</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>Result</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>CommandKey</td>
<td>Guid</td>
<td>Unique identifier relating to command</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>PrerequisiteCommandRow</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DataRows - Products

Products

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductKey</td>
<td>Guid</td>
<td>Unique identifier relating to product</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of product</td>
</tr>
<tr>
<td>Icon</td>
<td>Byte[]</td>
<td></td>
</tr>
<tr>
<td>SupportsAgents</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>SupportsConfigurations</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>SupportsSoftware</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>
DataRows - Reports

Reports

ReportDefinitionsRow

Represents a report definition on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReportDefinitionKey</td>
<td>Guid</td>
<td>Unique identifier relating to report definition</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of report definition</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of report definition</td>
</tr>
<tr>
<td>Category</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ProductKey</td>
<td>Guid</td>
<td>Unique identifier relating to product</td>
</tr>
<tr>
<td>ProductName</td>
<td>String</td>
<td>Name of product</td>
</tr>
<tr>
<td>Visible</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>DataLength</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>PolicyKey</td>
<td>Guid</td>
<td>Unique identifier relating to policy</td>
</tr>
<tr>
<td>OwnerSid</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>SecurityDescriptor</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecurityRoleKey</td>
<td>Guid</td>
<td>Unique identifier relating to security role</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of security role</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Description of security role</td>
</tr>
<tr>
<td>PermissionsMask</td>
<td>Int64</td>
<td></td>
</tr>
<tr>
<td>RoleType</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>

ServerPermissionsRow
Represents a server permission on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of server permission</td>
</tr>
<tr>
<td>Mask</td>
<td>Int64</td>
<td></td>
</tr>
</tbody>
</table>

ObjectPermissionsRow
Represents an object permission on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of object permission</td>
</tr>
<tr>
<td>Mask</td>
<td>Int64</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserKey</td>
<td>Guid</td>
<td>Unique identifier relating to user</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of user</td>
</tr>
<tr>
<td>Sid</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td>DataType</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>IsGroup</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>IsMember</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>PolicyKey</td>
<td>Guid</td>
<td>Unique identifier relating to policy</td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>SecurityDescriptor</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

**SecurityElementsRow**

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

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecurityElementKey</td>
<td>Guid</td>
<td>Unique identifier relating to security element</td>
</tr>
<tr>
<td>ElementType</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>PolicyKey</td>
<td>Guid</td>
<td>Unique identifier relating to policy</td>
</tr>
<tr>
<td>SecurityRoleKey</td>
<td>Guid</td>
<td>Unique identifier relating to security role</td>
</tr>
<tr>
<td>UserSid</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolicyKey</td>
<td>Guid</td>
<td>Unique identifier relating to policy</td>
</tr>
<tr>
<td>Type</td>
<td>Int32</td>
<td></td>
</tr>
</tbody>
</table>
DataRows - Servers

Servers

ServersRow

 Represents a server on the server

<table>
<thead>
<tr>
<th>Column</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerKey</td>
<td>Guid</td>
<td>Unique identifier relating to server</td>
</tr>
<tr>
<td>GroupKey</td>
<td>Guid</td>
<td>Unique identifier relating to group</td>
</tr>
<tr>
<td>Url</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Int32</td>
<td></td>
</tr>
<tr>
<td>PerformDiagnostics</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>DateTime</td>
<td>Time created</td>
</tr>
<tr>
<td>ModifiedTime</td>
<td>DateTime</td>
<td>Time modified</td>
</tr>
<tr>
<td>Disabled</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>
## Namespace Index

### Namespace List

Here is a list of all documented namespaces with brief descriptions:

<table>
<thead>
<tr>
<th>Namespace</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DataAccessServices</code></td>
<td>39</td>
</tr>
<tr>
<td><code>DataAccessServices.WebServices</code></td>
<td>40</td>
</tr>
</tbody>
</table>
Hierarchical Index

Class Hierarchy
This inheritance list is sorted roughly, but not completely, alphabetically:

WebService
  DataAccessServices.WebServices.Alerts.................................................................42
  DataAccessServices.WebServices.Conditions ..........................................................54
  DataAccessServices.WebServices.DiscoveredMachines .............................................69
  DataAccessServices.WebServices.Events ...............................................................80
  DataAccessServices.WebServices.Groups ..............................................................89
  DataAccessServices.WebServices.Machines .............................................................106
  DataAccessServices.WebServices.Maintenance .......................................................120
  DataAccessServices.WebServices.Products .............................................................152
  DataAccessServices.WebServices.Servers ................................................................171

WebService
  DataAccessServices.WebServices.Queries ..............................................................155
Class Index

Class List
Here are the classes, structs, unions and interfaces with brief descriptions:

**DataAccessServices.WebServices.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.) ........42

**DataAccessServices.WebServices.Conditions** (The **Conditions** WebService creates conditions for use by Machine Discovery. For more information, refer to the DiscoveredMachine WebService page.) 54

**DataAccessServices.WebServices.DatabaseWebService** (Provides methods for retrieving information about the current Management Server database.) ........................................59

**DataAccessServices.WebServices.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.) .....................62

**DataAccessServices.WebServices.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.) ..............................................................69

**DataAccessServices.WebServices.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) .................80

**DataAccessServices.WebServices.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.) ....................89

**DataAccessServices.WebServices.Housekeeping** (Summary description for **Events**) .................101

**DataAccessServices.WebServices.Licenses** (Manages licenses in the Management Centre. When using the AppSense Management Suite a valid license must be used.) ..................................................102

**DataAccessServices.WebServices.Machines** (Manages machines in the Management Centre. The **Machines** table stores an entry for each machine managed by the Management Center.) ......106

**DataAccessServices.WebServices.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 events/alerts that can be raised via the product agents) ..................................................120

**DataAccessServices.WebServices.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.) ..................................132

**DataAccessServices.WebServices.Products** (The ProductsWebService creates product entries needed for uploading and managing packages. For more information, refer to the PackagesWebService.) 152

**DataAccessServices.WebServices.Queries** (Provides methods for querying the Management Server database via the reporting engine.) .................................................................155

**DataAccessServices.WebServices.Reports** (Manages reports in the Management Centre. Report definitions store the necessary data for generating the reports visible within the Management Console.) ..................................................157
DataAccessServices.WebServices.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. ) .......................................................... 163

DataAccessServices.WebServices.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. ) ................................................................. 171
Namespace Documentation

DataAccessServices Namespace Reference

Namespaces
  • namespace WebServices
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 **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 events/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 an SNMP trap or SMTP (email) action when the alert is triggered.

Inheritance diagram for DataAccessServices.WebServices.Alerts:

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 an 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:**

<table>
<thead>
<tr>
<th>actionKey</th>
<th>The action key.</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The unique name of the configuration.</td>
</tr>
<tr>
<td>value</td>
<td>The value of the configuration property.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the configuration was last modified.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

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

Adds an event to an existing alert.

**Parameters:**

<table>
<thead>
<tr>
<th>alertKey</th>
<th>The key of the alert.</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventKey</td>
<td>The key of the event.</td>
</tr>
</tbody>
</table>
void DataAccessServices.WebServices.Alerts.ApplyActionChanges (ref ActionsDataSet actionsChanges)[inline]

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionsChanges</td>
<td>The data set consisting of changes.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertChanges</td>
<td>The data set consisting of changes.</td>
</tr>
</tbody>
</table>

Requires group or administrative access.


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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertRules</td>
<td>A data set consisting of changes.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.


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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertRules</td>
<td>A data set consisting of changes.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionKey</td>
<td>The key that identifies the action.</td>
</tr>
<tr>
<td>alertRuleKey</td>
<td>The alert rule key that this action is associated with.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the action.</td>
</tr>
<tr>
<td>description</td>
<td>A description of the action.</td>
</tr>
<tr>
<td>type</td>
<td>The type, such as SNMP or SMTP.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the action was last modified.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertRuleKey</td>
<td>The alert rule which defines this alert.</td>
</tr>
<tr>
<td>groupKey</td>
<td>The key of the group containing the machine that generated this alert (Optional).</td>
</tr>
<tr>
<td>machineKey</td>
<td>The key of the machine which generated this alert.</td>
</tr>
<tr>
<td>status</td>
<td>The status of the alert.</td>
</tr>
<tr>
<td>time</td>
<td>The time that the alert occurred.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the alert was last modified.</td>
</tr>
</tbody>
</table>

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:

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

Requires alert or administrative access.
void DataAccessServices.WebServices.Alerts.DeleteAction (Guid  actionKey, DateTime?  
modifiedTime)[inline]

Deletes an existing action from the database.

Parameters:

<table>
<thead>
<tr>
<th>actionKey</th>
<th>The key that identifies the action to delete.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>The time that the action was last modified.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

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

Deletes an alert from the database.

Parameters:

<table>
<thead>
<tr>
<th>alertKey</th>
<th>The key that identifies the alert.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>The time that the alert was last modified.</td>
</tr>
</tbody>
</table>

Requires administrative access.

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

Deletes an existing alert rule.

Parameters:

<table>
<thead>
<tr>
<th>alertRuleKey</th>
<th>The key that identifies the alert rule to be deleted.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>The time that the alert rule was last modified.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

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

Deletes all alerts generated by machines within the specified group.

Parameters:

<table>
<thead>
<tr>
<th>groupKey</th>
<th>The key that identifies the group.</th>
</tr>
</thead>
</table>

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.


Deletes all alerts with the given severity.

**Parameters:**

| severity         | The severity of alerts to delete. |

Requires administrative access.


Deletes all alerts with the given status.

**Parameters:**

| status           | The status of alerts to delete. |

Requires administrative access.


Deletes all alerts generated at the specified times.

**Parameters:**

<table>
<thead>
<tr>
<th>startTime</th>
<th>Start time of period of alerts to delete.</th>
</tr>
</thead>
<tbody>
<tr>
<td>endTime</td>
<td>End time of period of alerts to delete.</td>
</tr>
</tbody>
</table>

Requires administrative access.

**ActionsDataSet DataAccessServices.WebServices.Alerts.GetActionFromAlertRuleKey (Guid alertRuleKey)**

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.


Returns all alert rules.

**Returns:**

A data set consisting of all alert rules.
Requires alert or administrative access.


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:

<table>
<thead>
<tr>
<th>groupKey</th>
<th>The key of the group.</th>
</tr>
</thead>
</table>

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:

<table>
<thead>
<tr>
<th>machineKey</th>
<th>The key of the machine.</th>
</tr>
</thead>
</table>

Returns:
A data set consisting of the alerts.
Requires alert or administrative access.


Returns all alerts with the given severity.

Parameters:

<table>
<thead>
<tr>
<th>severity</th>
<th>The severity of the alerts to return.</th>
</tr>
</thead>
</table>

Returns:
All alerts with the specified severity.
Requires alert, group, or administrative access.


Returns all alerts with the given status.

Parameters:

<table>
<thead>
<tr>
<th>status</th>
<th>The status of the alerts to return.</th>
</tr>
</thead>
</table>

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:

<table>
<thead>
<tr>
<th>startTime</th>
<th>The earliest time alerts are to be returned.</th>
</tr>
</thead>
<tbody>
<tr>
<td>endTime</td>
<td>The latest time that alerts are to be returned.</td>
</tr>
</tbody>
</table>

Returns:

The alerts that were created within the specified time.
Requires alert, group, or administrative access.

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

Removes an existing configuration property from an action.

Parameters:

<table>
<thead>
<tr>
<th>actionKey</th>
<th>The key that identifies the action.</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The unique name of the configuration.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the configuration was last modified.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

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

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

Parameters:

<table>
<thead>
<tr>
<th>alertKey</th>
<th>The key of the alert.</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventKey</td>
<td>The key of the event.</td>
</tr>
</tbody>
</table>

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

Updates properties of an action.

Parameters:

<table>
<thead>
<tr>
<th>actionKey</th>
<th>The key that identifies the action.</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the action.</td>
</tr>
<tr>
<td>description</td>
<td>A description of the action.</td>
</tr>
<tr>
<td>type</td>
<td>The type, such as SNMP or SMTP.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the action was last modified.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

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

Updates properties of a configuration.
Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionKey</td>
<td>The action key.</td>
</tr>
<tr>
<td>name</td>
<td>The unique name of the configuration.</td>
</tr>
<tr>
<td>value</td>
<td>The value of the configuration property.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the configuration was last modified.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

```csharp
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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertKey</td>
<td>The key of the alert to update.</td>
</tr>
<tr>
<td>groupKey</td>
<td>The key of the group containing the machine that generated this alert (Optional).</td>
</tr>
<tr>
<td>machineKey</td>
<td>The key of the machine which generated this alert.</td>
</tr>
<tr>
<td>status</td>
<td>The status of the alert.</td>
</tr>
<tr>
<td>time</td>
<td>The time that the alert occurred.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the alert was last modified.</td>
</tr>
</tbody>
</table>

Requires administrative access.

```csharp
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:

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

Requires alert or administrative access.
void DataAccessServices.WebServices.Alerts.UpdateAlertRuleSecurity (Guid alertRuleKey, Guid? policyKey, String ownerSid, ref DateTime modifiedTime)

Updates security permissions associated with the alert rule.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertRuleKey</td>
<td>The key that identifies the new alert rule.</td>
</tr>
<tr>
<td>policyKey</td>
<td>The key with identifies the policy associated with this alert rule.</td>
</tr>
<tr>
<td>ownerSid</td>
<td>The security identifier associated with the user that owns this entry.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the alert rule was last modified.</td>
</tr>
</tbody>
</table>

Requires alert, security, or administrative access.

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

- Alerts.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.

Inheritance diagram for DataAccessServices.WebServices.Conditions:

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, 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:**

<table>
<thead>
<tr>
<th>conditions</th>
<th>The dataset to update.</th>
</tr>
</thead>
</table>

**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:**

<table>
<thead>
<tr>
<th>conditionKey</th>
<th>OUT parameter will contain the key of the condition when the call returns. Passed in value unused.</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The GUID of the group to attach this condition to.</td>
</tr>
<tr>
<td>conditionType</td>
<td>The type of the condition, changes how the condition is interpreted.</td>
</tr>
<tr>
<td>isInclude</td>
<td>Whether this condition includes or excludes the affected machines. Excludes take priority.</td>
</tr>
<tr>
<td>includeChildren</td>
<td>Whether to include child nodes in this condition (OU/Computer groups).</td>
</tr>
<tr>
<td>adObjectGuid</td>
<td>The GUID of the node in question (OU/Computer groups).</td>
</tr>
<tr>
<td>adObjectDistinguishedName</td>
<td>The distinguished name of the group in question (OU/Computer groups).</td>
</tr>
<tr>
<td>domain</td>
<td>The domain of the computer.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter for the computer.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the condition on return. Passed in value unused.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The GUID of the group to attach this condition to.</td>
</tr>
<tr>
<td>isInclude</td>
<td>Whether this condition includes or excludes the affected machines. Excludes take priority.</td>
</tr>
<tr>
<td>includeChildren</td>
<td>Whether to include child computer groups in this condition.</td>
</tr>
<tr>
<td>adObjectGuid</td>
<td>The GUID of the group in question.</td>
</tr>
<tr>
<td>adObjectDistinguishedName</td>
<td>The distinguished name of the group in question.</td>
</tr>
<tr>
<td>domain</td>
<td>The domain of the computer.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter for the computer.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The GUID of the group to attach this condition to.</td>
</tr>
<tr>
<td>isInclude</td>
<td>Whether this condition includes or excludes the affected machines. Excludes take priority.</td>
</tr>
<tr>
<td>includeChildren</td>
<td>Whether to include child containers in this condition.</td>
</tr>
<tr>
<td>adObjectGuid</td>
<td>The GUID of the container in question.</td>
</tr>
<tr>
<td>adObjectDistinguishedName</td>
<td>The distinguished name of the container in question.</td>
</tr>
<tr>
<td>domain</td>
<td>The domain of the computer.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter for the computer.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The GUID of the group to attach this condition to.</td>
</tr>
<tr>
<td>isInclude</td>
<td>Whether this condition includes or excludes the affected machines. Excludes take priority.</td>
</tr>
<tr>
<td>includeChildren</td>
<td>Whether to include child containers in this condition.</td>
</tr>
<tr>
<td>adObjectGuid</td>
<td>The GUID of the container in question.</td>
</tr>
<tr>
<td>adObjectDistinguishedName</td>
<td>The distinguished name of the container in question.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>groupKey</th>
<th>The GUID of the group to attach this computer to.</th>
</tr>
</thead>
<tbody>
<tr>
<td>isInclude</td>
<td>Whether this condition includes or excludes the affected machines. Excludes take priority.</td>
</tr>
<tr>
<td>domain</td>
<td>The domain of the computer.</td>
</tr>
<tr>
<td>filter</td>
<td>Filter for the computer.</td>
</tr>
</tbody>
</table>

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

Immediately deletes a condition.

Parameters:

<table>
<thead>
<tr>
<th>conditionKey</th>
<th>The primary key of the condition to delete.</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The group the the condition belongs to. Used for security checks.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>groupKey</th>
<th>The guid of the group in question.</th>
</tr>
</thead>
</table>

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, bool isInclude, bool? includeChildren, Guid? adObjectGuid, string adObjectDistinguishedName, String domain, String filter, ref DateTime modifiedTime) [inline]

Method to directly update a condition.
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.

Inheritance diagram for DataAccessServices.WebServices.DatabaseWebService:

Public Member Functions

- **NamedValuesDataSet GetInfo ()**
  Returns a data set consisting of name/value pairs of information about the database.

- **String GetName ()**
  Returns the name of this server.

- **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 ()**
  Returns the version of the schema within the database.

- **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 ()**

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.

**Returns:**
The name of this server.

**NamedValuesDataSet** **DataAccessServices.WebServices.DatabaseWebService.GetNameSpace (String nameSpace)**

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 ()**

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 ()**

Returns the date and time on the server.

**Returns:**
Returns a DateTime object in UTC format.

Returns the public key used for communication with the server.

**Returns:**
Public key used for communication with the server as string.


Returns the version of the schema within the database.

**Returns:**
The version of the schema within the database as string.

void DataAccessServices.WebServices.DatabaseWebService.SetNameSpace (String nameSpace, NamedValuesDataSet properties)

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

**Parameters:**

<table>
<thead>
<tr>
<th>nameSpace</th>
<th>Namespace from where the properties will be retrieved from.</th>
</tr>
</thead>
<tbody>
<tr>
<td>properties</td>
<td>Data set containing changes to properties.</td>
</tr>
</tbody>
</table>

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.

Inheritance diagram for DataAccessServices.WebServices.Deployment:

**Public Member Functions**

- **String ActivateDeploymentService (Int32 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 ApplyInstructionsStatusChanges (ref DeploymentDataSet deploymentChanges)**
  *Updates the database with the status changes in the Instructions table.*

- **void ClearStatusHistory (Guid instructionKey, Guid?groupKey)**
  *Clear status history*

- **void CreateCredentials (Guid credentialsKey, Guid?groupKey, String userName, String password, Int32 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, Int32 commandID, Guid?groupKey, String plugIn, Guid?machineKey, String status, out DateTime modifiedTime)**
  *Creates a new deployment instruction.*

- **void DeleteCredentials (Guid credentialsKey, DateTime?modifiedTime)**
  *Deletes credentials.*

- **void DeleteInstructions (Guid instructionKey, DateTime?modifiedTime)**
  *Deletes instructions.*
- 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 ()
  Returns a Deployment data set containing CCA deployment instructions.
- DeploymentDataSet GetDeploymentInstructionsFromDiscoveredMachineKey (Guid discoveredMachineKey)
  Returns a data set consisting of all instructions for a discovered machine
- DeploymentDataSet GetDeploymentInstructionsFromGroupKey (Int32 commandID)
  Returns a data set consisting of all instructions for a command ID.
- DeploymentDataSet GetStatusHistory (Guid instructionKey)
  Get status history
- void SetInstructionsCommandID (Guid instructionKey, Int32 commandID)
  Set the command ID for a given instruction key
- void UpdateCredentials (Guid credentialsKey, Guid?groupKey, String userName, String password, Int32 index, ref DateTime modifiedTime)
  Updates existing credentials. To encrypt the password use DatabaseWebService.GetTransportPublicKey to retrieve the public key and encrypt using RSACryptoServiceProvider.Encrypt.
- void UpdateInstructions (Guid instructionKey, Int32 commandID, Guid?groupKey, String plugIn, Guid?machineKey, String status, String settings, ref DateTime modifiedTime)
  Updates an existing deployment instruction.

### 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 (Int32 commandID)[inline]**

Activate Deployment Service

**Parameters:**

| commandID | Command for the deployment service. |

Here we will signal either the global pollnow event or the global DeployCCANow event. If the events are not present or the events don't have the required privilages then nothing happens

**Returns:**

Messages from the deployment service.
Command ID is 128 for deployment and 129 for poll now.

```csharp
```

Updates the database with the changes in the Credentials table.

**Parameters:**

| deploymentChanges | Data set containing changes to be applied. |

```csharp
void DataAccessServices.WebServices.Deployment.ApplyInstructionsChanges(ref DeploymentDataSet deploymentChanges, Guid? groupKey)
```

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. |

```csharp
void DataAccessServices.WebServices.Deployment.ApplyInstructionsStatusChanges(ref DeploymentDataSet deploymentChanges)
```

Updates the database with the status changes in the Instructions table.

**Parameters:**

| deploymentChanges | Data set containing changes to the instructions table. |

```csharp
void DataAccessServices.WebServices.Deployment.ClearStatusHistory(Guid instructionKey, Guid? groupKey)
```

Clear status history.

**Parameters:**

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

```csharp
void DataAccessServices.WebServices.Deployment.CreateCredentials(Guid credentialsKey, Guid? groupKey, String userName, String password, Int32 index, out DateTime modifiedTime)
```

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

<table>
<thead>
<tr>
<th>credentialsKey</th>
<th>A unique key for the credentials entry.</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The group this credentials entry is associated with.</td>
</tr>
<tr>
<td>userName</td>
<td>Username for the deployment credentials.</td>
</tr>
<tr>
<td>password</td>
<td>Password for the deployment credentials.</td>
</tr>
<tr>
<td>index</td>
<td>Order in which access with this credential will be attempted.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the credentials on return. Passed in value unused.</td>
</tr>
</tbody>
</table>

```csharp
void DataAccessServices.WebServices.Deployment.CreateInstructions (Guid instructionKey, Int32 commandID, Guid? groupKey, String plugIn, Guid? machineKey, String status, out DateTime modifiedTime)[inline]
```

Creates a new deployment instruction.

**Parameters:**

<table>
<thead>
<tr>
<th>instructionKey</th>
<th>A unique key for this deployment instruction.</th>
</tr>
</thead>
<tbody>
<tr>
<td>commandID</td>
<td>Command for this instruction.</td>
</tr>
<tr>
<td>groupKey</td>
<td>Key identifying which group this instruction is associated with.</td>
</tr>
<tr>
<td>plugIn</td>
<td>Plugin which will run this instruction.</td>
</tr>
<tr>
<td>machineKey</td>
<td>Key identifying which machine this instruction will be run against.</td>
</tr>
<tr>
<td>status</td>
<td>Initial status of this instruction.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the instruction on return. Passed in value unused.</td>
</tr>
</tbody>
</table>

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

```csharp
void DataAccessServices.WebServices.Deployment.DeleteCredentials (Guid credentialsKey, DateTime? modifiedTime)[inline]
```

Deletes credentials.

**Parameters:**

<table>
<thead>
<tr>
<th>credentialsKey</th>
<th>The key for the credentials entry.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>Time this entry was last modified.</td>
</tr>
</tbody>
</table>

```csharp
void DataAccessServices.WebServices.Deployment.DeleteInstructions (Guid instructionKey, DateTime? modifiedTime)[inline]
```

Deletes instructions.

**Parameters:**

<table>
<thead>
<tr>
<th>instructionKey</th>
<th>Key for the deployment instruction.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>Time this instruction entry was last modified.</td>
</tr>
</tbody>
</table>

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.


Returns a Deployment data set containing CCA deployment instructions.

Returns:
Deployment data set containing CCA deployment instructions.

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 (Int32 commandID)[inline]

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commandID</td>
<td>Command for this instruction.</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instructionKey</td>
<td>Key for the deployment instruction.</td>
</tr>
</tbody>
</table>

**Returns:**

Data set containing status history for the specified instruction.

**void**

`DataAccessServices.WebServices.Deployment.SetInstructionsCommandID (Guid instructionKey, Int32 commandID)` [inline]

Set the command ID for a given instruction key

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instructionKey</td>
<td>Key identifying the machine these instructions are associated with.</td>
</tr>
<tr>
<td>commandID</td>
<td>Command for this instruction.</td>
</tr>
</tbody>
</table>

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

**void**

`DataAccessServices.WebServices.Deployment.UpdateCredentials (Guid credentialsKey, Guid? groupKey, String userName, String password, Int32 index, ref DateTime modifiedTime)` [inline]

Updates existing credentials. To encrypt the password use `DatabaseWebService.GetTransportPublicKey` to retrieve the public key and encrypt using `RSACryptoServiceProvider.Encrypt`.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>credentialsKey</td>
<td>The key for the credentials entry.</td>
</tr>
<tr>
<td>groupKey</td>
<td>The group this credentials entry is associated with.</td>
</tr>
<tr>
<td>userName</td>
<td>Username for the deployment credentials.</td>
</tr>
<tr>
<td>password</td>
<td>Password for the deployment credentials.</td>
</tr>
<tr>
<td>index</td>
<td>Order in which access with this credential will be attempted.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Time this entry was last modified.</td>
</tr>
</tbody>
</table>

**void**

`DataAccessServices.WebServices.Deployment.UpdateInstructions (Guid instructionKey, Int32 commandID, Guid? groupKey, String plugIn, Guid? machineKey, String status, String settings, ref DateTime modifiedTime)` [inline]

Updates an existing deployment instruction.
Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instructionKey</td>
<td>Key for the deployment instruction.</td>
</tr>
<tr>
<td>commandID</td>
<td>Command for this instruction.</td>
</tr>
<tr>
<td>groupKey</td>
<td>Key identifying which group this instruction is associated with.</td>
</tr>
<tr>
<td>plugIn</td>
<td>Plugin which will run this instruction.</td>
</tr>
<tr>
<td>machineKey</td>
<td>Key identifying which machine this instruction will be run against.</td>
</tr>
<tr>
<td>status</td>
<td>Initial status of this instruction.</td>
</tr>
<tr>
<td>settings</td>
<td>Settings for this instruction.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Time this instruction entry was last modified.</td>
</tr>
</tbody>
</table>

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

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.

Inheritance diagram for DataAccessServices.WebServices.DiscoveredMachines:

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.

• 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:**

<table>
<thead>
<tr>
<th><strong>machineChanges</strong></th>
<th>The data set consisting of the changes.</th>
</tr>
</thead>
</table>

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:**

<table>
<thead>
<tr>
<th><strong>discoveredMachineIds</strong></th>
<th>Array of discovered machine identifiers.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>userSid</strong></td>
<td>The sid of the user.</td>
</tr>
<tr>
<td><strong>groupSids</strong></td>
<td>The list of group sids.</td>
</tr>
<tr>
<td><strong>permissions</strong></td>
<td>The list of permissions to check.</td>
</tr>
</tbody>
</table>

---

**DiscoveredMachinesDataSet**


Creates a new discovered machine entry.
Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>discoveredMachinPK</td>
<td>A unique key for this discovered machine.</td>
</tr>
<tr>
<td>expectedGroupFK</td>
<td>Key identifying the group this machine is expected to be associated with.</td>
</tr>
<tr>
<td>netBiosName</td>
<td>The NetBIOS name of the machine.</td>
</tr>
<tr>
<td>adObjectGuid</td>
<td>GUID identifier for the machine in Active Directory.</td>
</tr>
<tr>
<td>description</td>
<td>Description of this machine.</td>
</tr>
<tr>
<td>operatingSystem</td>
<td>Operating system of the machine.</td>
</tr>
<tr>
<td>actualGroupFK</td>
<td>Key identifying the group the machine is actually associated with.</td>
</tr>
<tr>
<td>actualGroupName</td>
<td>Name of the group the machine is actually associated with.</td>
</tr>
<tr>
<td>dns</td>
<td>Fully qualified DNS entry for the machine.</td>
</tr>
<tr>
<td>userSpecified</td>
<td>TRUE if specified by the user. FALSE if discovered via membership rules.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the machine on return. Passed in value unused.</td>
</tr>
</tbody>
</table>

Returns:

Returns data set containing details of the created machine

```csharp
void DataAccessServices.WebServices.DiscoveredMachines.DeleteAllMachines ()
```

Deletes all discovered machines.

```csharp
```

Deletes the specified machine.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>discoveredMachinPK</td>
<td>Key for the discovered machine.</td>
</tr>
</tbody>
</table>

Returns data set with details of all discovered machines.

```csharp
DiscoveredMachinesDataSet
DataAccessServices.WebServices.DiscoveredMachines.FindMachines (String match)
```

Parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>match</td>
<td>Not currently used.</td>
</tr>
</tbody>
</table>

Returns:

Data set with details of all discovered machines.

```csharp
SchedulerConfigurationDto
```

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adObjectGuid</td>
<td>GUID identifier for the machine in Active Directory.</td>
</tr>
<tr>
<td>dns</td>
<td>Fully qualified DNS entry for the machine.</td>
</tr>
<tr>
<td>netbiosName</td>
<td>The NetBIOS name of the machine.</td>
</tr>
</tbody>
</table>

Returns:
Data set with details of the specified discovered machine.


Returns data set with details of the specified discovered machine.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dns</td>
<td>The DNS name of the discoveredMachine</td>
</tr>
<tr>
<td>netbios</td>
<td>The NETBIOS name of the discoveredMachine</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>machineKey</td>
<td>Key for the discovered machine.</td>
</tr>
</tbody>
</table>

Returns:
Data set with details of the specified discovered machine.


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


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:

<table>
<thead>
<tr>
<th>GroupKey</th>
<th>Key for the group associated with the discovered machines.</th>
</tr>
</thead>
<tbody>
<tr>
<td>withSummary</td>
<td>Boolean flag indicating if the diagnostics and deployment state should be retrieved.</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>consoleId</td>
<td>Id of the invoking console used to obtain the last refresh date referenced in MachineRefresh table.</td>
</tr>
<tr>
<td>withSummary</td>
<td>Boolean flag indicating if the diagnostics and deployment state should be retrieved.</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>Key for the group associated with the discovered machines.</td>
</tr>
<tr>
<td>misgroupedCount</td>
<td>The count of misgrouped machines discovered via membership rules.</td>
</tr>
<tr>
<td>userSpecifiedItemCount</td>
<td>The count of user specified machines.</td>
</tr>
</tbody>
</table>

**MisgroupedMachinesDataSet**


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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>Key for the group associated with the discovered machines.</td>
</tr>
</tbody>
</table>

**Returns:**

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

Returns the first set of DiscoveredMachines.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>topRows</td>
<td>The number of rows to return.</td>
</tr>
</tbody>
</table>

Returns:

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

DiscoveredMachinesDataSet
DataAccessServices.WebServices.DiscoveredMachines.GetPreloadFromGroupKey(Guid  groupKey, int  topRows)

Returns the first set of DiscoveredMachines within a particular group.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>Id of the group.</td>
</tr>
<tr>
<td>topRows</td>
<td>The number of rows to return.</td>
</tr>
</tbody>
</table>

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)

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>Key for the group associated with the discovered machines.</td>
</tr>
</tbody>
</table>

Returns:

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

void DataAccessServices.WebServices.DiscoveredMachines.InvokeDiscovery()

Invokes the discovery service.

Boolean DataAccessServices.WebServices.DiscoveredMachines.IsDiscoveryActive()

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

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

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

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>destinationGroupKey</code></td>
<td>Move a list of discovered machines from one group to another.</td>
</tr>
<tr>
<td><code>discoveredMachineIds</code></td>
<td>List of ids for the machines to move.</td>
</tr>
</tbody>
</table>

```csharp
```

Defines the period to trigger a discovery.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>interval</code></td>
<td>The number of milliseconds between each automatic discovery.</td>
</tr>
</tbody>
</table>

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

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

Returns a SchedulerConfigurationDto object containing information about the Discovery Service.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

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.

```csharp
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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>discoveredMachineKey</code></td>
<td>Key for the discovered machine.</td>
</tr>
<tr>
<td><code>expectedGroupKey</code></td>
<td>Key identifying the group that this machine is expected to be associated with.</td>
</tr>
<tr>
<td>netBiosName</td>
<td>The NetBIOS name of the machine.</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>Description of this machine.</td>
</tr>
<tr>
<td>operatingSystem</td>
<td>Operating system of the machine.</td>
</tr>
<tr>
<td>dns</td>
<td>Fully qualified DNS entry for the machine.</td>
</tr>
<tr>
<td>modifiedDiscovery</td>
<td>Time this entry was last modified.</td>
</tr>
</tbody>
</table>

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.

Inheritance diagram for DataAccessServices.WebServices.Events:

Public Member Functions

- void AddEventDateTimeParam(Int64 eventKey, String paramName, DateTime paramValue)
  Adds a date/time parameter to an existing event.
- void AddEventDefinitionParam(Int32 eventDefinitionKey, String name, String description, EventParamType type, out DateTime modifiedTime)
  Adds a parameter to an existing event definition.
- void AddEventIntegerParam(Int64 eventKey, String paramName, Int64 paramValue)
  Adds an integer parameter to an existing event.
- void AddEventStringParam(Int64 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.
- Int64 CreateEvent(Int32 eventDefinitionKey, Guid? machineKey, Guid? groupKey, String userName, DateTime time)
  Creates a new event.
- void CreateEventDefinition(Int32 eventDefinitionKey, String name, String description, Int32 type, String eventDescription, Guid productKey, Boolean highVolume, Boolean defaultEnabledState, String version, out DateTime modifiedTime)
  Creates a new event definition. Supplied key must not exist.
- void DeleteEvent(Int64 eventKey)
  Deletes an existing event.
void DeleteEventDefinition (Int32 eventDefinitionKey, DateTime? modifiedTime)
Deletes an existing event definition.

void DeleteEventsFromAlertKey (Int32 alertKey)
Deletes events associated with a specific alert.

void DeleteEventsFromGroupKey (Guid? groupKey, Boolean unreferencedOnly)
Deletes events generated by the specified group.

void DeleteEventsFromMachineKey (Guid? machineKey, Boolean unreferencedOnly)
Deletes events generated by the specified machine.

EventDefinitionsDataSet GetEventDefinitions ()
Returns all event definitions.

EventDefinitionsDataSet GetEventDefinitionsFromKey (Int32 eventDefinitionKey)
Returns a specific event definition.

EventsDataSet GetEventFromKey (Int64 eventKey, Boolean withParameters)
Returns a single event.

EventParameterValuesDataSet GetEventParameterValues (Int32 eventNumber, String eventParameter)
Gets the distinct values for an event parameter

EventsDataSet GetEventsFromAlert (Int32 alertKey, Boolean withParameters)
Returns all events associated with the specific alert.

EventsDataSet GetEventsFromGroupKey (Guid? groupKey, Boolean withParameters)
Returns all events associated with a specific group.

EventsDataSet GetEventsFromMachineKey (Guid? machineKey, Boolean withParameters)
Returns all events associated with a specific machine.

EventsDataSet GetEventsFromQuery (String queryString)
Executes the supplied expression returning any results.

EventsDataSet GetEventsFromRange (Int64 firstEventKey, Int64 maxResults, Boolean withParameters)
Returns upto maxResults events, starting with firstEventKey.

DataSet GetExpandedEventsFromQuery (String queryString)
Executes the supplied expression returning any events in an expanded data set.

void RemoveEventDefinitionParam (Int32 eventDefinitionKey, String name, DateTime? modifiedTime)
Removes an existing parameter from an event definition.

void UpdateEventDefinition (Int32 eventDefinitionKey, String name, String description, Int32 type, String eventDescription, Guid productKey, Boolean highVolume, Boolean 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 (Int64 eventKey, String paramName, DateTime paramValue)[inline]

Adds a date/time parameter to an existing event.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventKey</td>
<td>The key of the event to add the parameter to.</td>
</tr>
<tr>
<td>paramName</td>
<td>The name of the parameter.</td>
</tr>
<tr>
<td>paramValue</td>
<td>The date/time value of the parameter.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

void DataAccessServices.WebServices.Events.AddEventDefinitionParam (Int32 eventDefinitionKey, String name, String description, EventParamType type, out DateTime modifiedTime)[inline]

Adds a parameter to an existing event definition.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventDefinitionKey</td>
<td>The key of the event definition to add the parameter onto.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the parameter.</td>
</tr>
<tr>
<td>description</td>
<td>A description of the parameter.</td>
</tr>
<tr>
<td>type</td>
<td>The data type of the parameter values.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the event definition on return. Passed in value unused.</td>
</tr>
</tbody>
</table>

Requires administrative access.

void DataAccessServices.WebServices.Events.AddEventIntegerParam (Int64 eventKey, String paramName, Int64 paramValue)[inline]

Adds an integer parameter to an existing event.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventKey</td>
<td>The key of the event to add the parameter to.</td>
</tr>
<tr>
<td>paramName</td>
<td>The name of the parameter.</td>
</tr>
<tr>
<td>paramValue</td>
<td>The integer value of the parameter.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

void DataAccessServices.WebServices.Events.AddEventStringParam (Int64 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.

```csharp
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

```csharp
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.

```csharp
Int64 DataAccessServices.WebServices.Events.CreateEvent (Int32 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.

```csharp
void DataAccessServices.WebServices.Events.CreateEventDefinition (Int32 eventDefinitionKey, String name, String description, Int32 type, String eventDescription, Guid productKey, Boolean highVolume, Boolean defaultEnabledState, String version, out DateTime modifiedTime)[inline]
```

Creates a new event definition. Supplied key must not exist.
### Parameters:

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

Requires administrative access.

```csharp
void DataAccessServices.WebServices.Events.DeleteEvent (Int64 eventKey) [inline]
```

Deletes an existing event.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventKey</td>
<td>The key of the event to delete.</td>
</tr>
</tbody>
</table>

Requires alert or administrative access.

```csharp
void DataAccessServices.WebServices.Events.DeleteEventDefinition (Int32 eventDefinitionKey, DateTime? modifiedTime) [inline]
```

Deletes an existing event definition.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventDefinitionKey</td>
<td>The key of the event definition.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the event definition was modified.</td>
</tr>
</tbody>
</table>

Requires administrative access.

```csharp
void DataAccessServices.WebServices.Events.DeleteEventsFromAlertKey (Int32 alertKey) [inline]
```

Deletes events associated with a specific alert.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertKey</td>
<td>The alert key whose events will be deleted</td>
</tr>
</tbody>
</table>

```csharp
void DataAccessServices.WebServices.Events.DeleteEventsFromGroupKey (Guid? groupKey, Boolean unreferencedOnly) [inline]
```
Deletes events generated by the specified group.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>groupKey</code></td>
<td>The group that contains the events to delete.</td>
</tr>
<tr>
<td><code>unreferencedOnly</code></td>
<td>Whether to just delete events that are not referenced by an alert.</td>
</tr>
</tbody>
</table>

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

Deletes events generated by the specified machine.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>machineKey</code></td>
<td>The machine that contains the events to delete.</td>
</tr>
<tr>
<td><code>unreferencedOnly</code></td>
<td>Whether to just delete events that are not referenced by an alert.</td>
</tr>
</tbody>
</table>

```csharp
```

Returns all event definitions.

**Returns:**

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

Requires alert or administrative access.

```csharp
EventDefinitionsDataSet DataAccessServices.WebServices.Events.GetEventDefinitionsFromKey (Int32 eventDefinitionKey) [inline]
```

Returns a specific event definition.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>eventDefinitionKey</code></td>
<td>The key of the event definition to return.</td>
</tr>
</tbody>
</table>

**Returns:**

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

Requires alert or administrative access.

```csharp
EventsDataSet DataAccessServices.WebServices.Events.GetEventFromKey (Int64 eventKey, Boolean withParameters) [inline]
```

Returns a single event.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>eventKey</code></td>
<td>The event to return.</td>
</tr>
<tr>
<td><code>withParameters</code></td>
<td>Whether to include event parameters.</td>
</tr>
</tbody>
</table>
**Returns:**
A data set consisting of the event and its parameters.
Requires alert, deployment or administrative access.

**EventParameterValuesDataSet**
`DataAccessServices.WebServices.Events.GetEventParameterValues (Int32 eventNumber, String eventParameter)` [inline]

Gets the distinct values for an event parameter

**Parameters:**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventNumber</td>
<td>The event number.</td>
</tr>
<tr>
<td>eventParameter</td>
<td>The event parameter.</td>
</tr>
</tbody>
</table>

**Returns:**
Data set containing event parameter values.

**EventsDataSet**
`DataAccessServices.WebServices.Events.GetEventsFromAlert (Int32 alertKey, Boolean withParameters)` [inline]

Returns all events associated with the specific alert.

**Parameters:**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alertKey</td>
<td>The key of the alert.</td>
</tr>
<tr>
<td>withParameters</td>
<td>Whether event parameters should be returned.</td>
</tr>
</tbody>
</table>

**Returns:**
Requires alert, deployment or administrative access.

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

Returns all events associated with a specific group.

**Parameters:**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The group key of the events to return.</td>
</tr>
<tr>
<td>withParameters</td>
<td>Whether to include event parameters.</td>
</tr>
</tbody>
</table>

**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, Boolean withParameters)` [inline]

Returns all events associated with a specific machine.
Parameters:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>machineKey</td>
<td>The machine key of the events to return.</td>
</tr>
<tr>
<td>withParameters</td>
<td>Whether to include event parameters.</td>
</tr>
</tbody>
</table>

Returns:

A data set consisting of all events and parameters associated with a machine.
Requires alert or administrative access.

EventsDataSet DataAccessServices.WebServices.Events.GetEventsFromQuery (String queryString) [inline]

Executes the supplied expression returning any results.

Parameters:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryString</td>
<td>The query string.</td>
</tr>
</tbody>
</table>

Returns:

Data set containing events.

EventsDataSet DataAccessServices.WebServices.Events.GetEventsFromRange (Int64 firstEventKey, Int64 maxResults, Boolean withParameters) [inline]

Returns upto maxResults events, starting with firstEventKey.

Parameters:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstEventKey</td>
<td>The earliest value of the key to return.</td>
</tr>
<tr>
<td>maxResults</td>
<td>The maximum number of events to return.</td>
</tr>
<tr>
<td>withParameters</td>
<td>Whether to include event parameters.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryString</td>
<td>The query string.</td>
</tr>
</tbody>
</table>

Returns:

Events in an expanded data set.

void DataAccessServices.WebServices.Events.RemoveEventDefinitionParam (Int32 eventDefinitionKey, String name, DateTime? modifiedTime) [inline]
Removes an existing parameter from an event definition.

**Parameters:**

<table>
<thead>
<tr>
<th>parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventDefinitionKey</td>
<td>The key of the event definition to remove the event from.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the event definition.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the event definition was modified.</td>
</tr>
</tbody>
</table>

Requires administrative access.

```csharp
void DataAccessServices.WebServices.Events.UpdateEventDefinition (Int32 eventDefinitionKey, String name, String description, Int32 type, String eventDescription, Guid productKey, Boolean highVolume, Boolean defaultEnabledState, String version, ref DateTime modifiedTime)
```

Updates an existing event definition.

**Parameters:**

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

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.

Inheritance diagram for DataAccessServices.WebServices.Groups:

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, Int32 major, Int32 minor, Int32 build, Int32 revision, 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.*

- void **CloneGroup** (Guid groupKey)

- void **CreateGroup** (Guid groupKey, String name, String description, Int32 pollPeriodSeconds, Int32 uploadPollPeriodSeconds, Boolean eventLogEnabled, Boolean fileLogEnabled, String fileLogFilename,
Boolean anonymousUserLogging, Boolean anonymousMachineLogging, Boolean overrideServerUrls, out DateTime modifiedTime, Boolean selfRegistrationEnabled, Boolean selfUnregistrationEnabled, Boolean selfUpdateEnabled, int priority, Int32 pollPeriodVariationSeconds, Int32 uploadPollPeriodVariationSeconds, Boolean 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 **GetGroupFromKey** (Guid groupKey, Boolean 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** (Boolean 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, Int32 eventDefinitionKey, Boolean 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, Int32 pollPeriodSeconds, Int32 uploadPollPeriodSeconds, Boolean eventLogEnabled, Boolean fileLogEnabled, String fileLogFilename, Boolean anonymousUserLogging, Boolean anonymousMachineLogging, Boolean overrideServerUrls, Guid?policyKey, String ownerSid, ref DateTime modifiedTime, Boolean selfRegistrationEnabled, Boolean selfUnregistrationEnabled, Boolean selfUpdateEnabled, Int32 priority, Int32 pollPeriodVariationSeconds, Int32 uploadPollPeriodVariationSeconds, Boolean 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, Int32 major, Int32 minor, Int32 build, Int32 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, Int64 mondayFlags, Int64 tuesdayFlags, Int64 wednesdayFlags, Int64 thursdayFlags, Int64 fridayFlags, Int64 saturdayFlags, Int64 sundayFlags, Int32 flags, Boolean doNotInstallAgent, ref DateTime modifiedTime, Int64 mondayConfigFlags, Int64 tuesdayConfigFlags, Int64 wednesdayConfigFlags, Int64 thursdayConfigFlags, Int64 fridayConfigFlags, Int64 saturdayConfigFlags, Int64 sundayConfigFlags, Boolean doNotInstallConfig, Boolean mirrorAgentSchedule, Boolean postponeInstallationPromptEnabled, Int32 postponeInstallationLimit, Boolean downloadConfigurationsOnStartup, Boolean midSessionUpdate, Boolean 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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The key of the group.</td>
</tr>
<tr>
<td>packageKey</td>
<td>The key of the package.</td>
</tr>
<tr>
<td>creationTime</td>
<td>Time the group was created.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Time the group was last modified.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

void DataAccessServices.WebServices.Groups.AddGroupPackage (Guid groupKey, Guid packageKey, Guid? patchKey, Int32 major, Int32 minor, Int32 build, Int32 revision, out DateTime creationTime, out DateTime modifiedTime)[inline]

Adds a package to a group.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The key of the group.</td>
</tr>
<tr>
<td>packageKey</td>
<td>The key of the package.</td>
</tr>
<tr>
<td>patchKey</td>
<td>The patch GUID.</td>
</tr>
<tr>
<td>major</td>
<td>Major version of package version to add.</td>
</tr>
<tr>
<td>minor</td>
<td>Minor version of package version to add.</td>
</tr>
<tr>
<td>build</td>
<td>Build of package version to add.</td>
</tr>
<tr>
<td>revision</td>
<td>Revision of package version to add.</td>
</tr>
<tr>
<td>creationTime</td>
<td>Time the group was created.</td>
</tr>
</tbody>
</table>
### void DataAccessServices.WebServices.Groups.AddGroupPatch (Guid groupKey, Guid packageKey, Guid patchKey, out DateTime creationTime, out DateTime modifiedTime)

Adds a patch to a group.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The key of the group.</td>
</tr>
<tr>
<td>packageKey</td>
<td>The key of the package.</td>
</tr>
<tr>
<td>patchKey</td>
<td>The key of the patch.</td>
</tr>
<tr>
<td>creationTime</td>
<td>Time the group was created.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Time the group was last modified.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.


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

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventFilterChanges</td>
<td>A modified data set.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

### void DataAccessServices.WebServices.Groups.ApplyGroupChanges (ref GroupsDataSet groupChanges)

Applies changes found within the data set to the database.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupChanges</td>
<td>A data set consisting of changes.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

### void DataAccessServices.WebServices.Groups.ApplyGroupPackagesChanges (ref GroupPackagesDataSet groupPackagesChanges)

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

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupPackagesChanges</td>
<td>A data set consisting of changes.</td>
</tr>
</tbody>
</table>
Requires deployment or administrative access.

```csharp
```

Applies changes found within the data set to the database.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupChanges</td>
<td>A data set consisting of changes.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

```csharp
```

Applies the changes of an installation schedule to the database.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scheduleChanges</td>
<td>The schedule changes.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

```csharp
```

Constructs a group.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The key which identifies this group.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the group.</td>
</tr>
<tr>
<td>description</td>
<td>Description of the group.</td>
</tr>
<tr>
<td>pollPeriodSeconds</td>
<td>The amount of seconds between polls.</td>
</tr>
<tr>
<td>uploadPollPeriodSeconds</td>
<td>The amount of seconds between uploads.</td>
</tr>
<tr>
<td>eventLogEnabled</td>
<td>Whether to log events to the event log.</td>
</tr>
<tr>
<td>fileLogEnabled</td>
<td>Whether to log events to the file log.</td>
</tr>
<tr>
<td>fileLogFilename</td>
<td>The file name to log events to.</td>
</tr>
<tr>
<td>anonymousUserLogging</td>
<td>Whether events from machines within this group will be logged with anonymous users.</td>
</tr>
<tr>
<td>anonymousMachineLogging</td>
<td>Whether events from machines within this group will be logged with anonymous machines.</td>
</tr>
<tr>
<td>overrideServerUrls</td>
<td>Whether this group overrides server URL’s.</td>
</tr>
</tbody>
</table>
**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.

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

Deletes an existing group.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The key which identifies the group.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the group was last modified.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

```csharp
```

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

**Returns:**

Data set containing the default group in the groups table.

```csharp
```

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.

```csharp
```
Returns an ordered list of all the group names.

**Returns:**
Array of DeploymentGroupDto's.

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

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.GetGroupFromKey (Guid groupKey, Boolean withSummary)`

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)`

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 (Boolean withSummary)`

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


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


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


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)**

Removes an existing package from a group.

Parameters:

<p>| groupKey | The key of the group. |</p>
<table>
<thead>
<tr>
<th><strong>packageKey</strong></th>
<th>The key of the package.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>modifiedTime</strong></td>
<td>The date time the group package row was last modified.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

```csharp
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.

```csharp
void DataAccessServices.WebServices.Groups.UpdateEventFilter (Guid groupKey, Int32 eventDefinitionKey, Boolean 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.

```csharp
```

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. |
<table>
<thead>
<tr>
<th>anonymousUserLogging</th>
<th>Whether events from machines within this group will be logged with anonymous users.</th>
</tr>
</thead>
<tbody>
<tr>
<td>anonymousMachineLogging</td>
<td>Whether events from machines within this group will be logged with anonymous machines.</td>
</tr>
<tr>
<td>overrideServerUrls</td>
<td>Whether this group overrides server URL’s.</td>
</tr>
<tr>
<td>policyKey</td>
<td>Policy associated with this group.</td>
</tr>
<tr>
<td>ownerSid</td>
<td>SID of user owning this group.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The time that the group was modified.</td>
</tr>
<tr>
<td>selfRegistrationEnabled</td>
<td>Whether or not self registration is allowed.</td>
</tr>
<tr>
<td>selfUnregistrationEnabled</td>
<td>Whether or not self unregistration is allowed.</td>
</tr>
<tr>
<td>selfUpdateEnabled</td>
<td>Whether or not self update of agents and configurations is allowed.</td>
</tr>
<tr>
<td>priority</td>
<td>Order in which this groups membership rules will be evaluated.</td>
</tr>
<tr>
<td>pollPeriodVariationSeconds</td>
<td>The VariationSeconds allowed for each poll.</td>
</tr>
<tr>
<td>uploadPollPeriodVariationSeconds</td>
<td>The VariationSeconds allowed for each upload.</td>
</tr>
<tr>
<td>nativeConfigurations</td>
<td>Whether configurations be deployed to this group in native format.</td>
</tr>
<tr>
<td>configurationLocation</td>
<td>Location on agent machine to store native configurations.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

```csharp
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.

```csharp
void DataAccessServices.WebServices.Groups.UpdateGroupPackage (Guid groupKey, Guid packageKey, Guid? patchKey, Int32 major, Int32 minor, Int32 build, Int32 revision, ref DateTime modifiedTime) [inline]
```

Updates the version of a package within a group.

**Parameters:**

<p>| 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. |</p>
<table>
<thead>
<tr>
<th>revision</th>
<th>Revision of package version to use.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>Time the group package was last modified.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

```csharp
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 associated with this group.
- `ownerSid` SID of user owning this group.
- `modifiedTime` The time that the group was last modified.

```csharp
void DataAccessServices.WebServices.Groups.UpdateInstallationSchedule (Guid groupKey, Int64 mondayFlags, Int64 tuesdayFlags, Int64 wednesdayFlags, Int64 thursdayFlags, Int64 fridayFlags, Int64 saturdayFlags, Int64 sundayFlags, Int32 flags, Boolean doNotInstallAgent, ref DateTime modifiedTime, Int64 mondayConfigFlags, Int64 tuesdayConfigFlags, Int64 wednesdayConfigFlags, Int64 thursdayConfigFlags, Int64 fridayConfigFlags, Int64 saturdayConfigFlags, Int64 sundayConfigFlags, Int32 configFlags, Boolean doNotInstallConfig, Boolean mirrorAgentSchedule, Boolean postponeInstallationPromptEnabled, Int32 postponeInstallationLimit, Boolean downloadConfigurationsOnStartup, Boolean midSessionUpdate, Boolean 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
<table>
<thead>
<tr>
<th>gs</th>
<th>The flags for Sunday Config</th>
</tr>
</thead>
<tbody>
<tr>
<td>sundayConfigFlags</td>
<td>The global config flags</td>
</tr>
<tr>
<td>doNotInstallConfig</td>
<td>Indicates the config installation schedule has been disabled</td>
</tr>
<tr>
<td>mirrorAgentSchedule</td>
<td>Indicates the config schedule should mirror the agent schedule</td>
</tr>
<tr>
<td>postponeInstallationPromptEnabled</td>
<td>Set for postpone prompt for scheduled installs</td>
</tr>
<tr>
<td>postponeInstallationLimit</td>
<td>Max limit for postponement in seconds</td>
</tr>
<tr>
<td>downloadConfigurationsOnStartup</td>
<td>Download and install configurations before signalling agents to start</td>
</tr>
<tr>
<td>midSessionUpdate</td>
<td>Whether non-rebooting packages can be installed or uninstalled on computers mid-session.</td>
</tr>
<tr>
<td>updateAtShutdown</td>
<td>Whether packages can be installed or uninstalled on computers at shutdown.</td>
</tr>
</tbody>
</table>

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

Inheritance diagram for DataAccessServices.WebServices.Housekeeping:

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.

Inheritance diagram for DataAccessServices.WebServices.Licenses:

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 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.

- LicensingV2DataSet GetV2Licenses ()
  Returns a data set containing all V2 licenses.

- void SetLicenses (DataSet ds)

- 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

```csharp
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:**

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

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

Adds a license using the new V2 license schema.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>licensingKey</td>
<td>String containing the license encrypted using a base 64 encryption.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the license on return. Passed in value unused.</td>
</tr>
</tbody>
</table>

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

Updates the database with the changes in the data set.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>licenseChanges</td>
<td>A data set containing changes to licenses.</td>
</tr>
</tbody>
</table>
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. |


Returns a data set containing all licenses.

**Returns:**

Data set containing all licenses.


Returns a data set containing all V2 licenses.

**Returns:**

Data set containing all V2 licenses.

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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>licenseKey</td>
<td>GUID identifying the license.</td>
</tr>
<tr>
<td>licenseCode</td>
<td>License code for this license.</td>
</tr>
<tr>
<td>activationCode</td>
<td>Activation code for this license.</td>
</tr>
<tr>
<td>licenseType</td>
<td>Type of license.</td>
</tr>
<tr>
<td>productKey</td>
<td>Product GUID this license applies to.</td>
</tr>
<tr>
<td>licenseCount</td>
<td>Number of machines.</td>
</tr>
<tr>
<td>expiryDate</td>
<td>Expiry date for this license.</td>
</tr>
<tr>
<td>baseLicense</td>
<td>Whether or not this license is a base license.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Time this license was last modified.</td>
</tr>
</tbody>
</table>

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.

Inheritance diagram for DataAccessServices.WebServices.Machines:

Public Member Functions

- void AddMachineDetails(Guid machineKey, String name, String value)
  Adds some detailed description about a machine.

  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, Boolean withSummary)
  Returns a machine from a given fully qualified domain name.

• MachinesDataSet GetFromGroupKey (Guid groupKey, Boolean withSummary)
  Returns all machines within a specific group.

• MachinesDataSet GetFromGroupKeyDelta (Guid consoleId, Guid groupKey, Boolean withSummary, ref DateTime?lastRefresh)
  Returns all changed machines within a specific group since the last refresh.

• MachinesDataSet GetFromKey (Guid machineKey, Boolean withSummary)
  Returns a single machine.

• MachinesDataSet GetFromObjectGuid (Guid objectGuid, Boolean 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, Boolean 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 (Boolean withSummary)
  Returns all machines

• MachinesDataSet GetMachinesDelta (Guid consoleId, Boolean withSummary, ref DateTime?lastRefresh)
  Returns all machines changed since the last refresh

• MachinesDataSet GetPendingDeletion (Boolean withSummary)
  Returns all machines which have the IsPendingDeletion flag set to true.

• MachinesDataSet GetPreloadFromGroupKey (Guid groupKey, Int32 preloadAmount)
  Returns the specified number of machines from the start of the Machines table belonging to the specified group.

• MachinesDataSet GetPreloadMachines (Int32 preloadAmount)
  Returns the specified number of machines from the start of the Machines table

• MachinesDataSet GetWithObjectGuid (Boolean 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, Boolean isPendingDeletion, Boolean 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, Boolean diagnosticsError, MachineDiagnosticsState diagnosticsState, out DateTime modifiedTime)
  Updates an existing machine.

  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**


Adds some detailed description about a machine.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>machineKey</td>
<td>The key that identifies the machine.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the descriptive information.</td>
</tr>
<tr>
<td>value</td>
<td>The value of the descriptive information.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

Associates a package with a machine.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>machineKey</code></td>
<td>The key of the machine.</td>
</tr>
<tr>
<td><code>packageKey</code></td>
<td>The key of the package.</td>
</tr>
<tr>
<td><code>name</code></td>
<td>The name of the package.</td>
</tr>
<tr>
<td><code>company</code></td>
<td>The company that supplied the package.</td>
</tr>
<tr>
<td><code>type</code></td>
<td>The type of the package, such as software or config.</td>
</tr>
<tr>
<td><code>platform</code></td>
<td>The platform of the package.</td>
</tr>
<tr>
<td><code>version</code></td>
<td>The version number of the package.</td>
</tr>
<tr>
<td><code>status</code></td>
<td>The status.</td>
</tr>
<tr>
<td><code>childVersion</code></td>
<td>The child version.</td>
</tr>
<tr>
<td><code>childStatus</code></td>
<td>The child status.</td>
</tr>
<tr>
<td><code>patchVersion</code></td>
<td>The version number of the patch.</td>
</tr>
<tr>
<td><code>patchCode</code></td>
<td>The patch code.</td>
</tr>
<tr>
<td><code>patchStatus</code></td>
<td>The patch status.</td>
</tr>
<tr>
<td><code>patchChildVersion</code></td>
<td>The patch child version.</td>
</tr>
<tr>
<td><code>patchChildCode</code></td>
<td>The patch child code.</td>
</tr>
<tr>
<td><code>patchChildStatus</code></td>
<td>The patch child status.</td>
</tr>
<tr>
<td><code>statusMessage</code></td>
<td>The status message.</td>
</tr>
<tr>
<td><code>modifiedTime</code></td>
<td>OUT parameter will contain the latest modified time of the license on return.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

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

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>machineChanges</code></td>
<td>The data set consisting of the changes.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

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

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>machineChanges</code></td>
<td>The data set consisting of the changes.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.

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.


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:**

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

Requires deployment or administrative access.

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

Deletes an existing machine.

**Parameters:**

<table>
<thead>
<tr>
<th>machineKey</th>
<th>The machine to delete.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
</tbody>
</table>

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.

(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, Boolean 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, Boolean 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, Boolean withSummary, ref DateTime? lastRefresh)[inline]

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>consoleId</code></td>
<td>Guid to identify the console making the request.</td>
</tr>
<tr>
<td><code>groupKey</code></td>
<td>The group that the returned machines belong to.</td>
</tr>
<tr>
<td><code>withSummary</code></td>
<td>Whether to include summary information.</td>
</tr>
<tr>
<td><code>lastRefresh</code></td>
<td>DateTime of the last refresh.</td>
</tr>
</tbody>
</table>

**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, Boolean withSummary)` [inline]

Returns a single machine.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>machineKey</code></td>
<td>The machine to return.</td>
</tr>
<tr>
<td><code>withSummary</code></td>
<td>Whether to include summary information.</td>
</tr>
</tbody>
</table>

**Returns:**

A data set consisting of a single machine.
Requires deployment or administrative access.

**MachinesDataSet**

`DataAccessServices.WebServices.Machines.GetFromObjectGuid(Guid objectGuid, Boolean withSummary)` [inline]

Returns a machine from a given objectGUID attribute.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>objectGuid</code></td>
<td>The objectGUID attribute returned from the LDAP server.</td>
</tr>
<tr>
<td><code>withSummary</code></td>
<td>Whether to include summary information.</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>machineKey</code></td>
<td>The key which identifies the machine.</td>
</tr>
</tbody>
</table>

**Returns:**

A data set consisting of the machine details.
Requires deployment or administrative access.
(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, Boolean  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.

(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 (Boolean  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, Boolean 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 (Boolean 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, Int32 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 (Int32 preloadAmount)[inline]

Returns the specified number of machines from the start of the Machines table
**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>preloadAmount</td>
<td>Number of machines to get.</td>
</tr>
</tbody>
</table>

**Returns:**

Data set containing the machines

---

MachinesDataSet DataAccessServices.WebServices.Machines.GetWithObjectGuid (Boolean withSummary) [inline]

Returns all machines which contain an objectGUID.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>withSummary</td>
<td>Whether to include summary information.</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packageKey</td>
<td>GUID specifying a key for the associated package.</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packageKey</td>
<td>GUID specifying a key for the associated package.</td>
</tr>
<tr>
<td>version</td>
<td>The version number of the package.</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destinationGroupKey</td>
<td>Guid of the group to move the machines into.</td>
</tr>
</tbody>
</table>
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, Boolean  isPendingDeletion, Boolean  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. |
void DataAccessServices.WebServices.Machines.UpdateMachineDetails (Guid machineKey, String name, String value)[inline]

Updates some detailed description about a machine.

Parameters:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>machineKey</td>
<td>The key that identifies the machine.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the descriptive information.</td>
</tr>
<tr>
<td>value</td>
<td>The value of the descriptive information.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.


Updates an existing machine.

Parameters:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>machineKey</td>
<td>The key which identifies this machine.</td>
</tr>
<tr>
<td>diagnosticsError</td>
<td>If set to true show a diagnostics error has occurred.</td>
</tr>
<tr>
<td>diagnosticsState</td>
<td>State of the diagnostics.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the machine on return. Passed in value unused.</td>
</tr>
</tbody>
</table>

Requires deployment or administrative access.


Updates a package associated with a machine.

Parameters:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>machineKey</td>
<td>The key of the machine.</td>
</tr>
<tr>
<td>packageKey</td>
<td>The key of the package.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the package.</td>
</tr>
<tr>
<td>company</td>
<td>The company that supplied the package.</td>
</tr>
<tr>
<td>type</td>
<td>The type of the package, such as software or config.</td>
</tr>
<tr>
<td>platform</td>
<td>The platform of the package.</td>
</tr>
<tr>
<td>version</td>
<td>The version number of the package.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>status</td>
<td>The status.</td>
</tr>
<tr>
<td>childVersion</td>
<td>The child version.</td>
</tr>
<tr>
<td>childStatus</td>
<td>The child status.</td>
</tr>
<tr>
<td>patchVersion</td>
<td>The version number of the patch.</td>
</tr>
<tr>
<td>patchCode</td>
<td>The patch code.</td>
</tr>
<tr>
<td>patchStatus</td>
<td>The patch status.</td>
</tr>
<tr>
<td>patchChildVersion</td>
<td>The patch child version.</td>
</tr>
<tr>
<td>patchChildCode</td>
<td>The patch child code.</td>
</tr>
<tr>
<td>patchChildStatus</td>
<td>The patch child status.</td>
</tr>
<tr>
<td>statusMessage</td>
<td>The status message.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Datetime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
</tbody>
</table>

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 events/alerts that can be raised via the product agents.

Inheritance diagram for DataAccessServices.WebServices.Maintenance:

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)

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

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

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

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

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

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

- 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 deployment group.

- MaintenanceHelper.PreviewEventsDto Maintenance_RemoveAllEventsOlderThanXDdays (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_RemoveAllHighVolumeEventsOlderThanXDdays (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 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_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 deployment 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_RemoveAllOrphanedAlerts' 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 events/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 |


Returns a dataset of all Scheduled maintenance actions

Returns:

Requires administrative access.

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>jobId</code></td>
<td>A unique job id created for the ScheduledMaintenance row.</td>
</tr>
<tr>
<td><code>activeServer</code></td>
<td>The server that is executing the job.</td>
</tr>
<tr>
<td><code>result</code></td>
<td>This is a return value if the task was successfully locked.</td>
</tr>
</tbody>
</table>

**MaintenanceHelper.PreviewEventsDto**

DataAccessServices.WebServices.Maintenance.Maintenance_RemoveAllEventsFromDeploymentGroup (string `deploymentGroup`, long `batchSize`, long `batchDelay`) [inline]

Removes all product events from the specified deployment group.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>deploymentGroup</code></td>
<td>The deployment group at which to target deletion action.</td>
</tr>
<tr>
<td><code>batchSize</code></td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).</td>
</tr>
<tr>
<td><code>batchDelay</code></td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>daysToRetain</code></td>
<td>The number of days worth of events to keep after this operation completes.</td>
</tr>
<tr>
<td><code>batchSize</code></td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).</td>
</tr>
<tr>
<td><code>batchDelay</code></td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>eventId</code></td>
<td>The product event ID for which all records are to be deleted.</td>
</tr>
<tr>
<td><code>batchSize</code></td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).</td>
</tr>
<tr>
<td><code>batchDelay</code></td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>
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:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startEventId</td>
<td>The lowest product event ID for which all records are to be deleted.</td>
</tr>
<tr>
<td>endEventId</td>
<td>The highest product event ID for which all records are to be deleted.</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

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:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>daysToRetain</td>
<td>The number of days worth of high volume events that should remain after this operation.</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

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:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations (Optional).</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>
Returns:

PreviewAlertsDto
Requires administrative access.

```csharp
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.

```csharp
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.

```csharp
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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobId</td>
<td>A unique job id created for the ScheduledMaintenance row.</td>
</tr>
<tr>
<td>timeCompleted</td>
<td>The time that the task is completed.</td>
</tr>
<tr>
<td>result</td>
<td>This is a return value if the task was successfully unlocked.</td>
</tr>
</tbody>
</table>

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>jobId</strong></td>
<td>Job id of scheduled task to update.</td>
</tr>
<tr>
<td><strong>groupName</strong></td>
<td>Name of deployment group for events to be removed by.</td>
</tr>
<tr>
<td><strong>description</strong></td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td><strong>dayOfWeek</strong></td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td><strong>startTime</strong></td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td><strong>batchSize</strong></td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td><strong>batchDelay</strong></td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

Requires administrative access.

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>jobId</strong></td>
<td>Job id of scheduled task to update.</td>
</tr>
<tr>
<td><strong>daysToRetain</strong></td>
<td>The number of days worth of events that should remain after this operation.</td>
</tr>
<tr>
<td><strong>description</strong></td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td><strong>dayOfWeek</strong></td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td><strong>startTime</strong></td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td><strong>batchSize</strong></td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td><strong>batchDelay</strong></td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

Requires administrative access.

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>jobId</strong></td>
<td>Job id of scheduled task to update.</td>
</tr>
<tr>
<td>startEventId</td>
<td>The product event ID for which all records are to be deleted.</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td>dayOfWeek</td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td>startTime</td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

Requires administrative access.

```csharp
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:**

<table>
<thead>
<tr>
<th>jobId</th>
<th>Job id of scheduled task to update.</th>
</tr>
</thead>
<tbody>
<tr>
<td>startEventId</td>
<td>The lowest product event ID for which all records are to be deleted.</td>
</tr>
<tr>
<td>endEventId</td>
<td>The highest product event ID for which all records are to be deleted.</td>
</tr>
<tr>
<td>description</td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td>dayOfWeek</td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td>startTime</td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

Requires administrative access.

```csharp
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:**

<table>
<thead>
<tr>
<th>jobId</th>
<th>Job id of scheduled task to update.</th>
</tr>
</thead>
<tbody>
<tr>
<td>daysToRetain</td>
<td>The number of days worth of high volume events that should remain after this operation.</td>
</tr>
<tr>
<td>description</td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td>dayOfWeek</td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td>startTime</td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobId</td>
<td>Job id of scheduled task to update.</td>
</tr>
<tr>
<td>description</td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td>dayOfWeek</td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td>startTime</td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

Requires administrative access.

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

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobId</td>
<td>Job id of scheduled task to update.</td>
</tr>
<tr>
<td>description</td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td>dayOfWeek</td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td>startTime</td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobId</td>
<td>Job id of scheduled task to update.</td>
</tr>
<tr>
<td>daysToRetain</td>
<td>The number of days worth of events to keep after this operation completes.</td>
</tr>
<tr>
<td>groupName</td>
<td>Name of deployment group for events to be removed by.</td>
</tr>
<tr>
<td>description</td>
<td>Description for the scheduled task.</td>
</tr>
<tr>
<td>dayOfWeek</td>
<td>Day of week the scheduled task is to run</td>
</tr>
<tr>
<td>startTime</td>
<td>Time of day for schedule task to start</td>
</tr>
<tr>
<td>maxRunHours</td>
<td>Maximum run time</td>
</tr>
<tr>
<td>batchSize</td>
<td>How many rows to delete at a time, so as to avoid locking db data during potentially lengthy operations</td>
</tr>
<tr>
<td>batchDelay</td>
<td>The delay between batches</td>
</tr>
</tbody>
</table>
**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 deployment 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)

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)

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)

Provides preview results of the 'Maintenance_RemoveAllOrphanedAlerts' 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:
(Nothing).

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.

Inheritance diagram for DataAccessServices.WebServices.Packages:

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 BeginPackageVersionDownload(Guid packageKey, Int32 major, Int32 minor, Int32 build, Int32 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 CommitPackageVersion(Guid packageVersionKey)
  

132
Commits the package version.

- void **CommitPatch** (Guid patchKey)
  Commits a patch to the database.

- Byte[] **ContinuePackageVersionDownload** (Guid downloadKey, Int32 offset, Int32 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, Int32 offset, Byte[] data)
  Continues an upload of a package version.

- Byte[] **ContinuePatchDownload** (Guid downloadKey, Int32 offset, Int32 length)
  Continues the patch download.

- void **ContinuePatchUpload** (Guid patchKey, ref DateTime modifiedTime, Guid uploadKey, Int32 offset, Byte[] data)
  Continues the patch upload.

- Byte[] **ContinuePrerequisiteResourceDownload** (Guid downloadKey, Int32 offset, Int32 length)
  Continues the prerequisite resource download.

- void **ContinuePrerequisiteResourceUpload** (Guid resourceKey, ref DateTime modifiedTime, Guid uploadKey, Int32 offset, Byte[] data)
  Continues the prerequisite resource upload.

- void **CreatePackage** (Guid key, String company, String type, PackagePlatform platform, Guid productKey, out DateTime modifiedTime)
  Creates a configuration package within the database.

  Creates the package version.

- void **CreatePatch** (Guid patchKey, Guid patchCode, Guid packageVersionKey, String name, Int32 major, Int32 minor, Int32 build, Int32 revision, string marketingVersion, Int32 targetMajor, Int32 targetMinor, Int32 targetBuild, Int32 targetRevision, Int32 validationFlags, string description, Boolean 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 prerequisiteKey, DateTime? modifiedTime)
  Deletes the prerequisite.

- void **FinalisePackageVersion** (Guid packageVersionKey)
  Finalises the package version.

- PackagesDataSet **GetAllDependentPatchesFromPatchKey** (Guid patchKey)
  Returns all dependents for a given patch.

- 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)

133
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:**

<table>
<thead>
<tr>
<th>key</th>
<th>The key that identifies the package to return.</th>
</tr>
</thead>
</table>

- 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.
- `GetInt32` `GetPackageVersionLength` (Guid packageKey, Int32 major, Int32 minor, Int32 build, Int32 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.
- `GetInt32` `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 `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 separate 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, Int32 major, Int32 minor, Int32 build, Int32 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 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 that this method currently only supports deletion of packages and package versions.

**Parameters:**

| packageChanges | The package changes. |

Applies package changes from a packages data set. Note that this method currently only supports deletion of packages and package versions.

**Parameters:**

| packageChanges | The package changes. |

```csharp
void DataAccessServices.WebServices.Packages.ApplyPrerequisiteChanges(ref PrerequisitesDataSet prerequisiteChanges)[inline]
```

Applies the prerequisite changes.

**Parameters:**

| prerequisiteChanges | The prerequisite changes. |

```csharp
```

Begins a package download.

**Parameters:**

<table>
<thead>
<tr>
<th>packageKey</th>
<th>The key which represents the package.</th>
</tr>
</thead>
<tbody>
<tr>
<td>major</td>
<td>The major version of the package.</td>
</tr>
<tr>
<td>minor</td>
<td>The minor version of the package.</td>
</tr>
<tr>
<td>build</td>
<td>The build version of the package.</td>
</tr>
<tr>
<td>revision</td>
<td>The revision version of the package.</td>
</tr>
</tbody>
</table>

**Returns:**

A guid which represents a download key.

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
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:**

<table>
<thead>
<tr>
<th>packageKey</th>
<th>The key which represents the package.</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>Description of the specified item.</td>
</tr>
<tr>
<td>packageVersionKey</td>
<td>The key which uniquely identifies this package.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
<tr>
<td>dataLength</td>
<td>The length of data that will be uploaded.</td>
</tr>
</tbody>
</table>
**Returns:**
An upload key used to add data to the upload.
Caution: 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:**

<table>
<thead>
<tr>
<th>resourceKey</th>
<th>The resource key.</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataLength</td>
<td>Length of the data.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
</tbody>
</table>

**Returns:**

An upload key used to add data to the upload.

```csharp
```

Commits the package version.

**Parameters:**

| packageVersionKey | The package version key. |

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
void DataAccessServices.WebServices.Packages.CommitPatch (Guid patchKey)[inline]
```

Commits a patch to the database.

**Parameters:**

| patchKey | The patch key |

```csharp
Byte [] DataAccessServices.WebServices.Packages.ContinuePackageVersionDownload (Guid downloadKey, Int32 offset, Int32 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.

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
void DataAccessServices.WebServices.Packages.ContinuePackageVersionUpload (Guid packageVersionKey, ref DateTime modifiedTime, Guid uploadKey, Int32 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.

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
Byte[] DataAccessServices.WebServices.Packages.ContinuePatchDownload(Guid downloadKey, Int32 offset, Int32 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.

```csharp
void DataAccessServices.WebServices.Packages.ContinuePatchUpload(Guid patchKey, ref DateTime modifiedTime, Guid uploadKey, Int32 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.

```csharp
Byte[] DataAccessServices.WebServices.Packages.ContinuePrerequisiteResourceDownload(Guid downloadKey, Int32 offset, Int32 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.

```csharp
void DataAccessServices.WebServices.Packages.ContinuePrerequisiteResourceUpload (Guid resourceKey, ref DateTime modifiedTime, Guid uploadKey, Int32 offset, Byte[] data)[inline]
```

Continues the prerequisite resource upload.

**Parameters:**

<table>
<thead>
<tr>
<th>resourceKey</th>
<th>The resource key.</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
<tr>
<td>uploadKey</td>
<td>The upload key.</td>
</tr>
<tr>
<td>offset</td>
<td>The offset.</td>
</tr>
<tr>
<td>data</td>
<td>The data.</td>
</tr>
</tbody>
</table>

```csharp
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:**

<table>
<thead>
<tr>
<th>key</th>
<th>The Guid which identifies the package.</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>The company name.</td>
</tr>
<tr>
<td>type</td>
<td>The type of the package, i.e. Software, Configuration.</td>
</tr>
<tr>
<td>platform</td>
<td>The platform that the package should be installed on.</td>
</tr>
<tr>
<td>productKey</td>
<td>The name of the product associated with this package.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>A date time indicating when the package was created.</td>
</tr>
</tbody>
</table>

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
```

Creates the package version.

**Parameters:**

<table>
<thead>
<tr>
<th>packageKey</th>
<th>GUID specifying a key for the associated package.</th>
</tr>
</thead>
<tbody>
<tr>
<td>packageVersionKey</td>
<td>The package version key.</td>
</tr>
</tbody>
</table>
**name**  
Name of the specified item.

**major**  
The major.

**minor**  
The minor.

**build**  
The build.

**revision**  
The revision.

**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.

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
```

Creates a patch.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>patchKey</code></td>
<td>GUID specifying a key for the associated package</td>
</tr>
<tr>
<td><code>patchCode</code></td>
<td>Patch Code property from MSP file</td>
</tr>
<tr>
<td><code>packageVersionKey</code></td>
<td>The package version key</td>
</tr>
<tr>
<td><code>name</code></td>
<td>Name of the specified item</td>
</tr>
<tr>
<td><code>major</code></td>
<td>The major</td>
</tr>
<tr>
<td><code>minor</code></td>
<td>The minor</td>
</tr>
<tr>
<td><code>build</code></td>
<td>The build</td>
</tr>
<tr>
<td><code>revision</code></td>
<td>The revision</td>
</tr>
<tr>
<td><code>targetMajor</code></td>
<td>The major version of the patch or package that this patch updates</td>
</tr>
<tr>
<td><code>targetMinor</code></td>
<td>The minor version of the patch or package that this patch updates</td>
</tr>
<tr>
<td><strong>targetBuild</strong></td>
<td>The build version of the patch or package that this patch updates</td>
</tr>
<tr>
<td><strong>targetRevision</strong></td>
<td>The revision version of the patch or package that this patch updates</td>
</tr>
<tr>
<td><strong>validationFlags</strong></td>
<td>The Validation Flags property from the Windows Installer MSP file</td>
</tr>
<tr>
<td><strong>description</strong></td>
<td>Description of the specified item.</td>
</tr>
<tr>
<td><strong>supportsMidSessionUpdate</strong></td>
<td>This defines whether an agent can be installed without requiring a reboot</td>
</tr>
<tr>
<td><strong>modifiedTime</strong></td>
<td>Time that the package version was created.</td>
</tr>
</tbody>
</table>

```csharp
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 |

```csharp
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. |

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
void DataAccessServices.WebServices.Packages.DeletePrerequisite (Guid prerequisiteKey, DateTime? modifiedTime)[inline]
```

Deletes the prerequisite.

**Parameters:**

| prerequisiteKey | The prerequisite key. |
| modifiedTime | DateTime specifying when this item was last modified by the application - used for concurrency purposes. |

```csharp
```

Finalises the package version.

**Parameters:**

| packageVersionKey | The package version key. |
Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

**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.

**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.

Caution: 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 with in progress.

**Parameters:**

| key    | The key that identifies the package to return. |

**Returns:**

A data set describing the packages and its versions.

Caution: 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)`

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)`

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()`

Returns all packages and their versions stored within the database.

**Returns:**

A data set consisting of all packages and versions.
Caution: 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()`

Gets the packages with in progress.
**Returns:**

A data set describing the packages and its versions.

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.


Determines the length of a version of a package.

**Parameters:**

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packageKey</td>
<td>The key which represents the package.</td>
</tr>
<tr>
<td>major</td>
<td>The major version of the package.</td>
</tr>
<tr>
<td>minor</td>
<td>The minor version of the package.</td>
</tr>
<tr>
<td>build</td>
<td>The build version of the package.</td>
</tr>
<tr>
<td>revision</td>
<td>The revision version of the package.</td>
</tr>
</tbody>
</table>

**Returns:**

The length of data in bytes.

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.


Gets the patches with in progress.

**Returns:**

A dataset consisting of all dependent patches.


Gets existing patches that need updating with additional metadata.

**Returns:**

A dataset consisting of all dependent patches.

PackagesDataSet DataAccessServices.WebServices.Packages.GetPatchFromPatchKey (Guid *patchKey*)

Returns all dependent patches for a given patch.

**Parameters:**

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>patchKey</td>
<td>The key which represents the patch.</td>
</tr>
</tbody>
</table>
Returns:
A dataset consisting of all dependent patches.

**PackagesDataSet**
\`
\`

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.

**Int32**
\`
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.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. |

Caution: This method is also defined in the PackageManagement Endpoint. Developers should use this alternative endpoint whenever possible.

```csharp
```

The CommitPackageVersion launches a separate 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 |

```csharp
```

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 |

```csharp
```

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. |

```csharp
void DataAccessServices.WebServices.Packages.RemovePatch (Guid patchKey, DateTime? modifiedTime)[inline]
```

Deletes the patch.
Parameters:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>patchKey</td>
<td>The patch key</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
</tbody>
</table>

void DataAccessServices.WebServices.Packages.UnlockPackage (Guid packageKey)[inline]

Unlocks the package.

Parameters:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packageKey</td>
<td>GUID specifying a key for the associated package.</td>
</tr>
</tbody>
</table>

Caution: 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:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>The Guid which identifies the package.</td>
</tr>
<tr>
<td>company</td>
<td>The company name.</td>
</tr>
<tr>
<td>type</td>
<td>The type of the package, i.e. Software, Configuration.</td>
</tr>
<tr>
<td>platform</td>
<td>The platform that the package should be installed on.</td>
</tr>
<tr>
<td>productKey</td>
<td>The name of the product associated with this package.</td>
</tr>
<tr>
<td>policyKey</td>
<td>The policy key.</td>
</tr>
<tr>
<td>ownerSid</td>
<td>The owner sid.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
</tbody>
</table>

void DataAccessServices.WebServices.Packages.UpdatePackageSecurity (Guid key, Guid? policyKey, String ownerSid, ref DateTime modifiedTime)[inline]

Updates the package security.

Parameters:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>The Guid which identifies the package.</td>
</tr>
<tr>
<td>policyKey</td>
<td>The policy key.</td>
</tr>
<tr>
<td>ownerSid</td>
<td>The owner sid.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
</tbody>
</table>

void DataAccessServices.WebServices.Packages.WIPSaved (Guid packageKey)[inline]
Marks a package as a work-in-progress.

**Parameters:**

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>packageKey</code></td>
<td>The key which represents the package.</td>
</tr>
</tbody>
</table>

Caution: 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.

Inheritance diagram for DataAccessServices.WebServices.Products:

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:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>productChanges</td>
<td>The product changes.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>productKey</td>
<td>GUID specifying a key for the associated product.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the specified item.</td>
</tr>
<tr>
<td>icon</td>
<td>Byte array containing a Windows Icon file.</td>
</tr>
<tr>
<td>supportsAgents</td>
<td>Whether or not the product can support agents.</td>
</tr>
<tr>
<td>supportsConfigurations</td>
<td>Whether or not the product can support configuration files.</td>
</tr>
<tr>
<td>supportsSoftware</td>
<td>Whether or not the product can support software such as consoles.</td>
</tr>
<tr>
<td>highestVersionNumber</td>
<td></td>
</tr>
<tr>
<td>modifiedTime</td>
<td>A date time indicating when the product was created.</td>
</tr>
</tbody>
</table>

void DataAccessServices.WebServices.Products.DeleteProduct (Guid productKey, DateTime? modifiedTime)[inline]

Deletes a product.

Parameters:

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>productKey</td>
<td>GUID specifying a key for the associated product.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application -</td>
</tr>
<tr>
<td></td>
<td>used for concurrency purposes.</td>
</tr>
</tbody>
</table>


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)

Updates an existing product.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>productKey</td>
<td>GUID specifying a key for the associated product.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the specified item.</td>
</tr>
<tr>
<td>icon</td>
<td>Byte array containing a Windows Icon file.</td>
</tr>
<tr>
<td>supportsAgents</td>
<td>Whether or not the product can support agents.</td>
</tr>
<tr>
<td>supportsConfigurations</td>
<td>Whether or not the product can support configuration files.</td>
</tr>
<tr>
<td>supportsSoftware</td>
<td>Whether or not the product can support software such as consoles.</td>
</tr>
<tr>
<td>highestVersionNumber</td>
<td></td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application -</td>
</tr>
<tr>
<td></td>
<td>used for concurrency purposes.</td>
</tr>
</tbody>
</table>

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.

Inheritance diagram for DataAccessServices.WebServices.Queries:

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryString</td>
<td>The query to execute, in serialised XML.</td>
</tr>
<tr>
<td>timeout</td>
<td>The timeout before the query is cancelled (in milliseconds)</td>
</tr>
</tbody>
</table>

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)
**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**


Applies changes within a report definition data set.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportDefinitionChanges</td>
<td>The report definition changes.</td>
</tr>
</tbody>
</table>


Updates owner sid for report definitions specified in a report definition data set.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportDefinitionChanges</td>
<td>The report definition changes containing the owner sids to apply.</td>
</tr>
</tbody>
</table>

**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)

Creates a new report definition within the database. The report definition will only be available when data has been added.

#### Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportDefinitionKey</td>
<td>The report definition key.</td>
</tr>
<tr>
<td>name</td>
<td>Name of the specified item.</td>
</tr>
<tr>
<td>description</td>
<td>Description of the specified item.</td>
</tr>
<tr>
<td>category</td>
<td>The category.</td>
</tr>
<tr>
<td>type</td>
<td>The type.</td>
</tr>
<tr>
<td>productKey</td>
<td>GUID specifying a key for the associated product.</td>
</tr>
<tr>
<td>visible</td>
<td>If set to true report is visible.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>OUT parameter will contain the latest modified time of the report definition on return. Passed in value unused.</td>
</tr>
</tbody>
</table>


Creates the supplied report definitions within the database.

#### Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportDefinitionList</td>
<td>A list of report definitions to add.</td>
</tr>
<tr>
<td>compress</td>
<td>Whether or not the report definitions are compressed.</td>
</tr>
</tbody>
</table>

### void DataAccessServices.WebServices.Reports.DeleteReportDefinition (Guid reportDefinitionKey, DateTime? modifiedTime)

Deletes a report definition from the database.

#### Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reportDefinitionKey</td>
<td>Key of the report to delete.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>DateTime specifying when this item was last modified by the application - used for concurrency purposes.</td>
</tr>
</tbody>
</table>


Deletes the supplied report definitions from the database.
Parameters:

- `reportDefinitionList` | The report definition list.

`ReportDefinitionsDataSet` `DataAccessServices.WebServices.Reports.GetReportDefinition (Guid reportDefinitionKey)`

Returns a specific report definition from the database.

Parameters:

- `reportDefinitionKey` | The report definition key.

Returns:

- Data set containing specific report definition from the database.


Returns a list of all report definitions stored within the database.

Returns:

- Data set containing all report definitions stored within the database.


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)`

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. |

```csharp
void DataAccessServices.WebServices.Reports.UpdateReportDefinitionSecurity (Guid reportDefinitionKey, Guid? policyKey, String ownerSid, ref DateTime modifiedTime)
```

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

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.

Inheritance diagram for DataAccessServices.WebServices.Security:

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.
- Boolean CanLogin ()
  Determine if the current user can login to the database.
- Boolean CanLoginEx (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, Boolean 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, Boolean readOnly)
  Create a Server security role.
- void CreateUser (Guid userKey, String name, String sid, Boolean 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** (Boolean 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, Boolean 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**

```csharp
```

Apply changes made to a users data set into the database.

**Parameters:**

| changes | The data set consisting of the changes. |

```csharp
```

Apply changes made to a users data set into the database.

**Parameters:**

| changes | The data set consisting of the changes. |

```csharp
```

Apply changes made to the users data set.

**Parameters:**

| userChanges | The data set consisting of the changes. |

```csharp
```

Determine if the current user can login to the database.

**Returns:**

True if the user can login, otherwise False.

```csharp
```

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.
Create an Object security role.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>securityRoleKey</td>
<td>The key for the role.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the role.</td>
</tr>
<tr>
<td>description</td>
<td>The description for the role.</td>
</tr>
<tr>
<td>mask</td>
<td>The permissions for the role.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Returns the modified time of the role.</td>
</tr>
<tr>
<td>readOnly</td>
<td>Whether the role is read only.</td>
</tr>
</tbody>
</table>

Create a new policy.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyKey</td>
<td>The key for the new policy.</td>
</tr>
<tr>
<td>type</td>
<td>The type of the policy to create.</td>
</tr>
</tbody>
</table>

Create a security element.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>securityElementKey</td>
<td>The key for the new element.</td>
</tr>
<tr>
<td>policyKey</td>
<td>The policy key associated with this element.</td>
</tr>
<tr>
<td>elementType</td>
<td>The element type.</td>
</tr>
<tr>
<td>securityRoleKey</td>
<td>The security role associated with this element.</td>
</tr>
<tr>
<td>userSid</td>
<td>The security identifier of the user associated with this element.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Returns the modified time of the new element.</td>
</tr>
</tbody>
</table>

Create a Server security role.

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>securityRoleKey</td>
<td>The key for the role.</td>
</tr>
</tbody>
</table>
**void DataAccessServices.WebServices.Security.CreateUser (Guid userKey, String name, String sid, Boolean isGroup, out DateTime modifiedTime)**

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.


Delete a policy.

**Parameters:**

- **policyKey** The key of the policy to delete.


Delete a security element.

**Parameters:**

- **securityElementKey** The key of the element to delete.
- **modifiedTime** Modified time of the element.


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:**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userKey</td>
<td>The key which identifies the user.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Modified time of the user to delete, or Null.</td>
</tr>
</tbody>
</table>


Get all policy keys.

**Returns:**

A data set containing all policy keys.


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.


Get all security elements.

**Returns:**

A data set containing all security elements.


Get security elements with the specific policy key.

**Parameters:**

| policyKey | The policy key to find. |
**SecurityRolesDataSet**

DataAccessServices.WebServices.Security.GetSecurityRoleFromKey (Guid securityRoleKey)

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**


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)

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**


Get all server permissions for the current user.

**Returns:**

The server permissions for the current user.

**String**

Get the current user name.

**Returns:**

The current user name.

```csharp
GetUsers ()
```

Get all users.

**Returns:**

A data set containing all users.

```csharp
GetUsers ()
```

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. |

```csharp
UpdateSecurityRole (securityRoleKey, name, description, mask, modifiedTime, readOnly)
```

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. |

```csharp
UpdateUser (userKey, name, sid, modifiedTime)
```

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.

Inheritance diagram for DataAccessServices.WebServices.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


Update the database with the changes in the data set.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverChanges</td>
<td>A data set containing the server changes to apply.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverKey</td>
<td>The key of the new server.</td>
</tr>
<tr>
<td>groupKey</td>
<td>The group key this server will be added to.</td>
</tr>
<tr>
<td>url</td>
<td>The URL of the server.</td>
</tr>
<tr>
<td>index</td>
<td>The index of the server.</td>
</tr>
<tr>
<td>performDiagnostics</td>
<td>Whether the group can perform diagnostics.</td>
</tr>
<tr>
<td>disabled</td>
<td>Whether the group is disabled.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Returns the modified time of the new server.</td>
</tr>
</tbody>
</table>

void DataAccessServices.WebServices.Servers.DeleteServer (Guid serverKey, DateTime? modifiedTime)[inline]

Delete a server.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverKey</td>
<td>The key of the server to delete.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>Modified time of the server to delete.</td>
</tr>
</tbody>
</table>

ServersDataSet.DataAccessServices.WebServices.Servers.GetServers (Boolean includeDisabled)[inline]

Get all servers.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>includeDisabled</td>
<td>Whether to include disabled servers.</td>
</tr>
</tbody>
</table>
Returns:
A data set containing the servers.

ServersDataSet DataAccessServices.WebServices.Servers.GetServersFromGroupKey (Guid? groupKey, Boolean includeDisabled)[inline]

Get all servers for a group.

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupKey</td>
<td>The group key to retrieve.</td>
</tr>
<tr>
<td>includeDisabled</td>
<td>Whether to include disabled servers.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverKey</td>
<td>The key of the server to update.</td>
</tr>
<tr>
<td>groupKey</td>
<td>The group key for the server.</td>
</tr>
<tr>
<td>url</td>
<td>The URL for the server.</td>
</tr>
<tr>
<td>index</td>
<td>The index of the server.</td>
</tr>
<tr>
<td>performDiagnostics</td>
<td>Whether the server should be enabled or disabled for diagnostics.</td>
</tr>
<tr>
<td>disabled</td>
<td>Whether the server should be enabled or disabled.</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>The modified time of the server. Returns the new modified time of the server.</td>
</tr>
</tbody>
</table>

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

- Servers.cs
Index

ActivateDeploymentService
  DataAccessServices::WebServices::Deployment, 63
AddActionConfiguration
  DataAccessServices::WebServices::Alerts, 44
AddEventDateTimeParam
  DataAccessServices::WebServices::Events, 82
AddEventDefinitionParam
  DataAccessServices::WebServices::Events, 82
AddEventIntegerParam
 .DataAccessServices::WebServices::Events, 82
AddEventStringParam
  DataAccessServices::WebServices::Events, 82
AddEventToAlert
  DataAccessServices::WebServices::Alerts, 44
AddGroupLatestPackage
  DataAccessServices::WebServices::Groups, 91
AddGroupPackage
  DataAccessServices::WebServices::Groups, 91
AddGroupPatch
  DataAccessServices::WebServices::Groups, 92
AddLicense
  DataAccessServices::WebServices::Licenses, 103
AddLicenseV2
  DataAccessServices::WebServices::Licenses, 103
AddMachineDetails
  DataAccessServices::WebServices::Machines, 108
AddMachinePackage
  DataAccessServices::WebServices::Machines, 108
ApplyActionChanges
  DataAccessServices::WebServices::Alerts, 45
ApplyAlertChanges
  DataAccessServices::WebServices::Alerts, 45
ApplyAlertRuleChanges
  DataAccessServices::WebServices::Alerts, 45
ApplyAlertRuleSecurityChanges
  DataAccessServices::WebServices::Alerts, 45
ApplyAndUpdateMachines
  DataAccessServices::WebServices::Machines, 109
ApplyAndUpdateMachines
  DataAccessServices::WebServices::Machines, 109
ApplyChanges
  DataAccessServices::WebServices::Deployment, 64
  DataAccessServices::WebServices::Licenses, 103
  DataAccessServices::WebServices::Products, 153
  DataAccessServices::WebServices::Servers, 172
ApplyEventChanges
  DataAccessServices::WebServices::Events, 83
ApplyEventDefinitionChanges
  DataAccessServices::WebServices::Events, 83
ApplyEventFilterChanges
  DataAccessServices::WebServices::Groups, 92
ApplyGroupChanges
  DataAccessServices::WebServices::Groups, 92
ApplyGroupPackagesChanges
  DataAccessServices::WebServices::Groups, 92
ApplyGroupSecurityChanges
  DataAccessServices::WebServices::Groups, 93
ApplyInstallationScheduleChanges
  DataAccessServices::WebServices::Groups, 93
ApplyInstructionsChanges
  DataAccessServices::WebServices::Deployment, 64
ApplyInstructionsStatusChanges
  DataAccessServices::WebServices::Deployment, 64
ApplyMachineChanges
  DataAccessServices::WebServices::DiscoveredMachines, 71
  DataAccessServices::WebServices::Machines, 109
ApplyMachineDetailsChanges
  DataAccessServices::WebServices::Machines, 110
ApplyMachineDiagnosticsStateChanges
  DataAccessServices::WebServices::Machines, 110
ApplyMachinePackageChanges
  DataAccessServices::WebServices::Machines, 110
ApplyPackageChanges
  DataAccessServices::WebServices::Packages, 135
ApplyPackageSecurityChanges
  DataAccessServices::WebServices::Packages, 135
ApplyPrerequisiteChanges
  DataAccessServices::WebServices::Packages, 136
ApplyReportDefinitionChanges
  DataAccessServices::WebServices::Reports, 158
ApplyReportDefinitionSecurityChanges
  DataAccessServices::WebServices::Reports, 158
ApplySecurityElementChanges
  DataAccessServices::WebServices::Security, 165
ApplySecurityRoleChanges
  DataAccessServices::WebServices::Security, 165
ApplyUserChanges
  DataAccessServices::WebServices::Security, 165
AutoMove
  DataAccessServices::WebServices::Security, 165
BeginPackageVersionDownload
  DataAccessServices::WebServices::Packages, 136
BeginPackageVersionUpload
  DataAccessServices::WebServices::Packages, 136
BeginPatchDownload
  DataAccessServices::WebServices::Packages, 137
BeginPatchUpload
  DataAccessServices::WebServices::Packages, 137
BeginPrerequisiteResourceDownload
  DataAccessServices::WebServices::Packages, 137
BeginPrerequisiteResourceUpload
  DataAccessServices::WebServices::Packages, 137
BeginReportDefinitionDownload
  DataAccessServices::WebServices::Reports, 158
BeginReportDefinitionUpload
  DataAccessServices::WebServices::Reports, 159
CancelSchedule
  DataAccessServices::WebServices::Maintenance, 123
CanLogin
  DataAccessServices::WebServices::Security, 165
CanLoginEx
  DataAccessServices::WebServices::Security, 165
CheckExpectedGroupPermissions
  DataAccessServices::WebServices::DiscoveredMachines, 71
CheckMachineGroupPermissions
  DataAccessServices::WebServices::Machines, 110
ClearStatusHistory
  DataAccessServices::WebServices::Deployment, 64
CommitPackageVersion
  DataAccessServices::WebServices::Packages, 138
CommitPatch
  DataAccessServices::WebServices::Packages, 138
ContinuePackageVersionDownload
  DataAccessServices::WebServices::Packages, 138
ContinuePackageVersionUpload
  DataAccessServices::WebServices::Packages, 138
ContinuePatchDownload
  DataAccessServices::WebServices::Packages, 139
ContinuePatchUpload
  DataAccessServices::WebServices::Packages, 139
ContinuePrerequisiteResourceDownload
  DataAccessServices::WebServices::Packages, 139
ContinuePrerequisiteResourceUpload
  DataAccessServices::WebServices::Packages, 140
ContinueReportDefinitionDownload
  DataAccessServices::WebServices::Reports, 159
ContinueReportDefinitionUpload
  DataAccessServices::WebServices::Reports, 159
Count
  DataAccessServices::WebServices::Machines, 111
CreateAction
  DataAccessServices::WebServices::Alerts, 45
CreateAlert
  DataAccessServices::WebServices::Alerts, 46
CreateAlertRule
  DataAccessServices::WebServices::Alerts, 46
CreateCondition
  DataAccessServices::WebServices::Conditions, 55
CreateCondition_ComputerGroup
  DataAccessServices::WebServices::Conditions, 56
CreateCondition_Container
  DataAccessServices::WebServices::Conditions, 56
CreateCondition_Domain
  DataAccessServices::WebServices::Conditions, 56
CreateCondition_NetBIOS
  DataAccessServices::WebServices::Conditions, 57
CreateCredentials
  DataAccessServices::WebServices::Deployment, 64
CreateEvent
  DataAccessServices::WebServices::Events, 83
CreateEventDefinition
  DataAccessServices::WebServices::Events, 83
CreateGroup
  DataAccessServices::WebServices::Groups, 93
CreateInstructions
  DataAccessServices::WebServices::Deployment, 65
CreateMachine
  DataAccessServices::WebServices::DiscoveredMachines, 71
  DataAccessServices::WebServices::Machines, 111
CreateObjectSecurityRole
  DataAccessServices::WebServices::Security, 166
CreatePackage
  DataAccessServices::WebServices::Packages, 140
CreatePackageVersion
  DataAccessServices::WebServices::Packages, 140
CreatePatch
  DataAccessServices::WebServices::Packages, 141
CreatePolicy
  DataAccessServices::WebServices::Security, 166
CreateProduct
  DataAccessServices::WebServices::Products, 153
CreateReportDefinition
  DataAccessServices::WebServices::Reports, 160
CreateReportDefinitions
  DataAccessServices::WebServices::Reports, 160
CreateSecurityElement
  DataAccessServices::WebServices::Security, 166
CreateServer
  DataAccessServices::WebServices::Servers, 172
CreateServerSecurityRole
  DataAccessServices::WebServices::Security, 166
CreateUser
  DataAccessServices::WebServices::Security, 167
DataAccessServices, 39
DataAccessServices.WebServices, 40
DataAccessServices.WebServices.Alerts, 42
DataAccessServices.WebServices.DiscoveredMachines, 69
DataAccessServices.WebServices.Events, 80
DataAccessServices.WebServices.Machines, 106
DataAccessServices.WebServices.Maintenance, 120
DataAccessServices.WebServices.Products, 152
DataAccessServices.WebServices::Alerts
  AddActionConfiguration, 44
  AddEventToAlert, 44
  ApplyActionChanges, 45
  ApplyAlertChanges, 45
  ApplyAlertRuleChanges, 45
  ApplyAlertRuleSecurityChanges, 45
  CreateAlert, 46
  CreateAlertRule, 46
  DeleteAlert, 47
  DeleteAlertRule, 47
DeleteAlertsFromGroupKey, 47
DeleteAlertsFromMachineKey, 47
DeleteAlertsFromSeverity, 48
DeleteAlertsFromStatus, 48
DeleteAlertsFromTime, 48
GetActionFromAlertRuleKey, 48
GetAlertRuleFromKey, 49
GetAlertRules, 49
GetAlerts, 49
GetAlertsFromAlertRuleKey, 49
GetAlertsFromGroupKey, 49
GetAlertsFromMachineKey, 50
GetAlertsFromSeverity, 50
GetAlertsFromStatus, 50
GetAlertsFromTimePeriod, 50
RemoveActionConfiguration, 51
RemoveEventFromAlert, 51
UpdateAction, 51
UpdateActionConfiguration, 51
UpdateAlert, 52
UpdateAlertRule, 52
UpdateAlertRuleSecurity, 53
DataAccessServices.WebServices::Conditions
  ApplyChanges, 55
  CreateCondition, 55
  CreateCondition_ComputerGroup, 56
  CreateCondition_Container, 56
  CreateCondition_Domain, 56
  CreateCondition_NetBIOS, 57
Delete, 57
GetConditions, 57
Update, 57
DataAccessServices.WebServices::DatabaseWebService
  GetInfo, 60
  GetName, 60
  GetNameSpace, 60
  GetServerError, 60
  GetServerTime, 60
  GetTransportPublicKey, 61
  GetVersion, 61
SetNameSpace, 61
DataAccessServices.WebServices::Deployment
  ActivateDeploymentService, 63
  ApplyChanges, 64
  ApplyInstructionsChanges, 64
  ApplyInstructionsStatusChanges, 64
  ClearStatusHistory, 64
  CreateCredentials, 64
  CreateInstructions, 65
  DeleteCredentials, 65
  DeleteInstructions, 65
  GetDeploymentCredentials, 66
  GetDeploymentCredentialsFromGroupKey, 66
  GetDeploymentInstructions, 66
  GetDeploymentInstructionsFromDiscoveredMachineKey, 66
  GetDeploymentInstructionsFromGroupKey, 66
  GetStatusHistory, 67
  SetInstructionsCommandID, 67
  UpdateCredentials, 67
  UpdateInstructions, 67
DataAccessServices.WebServices::DiscoveredMachines
  ApplyMachineChanges, 71
  CheckExpectedGroupPermissions, 71
  CreateMachine, 71
  DeleteAllMachines, 72
  DeleteMachine, 72
  FindMachines, 72
  GetDiscoveryServiceSettings, 72
  GetMachine, 73
  GetMachineFromDnsAndNetbiosServer, 73
  GetMachineFromKey, 73
  GetMachineFromObjectGuidServer, 73
  GetMachines, 74
  GetMachinesDelta, 74
  GetMachinesFromGroupKey, 74
  GetMachinesFromGroupKeyDelta, 74
  GetMachinesFromGroupKeyWithSummary, 75
  GetMachinesFromGroupKeyWithSummaryDelta, 75
  GetMachinesWithSummary, 75
  GetMachinesWithSummaryDelta, 76
  GetMisgroupedDataFromGroupKey, 76
  GetMisgroupedMachineCounts, 76
  GetMisgroupedMachinesFromGroupKey, 76
  GetPreload, 77
  GetPreloadFromGroupKey, 77
  GetUserSpecifiedMachinesFromGroupKey, 77
InvokeDiscovery, 77
IsDiscoveryActive, 77
Move, 78
UpdateDiscoveryInterval, 78
UpdateDiscoveryMode, 78
UpdateMachineDiscovery, 78
DataAccessServices::WebServices::Events
AddEventDateTimeParam, 82
AddEventDefinitionParam, 82
AddEventIntegerParam, 82
AddEventStringParam, 82
ApplyEventChanges, 83
ApplyEventDefinitionChanges, 83
CreateEvent, 83
CreateEventDefinition, 83
DeleteEvent, 84
DeleteEventDefinition, 84
DeleteEventsFromAlertKey, 84
DeleteEventsFromGroupKey, 84
DeleteEventsFromMachineKey, 85
GetEventDefinitions, 85
GetEventDefinitionsFromKey, 85
GetEventFromKey, 85
GetEventParameterValues, 86
GetEventsFromAlert, 86
GetEventsFromGroupKey, 86
GetEventsFromMachineKey, 86
GetEventsFromQuery, 87
GetEventsFromRange, 87
GetExpandedEventsFromQuery, 87
RemoveEventDefinitionParam, 87
UpdateEventDefinition, 88
DataAccessServices::WebServices::Groups
AddGroupLatestPackage, 91
AddGroupPackage, 91
AddGroupPatch, 92
ApplyEventFilterChanges, 92
ApplyGroupChanges, 92
ApplyGroupPackagesChanges, 92
ApplyGroupSecurityChanges, 93
ApplyInstallationScheduleChanges, 93
CreateGroup, 93
DeleteGroup, 94
GetDefault, 94
GetDeploymentGroupsLight, 94
GetDeploymentGroupsLightDto, 94
GetEventFilter, 95
GetGroupFromKey, 95
GetGroupPackages, 95
GetGroups, 95
GetInfo, 96
GetInstallationSchedule, 96
GetStatistics, 96
RemoveGroupPackage, 96
RemoveGroupPatch, 97
UpdateEventFilter, 97
UpdateGroup, 97
UpdateGroupLatestPackage, 98
UpdateGroupPackage, 98
UpdateGroupSecurity, 99
UpdateInstallationSchedule, 99
DataAccessServices::WebServices::Licenses
AddLicense, 103
AddLicenseV2, 103
ApplyChanges, 103
DeleteLicense, 104
DeleteLicenseV1FromReg, 104
DeleteLicenseV2, 104
GetLicenses, 104
GetV2Licenses, 104
UpdateLicenses, 104
DataAccessServices::WebServices::Machines
AddMachineDetails, 108
AddMachinePackage, 108
ApplyAndUpdateMachines, 109
ApplyMachineChanges, 109
ApplyMachineDetailsChanges, 110
ApplyMachineDiagnosticsStateChanges, 110
ApplyMachinePackageChanges, 110
AutoMove, 110
CheckMachineGroupPermissions, 110
Count, 111
CreateMachine, 111
DeleteMachine, 111
FindMachines, 111
GetDeploymentStatistics, 112
GetFromDns, 112
GetFromGroupKey, 112
GetFromGroupKeyDelta, 112
GetFromKey, 113
GetFromObjectGuid, 113
GetMachineDetails, 113
GetMachineDiagnostics, 114
GetMachineFromDnsAndNetbios, 114
GetMachinePackages, 114
GetMachines, 114
GetMachinesDelta, 115
GetPendingDeletion, 115
GetPreloadFromGroupKey, 115
GetPreloadMachines, 115
GetWithObjectGuid, 116
GetWithPackage, 116
GetWithPackageVersion, 116
Move, 116
RemoveMachineDetails, 117
RemoveMachinePackage, 117
UpdateMachine, 117
UpdateMachineDetails, 118
UpdateMachineDiagnosticsState, 118
UpdateMachinePackage, 118
DataAccessServices::WebServices::Maintenance
CancelSchedule, 123
GetSchedules, 123
Maintenance_DisableSchedule, 123
Maintenance_EnableSchedule, 123
Maintenance_LockSchedulingJob, 123
Maintenance_RemoveAllEventsFromDeploymentGroup, 124
Maintenance_RemoveAllEventsOlderThanXDays, 124
Maintenance_RemoveAllEventsWithId, 124
Maintenance_RemoveAllEventsWithinIdRange, 125
Maintenance_RemoveAllEventsOlderThanXDays, 125
Maintenance_RemoveAllOrphanedAlerts, 125
Maintenance_ResetSchedule, 126
Maintenance_UnlockSchedulingJob, 126
Maintenance_UpdateSchedule, 126
Maintenance_UpdateScheduleRemoveEventsFromDeploymentGroup, 127
Maintenance_UpdateScheduleRemoveEventsOlderThan, 127
Maintenance_UpdateScheduleRemoveEventsWithId, 127
Maintenance_UpdateScheduleRemoveEventsWithIdRange, 128
Maintenance_UpdateScheduleRemoveHighVolumeEventsOlderThan, 128
Maintenance_UpdateScheduleRemoveOrphanedAlerts, 129
Maintenance_UpdateScheduleRemoveUnresponsiveMachines, 129
Preview_RemoveAllEventsFromDeploymentGroup, 130
Preview_RemoveAllEventsOlderThanXDays, 130
Preview_RemoveAllEventsWithId, 130
Preview_RemoveAllEventsWithinIdRange, 131
Preview_RemoveAllHighVolumeEventsOlderThanXDays, 131
Preview_RemoveAllOrphanedAlerts, 131
DataAccessServices::WebServices::Packages
ApplyPackageChanges, 135
ApplyPackageSecurityChanges, 135
ApplyPrerequisiteChanges, 136
BeginPackageVersionDownload, 136
BeginPackageVersionUpload, 136
BeginPatchDownload, 137
BeginPatchUpload, 137
BeginPrerequisiteResourceDownload, 137
BeginPrerequisiteResourceUpload, 137
CommitPackageVersion, 138
CommitPatch, 138
ContinuePackageVersionDownload, 138
ContinuePackageVersionUpload, 138
ContinuePatchDownload, 139
ContinuePatchUpload, 139
ContinuePrerequisiteResourceDownload, 139
ContinuePrerequisiteResourceUpload, 140
CreatePackage, 140
CreatePackageVersion, 140
CreatePatch, 141
DeleteAndUnlockPackage, 142
DeletePackage, 142
DeletePrerequisite, 142
FinalisePackageVersion, 142
GetAllDependentPatchesFromPatchKey, 143
GetLegacyPrerequisitesXmlV1, 143
GetLegacyPrerequisitesXmlV2, 143
GetPackageFromKey, 144
GetPackageFromKeyWithInProgress, 144
GetPackageFromPackageVersionKey, 144
GetPackageFromProductKey, 144
GetPackageFromProductName, 145
GetPackageFromType, 145
GetPackages, 145
GetPackagesWithInProgress, 145
GetPackageVersionLength, 146
GetPatchesWithInProgress, 146
GetPatchesWithMissingMetadata, 146
GetPatchFromPackageKey, 146
GetPatchInstallSequenceFromPatchCode, 147
GetPatchInstallSequenceFromPatchKey, 147
GetPatchLength, 147
GetPrerequisiteFromPrerequisiteKey, 147
GetPrerequisiteFromPrerequisiteName, 148
GetPrerequisites, 148
GetPrerequisitesForPackageVersionKey, 148
GetPrerequisitesFromXml, 148
LockPackage, 148
QueryCommitPackageVersionStatus, 149
QueryCommitPatchStatus, 149
RemovePackageVersion, 149
RemovePatch, 149
UnlockPackage, 150
UpdatePackage, 150
UpdatePackageSecurity, 150
WIPSaved, 150
DataAccessServices::WebServices::Products
ApplyChanges, 153
CreateProduct, 153
DeleteProduct, 153
GetProducts, 153
UpdateProduct, 154
DataAccessServices::WebServices::Queries
Execute, 155
DataAccessServices::WebServices::Reports
ApplyReportDefinitionChanges, 158
ApplyReportDefinitionSecurityChanges, 158
BeginReportDefinitionDownload, 158
BeginReportDefinitionUpload, 159
ContinueReportDefinitionDownload, 159
ContinueReportDefinitionUpload, 159
DataAccessServices::WebServices::DiscoveredMachines, 73
GetMachineDetails
DataAccessServices::WebServices::Machines, 113
GetMachineDiagnostics
DataAccessServices::WebServices::Machines, 114
GetMachineFromDnsAndNetbios
DataAccessServices::WebServices::DiscoveredMachines, 73
GetMachineFromDnsAndNetbiosServer
DataAccessServices::WebServices::DiscoveredMachines, 73
GetMachineFromKey
DataAccessServices::WebServices::DiscoveredMachines, 73
GetMachineFromObjectGuidServer
DataAccessServices::WebServices::DiscoveredMachines, 73
GetMachinePackages
DataAccessServices::WebServices::Machines, 114
GetMachines
DataAccessServices::WebServices::Machines, 114
GetMachinesDelta
DataAccessServices::WebServices::DiscoveredMachines, 74
DataAccessServices::WebServices::Machines, 114
GetMachinesFromGroupKey
DataAccessServices::WebServices::DiscoveredMachines, 74
GetMachinesFromGroupKeyDelta
DataAccessServices::WebServices::DiscoveredMachines, 74
GetMachinesFromGroupKeyWithSummary
DataAccessServices::WebServices::DiscoveredMachines, 75
GetMachinesFromGroupKeyWithSummaryDelta
DataAccessServices::WebServices::DiscoveredMachines, 75
GetMachinesWithSummary
DataAccessServices::WebServices::DiscoveredMachines, 75
GetMachinesWithSummaryDelta
DataAccessServices::WebServices::DiscoveredMachines, 76
GetMisgroupedDataFromGroupKey
DataAccessServices::WebServices::DiscoveredMachines, 76
GetMisgroupedMachineCounts
DataAccessServices::WebServices::DiscoveredMachines, 76
GetMisgroupedMachinesFromGroupKey
DataAccessServices::WebServices::DiscoveredMachines, 76
GetName
DataAccessServices::WebServices::DatabaseWebService, 60
GetNameSpace
DataAccessServices::WebServices::DatabaseWebService, 60
GetPackageFromKey
DataAccessServices::WebServices::Packages, 144
GetPackageFromKeyWithInProgress
DataAccessServices::WebServices::Packages, 144
GetPackageFromPackageVersionKey
DataAccessServices::WebServices::Packages, 144
GetPackageFromProductKey
DataAccessServices::WebServices::Packages, 144
GetPackageFromproductName
DataAccessServices::WebServices::Packages, 145
GetPackageType
DataAccessServices::WebServices::Packages, 145
GetPackages
DataAccessServices::WebServices::Packages, 145
GetPackagesWithInProgress
DataAccessServices::WebServices::Packages, 145
GetPackageVersionLength
DataAccessServices::WebServices::Packages, 146
GetPatchesWithInProgress
DataAccessServices::WebServices::Packages, 146
GetPatchesWithMissingMetadata
DataAccessServices::WebServices::Packages, 146
GetPatchFromPatchKey
DataAccessServices::WebServices::Packages, 146
GetPatchInstallSequenceFromPatchCode
DataAccessServices::WebServices::Packages, 147
GetPatchInstallSequenceFromPatchKey
DataAccessServices::WebServices::Packages, 147
GetPatchLength
DataAccessServices::WebServices::Packages, 147
GetPendingDeletion
DataAccessServices::WebServices::Machines, 115
GetPolicies
DataAccessServices::WebServices::Security, 168
GetPreload
DataAccessServices::WebServices::DiscoveredMachines, 77
GetPreloadFromGroupKey
DataAccessServices::WebServices::DiscoveredMachines, 77
DataAccessServices::WebServices::Machines, 115
GetPreloadMachines
DataAccessServices::WebServices::Machines, 115
GetPrerequisiteFromPrerequisiteKey
DataAccessServices::WebServices::Packages, 147
GetPrerequisiteFromPrerequisiteName
DataAccessServices::WebServices::Packages, 148
GetPrerequisites
DataAccessServices::WebServices::Packages, 148
GetPrerequisitesForPackageVersionKey
DataAccessServices::WebServices::Packages, 148
GetPrerequisitesFromXml
  DataAccessServices::WebServices::Packages, 148
GetProducts
  DataAccessServices::WebServices::Products, 153
GetReportDefinition
  DataAccessServices::WebServices::Reports, 161
GetReportDefinitions
  DataAccessServices::WebServices::Reports, 161
GetSchedules
  DataAccessServices::WebServices::Maintenance, 123
GetSecurityElementFromKey
  DataAccessServices::WebServices::Security, 168
GetSecurityElements
 .DataAccessServices::WebServices::Security, 168
GetSecurityElementsFromPolicy
  DataAccessServices::WebServices::Security, 168
GetSecurityRoleFromKey
  DataAccessServices::WebServices::Security, 169
GetSecurityRoles
  DataAccessServices::WebServices::Security, 169
GetSecurityRolesFromType
  DataAccessServices::WebServices::Security, 169
GetServerError
  DataAccessServices::WebServices::DatabaseWebService, 60
GetServerPermissions
 .DataAccessServices::WebServices::Security, 169
GetServers
  DataAccessServices::WebServices::Servers, 172
GetServersFromGroupKey
  DataAccessServices::WebServices::Servers, 173
GetServerTime
 .DataAccessServices::WebServices::DatabaseWebService, 60
GetStatistics
  DataAccessServices::WebServices::Groups, 96
GetStatusHistory
  DataAccessServices::WebServices::Deployment, 67
GetTransportPublicKey
  DataAccessServices::WebServices::DatabaseWebService, 61
GetUserName
  DataAccessServices::WebServices::Security, 169
GetUsers
  DataAccessServices::WebServices::Security, 170
GetUserSpecifiedMachinesFromGroupKey
  DataAccessServices::WebServices::DiscoveredMachines, 77
GetV2Licenses
  DataAccessServices::WebServices::Licenses, 104
GetVersion
 .DataAccessServices::WebServices::DatabaseWebService, 61
GetVisibleReportDefinitions
  DataAccessServices::WebServices::Reports, 161
GetWithObjectGuid
  DataAccessServices::WebServices::Reports, 161
GetWithPackage
 .DataAccessServices::WebServices::Machines, 116
GetWithPackageVersion
  DataAccessServices::WebServices::Machines, 116
InvokeDiscovery
 .DataAccessServices::WebServices::DiscoveredMachines, 77
IsDiscoveryActive
  DataAccessServices::WebServices::DiscoveredMachines, 77
LockPackage
  DataAccessServices::WebServices::Packages, 148
Maintenance_DisableSchedule
  DataAccessServices::WebServices::Maintenance, 123
Maintenance_EnableSchedule
 .DataAccessServices::WebServices::Maintenance, 123
Maintenance_LockSchedulingJob
  DataAccessServices::WebServices::Maintenance, 123
Maintenance_RemoveAllEventsFromDeploymentGroup
  DataAccessServices::WebServices::Maintenance, 124
Maintenance_RemoveAllEventsOlderThanXDays
  DataAccessServices::WebServices::Maintenance, 124
Maintenance_RemoveAllEventsWithId
  DataAccessServices::WebServices::Maintenance, 124
Maintenance_RemoveAllEventsWithinIdRange
  DataAccessServices::WebServices::Maintenance, 125
Maintenance_RemoveAllHighVolumeEventsOlderThanXDays
  DataAccessServices::WebServices::Maintenance, 125
Maintenance_RemoveAllOrphanedAlerts
  DataAccessServices::WebServices::Maintenance, 125
Maintenance_RemoveSchedule
  DataAccessServices::WebServices::Maintenance, 126
Maintenance_ResetSchedule
  DataAccessServices::WebServices::Maintenance, 126
Maintenance_UnlockSchedulingJob
  DataAccessServices::WebServices::Maintenance, 126
Maintenance_UpdateSchedule
 .DataAccessServices::WebServices::Maintenance, 126
Maintenance_UpdateScheduleRemoveEventsFromDeploymentGroup
DataAccessServices::WebServices::Maintenance, 127
Maintenance_UpdateScheduleRemoveEventsOlderThan
DataAccessServices::WebServices::Maintenance, 127
Maintenance_UpdateScheduleRemoveEventsWithId
DataAccessServices::WebServices::Maintenance, 127
Maintenance_UpdateScheduleRemoveEventsWithIdRange
DataAccessServices::WebServices::Maintenance, 128
Maintenance_UpdateScheduleRemoveHighVolumeEventsOlderThan
DataAccessServices::WebServices::Maintenance, 128
Maintenance_UpdateScheduleRemoveOrphanedAlerts
DataAccessServices::WebServices::Maintenance, 129
Maintenance_UpdateScheduleRemoveUnresponsiveMachines
DataAccessServices::WebServices::Maintenance, 129
Move
DataAccessServices::WebServices::DiscoveredMachines, 78
DataAccessServices::WebServices::Machines, 116
Preview_RemoveAllEventsFromDeploymentGroup
DataAccessServices::WebServices::Maintenance, 130
Preview_RemoveAllEventsOlderThanXDays
DataAccessServices::WebServices::Maintenance, 130
Preview_RemoveAllEventsWithId
DataAccessServices::WebServices::Maintenance, 130
Preview_RemoveAllEventsWithinIdRange
DataAccessServices::WebServices::Maintenance, 131
Preview_RemoveAllHighVolumeEventsOlderThanXDays
DataAccessServices::WebServices::Maintenance, 131
Preview_RemoveAllOrphanedAlerts
DataAccessServices::WebServices::Maintenance, 131
QueryCommitPackageVersionStatus
DataAccessServices::WebServices::Packages, 149
QueryCommitPatchStatus
DataAccessServices::WebServices::Packages, 149
RemoveActionConfiguration
DataAccessServices::WebServices::Alerts, 51
RemoveEventDefinitionParam
DataAccessServices::WebServices::Events, 87
RemoveEventFromAlert
DataAccessServices::WebServices::Alerts, 51
RemoveGroupPackage
DataAccessServices::WebServices::Groups, 96
RemoveGroupPatch
DataAccessServices::WebServices::Groups, 97
RemoveMachineDetails
DataAccessServices::WebServices::Machines, 117
RemoveMachinePackage
DataAccessServices::WebServices::Machines, 117
RemovePackageVersion
DataAccessServices::WebServices::Packages, 149
RemovePatch
DataAccessServices::WebServices::Packages, 149
SetInstructionsCommandID
DataAccessServices::WebServices::Deployment, 67
SetNameSpace
DataAccessServices::WebServices::DatabaseWebService, 61
UnlockPackage
DataAccessServices::WebServices::Packages, 150
Update
DataAccessServices::WebServices::Conditions, 57
UpdateAction
DataAccessServices::WebServices::Alerts, 51
UpdateActionConfiguration
DataAccessServices::WebServices::Alerts, 51
UpdateAlert
DataAccessServices::WebServices::Alerts, 52
UpdateAlertRule
DataAccessServices::WebServices::Alerts, 52
UpdateAlertRuleSecurity
DataAccessServices::WebServices::Alerts, 53
UpdateCredentials
DataAccessServices::WebServices::Deployment, 67
UpdateDiscoveryInterval
DataAccessServices::WebServices::DiscoveredMachines, 78
UpdateDiscoveryMode
DataAccessServices::WebServices::DiscoveredMachines, 78
UpdateEventDefinition
DataAccessServices::WebServices::Events, 88
UpdateEventFilter
DataAccessServices::WebServices::Groups, 97
UpdateGroup
DataAccessServices::WebServices::Groups, 97
UpdateGroupLatestPackage
DataAccessServices::WebServices::Groups, 98
UpdateGroupPackage
DataAccessServices::WebServices::Groups, 98
UpdateGroupSecurity
UpdateInstallationSchedule
DataAccessServices::WebServices::Groups, 99
UpdateInstructions
DataAccessServices::WebServices::Deployment, 67
UpdateLicenses
DataAccessServices::WebServices::Licenses, 104
UpdateMachine
DataAccessServices::WebServices::Machines, 117
UpdateMachineDetails
DataAccessServices::WebServices::Machines, 118
UpdateMachineDiagnosticsState
DataAccessServices::WebServices::Machines, 118
UpdateMachineDiscovery
DataAccessServices::WebServices::DiscoveredMachines, 78
UpdateMachinePackage
DataAccessServices::WebServices::Machines, 118
UpdatePackage
DataAccessServices::WebServices::Packages, 150
UpdatePackageSecurity
DataAccessServices::WebServices::Packages, 150
UpdateProduct
DataAccessServices::WebServices::Products, 154
UpdateReportDefinition
DataAccessServices::WebServices::Reports, 161
UpdateReportDefinitionSecurity
DataAccessServices::WebServices::Reports, 162
UpdateSecurityRole
DataAccessServices::WebServices::Security, 170
UpdateServer
DataAccessServices::WebServices::Servers, 173
UpdateUser
DataAccessServices::WebServices::Security, 170
WIPSaved
DataAccessServices::WebServices::Packages, 150