ivanti | Environment Manager

Configuration API

Version 2022.3

Contents

Page 18 of 128

Introduction

This document details the Application Programming Interface (API) provided with the Environment Manager (EM) Console which allows configuration files (i.e. .aemp files) to be generated programmatically, either by a program written in a .NET language or by a PowerShell script. The document is aimed primarily at the PowerShell user.

Format of a Configuration File

EM configuration files have the extension **.aemp** and are normally produced by the "Policy" part of the EM console. The files are deployed to endpoints either manually or via the AppSense Management Centre. The aemp file is a container in zip format, and the contained files may be easily examined by renaming an aemp file to **.zip** and examining the contents with Windows Explorer. The main configuration generated by the console and the API is called "Configuration.xml", and auditing information can be found in "Auditing.xml". When the file is saved by the console or the API, other files are generated by the save process.

Configuration.xml is generated and maintained by an AppSense tool called AOM, and its schema is known as the "UEM" schema. It is not necessary to know the details of AOM to use the configuration SDK.

Prerequisite Knowledge

It is assumed that users of the API are familiar with the EM Console, and the API objects and parameters are named to match the corresponding dialogs and properties on the console. For this reason in many cases no descriptions are provided of objects - they work the same way as they do in the console.

Using the API

The API classes are all contained in the assembly **EMConfigAPI.dll**. This may be found in the EM console directory, normally C:\Program Files\AppSense\Environment Manager\Console. This DLL is dependent on other assemblies in this directory and cannot operate from a different directory unless the dependencies are also moved.

From a .NET Language

It is necessary to add a reference not only to EMConfigAPI.dll, but also to its dependencies. The easiest way to do this is simply to let the compile fail and the compiler to tell you what other dependencies need to be added. If Visual Studio is used, dependent assemblies are copied to the program's bin folder.

EMConfigAPI.dll and all EM Console assemblies are dependent on the .NET Framework Version 4.

From PowerShell

The version of PowerShell shipped with Windows 7 and Windows Server 2008 R2 is linked against version 3.5 of the .NET Framework and will not load EMConfigAPI.dll without altering the configuration of powershell.exe. This will require administrative privilege as the files must be created in the windows system directories.

To do this, create the file **powershell.exe.config** containing the following:

```
<?xml version="1.0"?>
   <configuration>
        <startup useLegacyV2RuntimeActivationPolicy="true">
            <supportedRuntime version="v4.0.30319"/>
            <supportedRuntime version="v2.0.50727"/>
        </startup>
</configuration>
```

This file must be written to C:\Windows\System32\WindowsPowerShell\v1.0. On 64-bit systems, it should also be written to C:\Windows\SysWow64\WindowsPowerShell\v1.0 so that 32-bit PowerShell can load .NET 4 assemblies.   If the integrated script editor powershell_ise.exe is to be used, the same configuration file should be copied to **powershell_ise.exe.config** in both locations.

Loading the API

For .NET languages, all that is required is a reference to EMConfigAPI.dll and its dependencies as explained above. In PowerShell it is necessary to explicitly load the DLL as a .NET assembly as follows:

```
[Reflection.Assembly]::LoadFrom("<path to EMConfigAPI.dll>")
```

For example:

```
[Reflection.Assembly]::LoadFrom("C:\Program Files\AppSense\Environment
Manager\Console\EMConfigAPI.dll")
```

Limitations

In this release there are no edit nor delete facilities. You can only create a configuration from scratch and save it. Also not all functions of the console are currently supported, for example App-V actions and Lockdown. To add either of these actions they can be exported from the console to a template (also known as a snippet) and imported via the ImportPolicyTemplate functions.

Validation of input is not as comprehensive as in the console. For example a MAC address must be added in the correct format (refer to the console). If not the configuration will still be created but won't work on the endpoint.

Classes Overview

EMConfigAPI.EMConfiguration Class

The EMConfiguration is the central class which represents an EM configuration (i.e. an aemp file). All the operations of the API are performed on an instance of EMConfiguration, which is persisted to the aemp file with the SaveConfig function. Here is a simple PowerShell script which saves an empty configuration:

```
[Reflection.Assembly]::LoadFrom("C:\Program Files\AppSense\Environment
Manager\Console\EMConfigAPI.dll")


$config = new-object EMConfigAPI.EMConfiguration

# do nothing here...

$config.SaveConfig("MyConfig.aemp")
```

All modifications to the EMConfiguration class can be cleared by calling the NewConfig() method. In this way multiple files can be generated with the same EMConfiguration object

Nodes

To create a top-level node, use EMConfiguration.AddNodeToTrigger. This function takes a trigger type, an optional node name, optional node description, an optional note and returns a link ID. The link ID is a GUID (globally-unique identifier) which can be used to add child nodes, actions or conditions to the node. The following script adds a node called "Logon" to the user logon trigger and gives it a brief description and note.

```
$userlogonnodeid = $config.AddNodeToTrigger("UserLogon", "Logon", "This is a
description for the parent node", "This is a note")
```

Note above that it is easier in PowerShell to use the string value of an enum, as the syntax for the enum is somewhat verbose. PowerShell will throw an exception if the string does not match one of the enum values.

To add a child node, use AddNodeToParent, passing the link ID of the parent, e.g.

```
$userchildnodeid = $config.AddNodeToParent($userlogonnodeid, "Child", "This is a
description for the child node" , "This is a note")
```

Scheduled Nodes

To create a top-level scheduled node, use EMConfiguration.AddScheduleNodeToTrigger. This function takes a trigger type, a ScheduleNodeObject and an optional node name and returns a link ID. The link ID is a GUID (globally-unique identifier) which can be used to add child nodes, actions or conditions to the node. The

following script adds a schedule node called "Desktop Created" to the user desktop created trigger, do not execute the node immediately and configures it to be a OneTime schedule.

```
$date = Get-Date "07/09/2016 15:30" -format "dd/MM/yyyy hh:mm"

$scheduleType = [EMConfigAPI.Generic.ScheduleType]::OneTimeSchedule("One Time",
$date)

$nodeId = $config.AddScheduleNodeToTrigger("UserDesktopCreated", $false,
$ scheduleType)
```

Note above that it is easier in PowerShell to use the string value of an enum, as the syntax for the enum is somewhat verbose. PowerShell will throw an exception if the string does not match one of the enum values.

To add a child schedule node, use AddScheduleNodeToParent, passing the link ID of the parent, e.g.

```
$userchildnodeid = $config.AddScheduleNodeToParent($nodeId, $false,
$ scheduleType, "Child")
```

Action and Condition Classes

All actions and conditions are represented by classes derived from EMConfigAPI.Generic.ActionBase. All methods of EMConfiguration taking an action or condition take an ActionBase - normally a derivative of ActionBase is passed in.

To add an action or condition to the configuration, use the AddActionOrCondition function. This takes a link ID (telling it where to add the action/condition) and an action or condition (derived from ActionBase). Here is a sample:

```
$ipCondition = new-object EMConfigAPI.Conditions.ComputerIPAddress(
                    "Between","10.0.8.147","10.0.9.122",$true,$true)


$conditionLinkId = $config.AddActionOrCondition($parentId, $ipCondition)
```

The link ID ($parentId in this case) can here be either a node or another action or condition. Actions will usually have conditions as parents. AddActionOrCondition returns the link ID of the created action or condition, which can be used to add child actions to the new action/condition.

Action and Condition Constructors

Most simple action and condition objects have only constructors (no properties or methods). In most cases there are two constructors - the full one, with a full set of parameters, and a simple one taking only the important parameters. The simple constructor provides defaults for the missing parameters (documented below). Note that there is no guarantee that the simple constructor is always valid in all circumstances. In these cases it is necessary to create the action with the full constructor.

Error Reporting

Any errors are reported by raising ApplicationExceptions with appropriate messages. These can be caught in the normal way. The EMConfiguration object validates action and condition placement (i.e. it won't allow actions or conditions to be added to the wrong triggers). The console equivalent would be the action or condition being "greyed out".

Action/Condition Object Re-use

When action and condition objects are added to the configuration the data is copied so that the same original object may be added in multiple places to the configuration if required.

In this case each instance of the action/condition will have a different link ID. Always use the link IDs returned from the add functions (AddActionOrCondition, AddEnvironmentActionOrCondition) and not the linkIDs obtainable via the ActionBase.Action property.

Core Classes

Core classes include EMConfiguration (which represents the configuration itself) and the ActionBase class, which is the base for all actions and conditions.

Enums

The following enums are defined in the EMConfigAPI namespace:

TriggerType

enum **TriggerType** { **UserLogon, UserLogoff, UserProcessStarted, UserProcessStopped, UserNetworkDisconnected, UserNetworkConnected, UserSessionDisconnected, UserSessionReconnected, UserSessionLocked, UserSessionUnlocked, ComputerStartup, ComputerNetworkAvailable, ComputerShutdown, ComputerProcessStarted, ComputerProcessStopped, UserPreSession, UserPreDesktop, UserDesktopCreated,**}

Defines the trigger for the AddNodeToTrigger, AddReusableNodeToTrigger and ImportPolicyTemplateToTrigger methods of the EMConfiguration object.

EnvironmentType

enum **EnvironmentType** { **UserLogonEnvironment, UserLogoffEnvironment, UserNetworkDisconnectedEnvironment, UserNetworkConnectedEnvironment, UserSessionDisconnectedEnvironment, UserSessionReconnectedEnvironment, UserSessionLockedEnvironment, UserSessionUnlockedEnvironment, ComputerStartupEnvironment, ComputerNetworkAvailableEnvironment, ComputerShutdownEnvironment** }

Defines the environment tab for the AddEnvironmentActionOrCondition method of the EMConfiguration. The environment tab is also known as the "pretrigger" tab.

ConfigChangeType

enum **ConfigChangeType** { **Immediately, OnLogon, OnStartup** }

Defines when to apply the changes made to the configuration

LocalAuditLogFormat

enum **LocalAuditLogFormat** { **XML, CSV** }

Defines the local audit log formats

EMConfigAPI.EMConfiguration

Represents the configuration being built. Start point is usually a call to AddNodeToTrigger, but additionaly AddScheduleNodeToTrigger could be called.

Constructor

EMConfiguration()

Constructs a new blank configuration.

Methods

GUID AddActionOrCondition(Guid parentId,Generic.ActionBase actionOrCondition)

Adds an action or condition to the parent identified by parentId. This ID may be a node or another action or condition.   Error checking is performed to make sure the placement of the particular action or condition is valid.

Guid AddEnvironmentActionOrCondition(EnvironmentType environmentType, Generic.ActionBase actionOrCondition)

Adds an action or condition to a trigger's environment area (also known as a pretrigger). This will appear on the console environment tab for the trigger.

Guid[ ] AddIfElseExpression(Guid parentId,string description,string ifGroupDescription)

Adds an "if else" expression to the specified parent entity. Returns an array of 3 IDs as follows:

Index 0 - ID of condition expression to be used to add conditions

Index 1 - ID of "if action". Use to add actions when condition is true

Index 2 - ID of "else action". Use to add actions when condition is false

Results are undefined if no condition is added to the condition expression.

Guid InsertElseIf(Guid ifBlock, string description)

Inserts an "else if" block into an existing IfElse expression:

ifBlock - ID of block to receive the the 'else if'. This may be either the top if block or a previously added ElseIf block. New block follows it.

Returns the id of the added ElseIf block, which is used to add condition(s) and actions

Guid AddNodeToParent(Guid parentId, string name = "", string description = "", string description = "", string notes = "")

Adds a node to a parent node identified by parentId returning the ID of the added node. This ID must represent a node, not an action or condition. If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommended that a meaningful name is used.

The description is optional. If set, the description will appear in the console under the name of the node.

The notes parameter is optional. If set, the description will appear within the Notes tab in the node created.

Guid AddNodeGroupToParent(Guid parentId, string name = "")

Adds a node group to a parent node identified by parentId returning the ID of the added node. This ID must represent a node, not an action or condition. If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommend that a meaningful name is used.

Guid AddNodeToTrigger(TriggerType triggerType, string name = "", string description = "", string notes = "")

Adds a node to the specified trigger, returning a GUID node ID (LinkID). If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommended that a meaningful name is used.

The description is optional. If set, the description will appear in the console under the name of the node.

The notes parameter is optional. If set, the description will appear within the Notes tab in the node created.

Guid AddScheduleNodeToTrigger(TriggerType triggerType, bool runImmediately, ScheduledNodeObject scheduleObject, string name = "", string description = "", string notes = "")

Adds a schedule node to the specified trigger. The runimmediately Boolean is used to allow the scheduled node to be executed as-well as being scheduled. To create a schedule node, you will need to create a schedule node object first. The method will return a GUID node ID (LinkID). If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommend that a meaningful name is used.

The description is optional. If set, the description will appear in the console under the name of the node.

The notes parameter is optional. If set, the description will appear within the Notes tab in the node created.

Guid AddScheduleNodeToParent(Guid parentId, TriggerType triggerType, bool runImmediately, ScheduledNodeObject scheduleObject, string name = "" , string description = "", string notes = "")

Adds a schedule node to a parent node identified by the parentId returning the ID of the added schedule node. The runImmediately boolean is used to allow the scheduled node to be executed as-well as being scheduled. To create a schedule node, you will need to create a schedule node object first. The method will return a GUID node ID (LinkID). If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommend that a meaningful name is used.

The description is optional. If set, the description will appear in the console under the name of the node.

The notes parameter is optional. If set, the description will appear within the Notes tab in the node created.

Guid AddNodeGroupToTrigger(TriggerType triggerType, string name = "")

Adds a node group to the specified trigger, returning a GUID node ID (LinkID). If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommend that a meaningful name is used.

void AddPersonalizationServer(string serverName, string friendlyName = "", bool https = false, int port = 0, bool virtualHost = false)

Adds a personalization server to the configuration. Can be called multiple times to add a list of servers.

Guid AddReusableCondition(Guid parentId, Guid reusableConditionId, string stopOnFail = true)

Adds a condition from the reusable condition library to a parent ID (node or other condition/action). The stop-on-fail flag is set to true by default and mimics console behaviour.

Guid AddReusableNode(Guid parentId, Guid reusableNodeId)

Adds a node from the reusable node library to the specified parent, returning the ID of the added node.

> **ⓘ Note**
>
> This method is used to invoke a reusable node in the main configuration, not to add a sub-node in the reusable node library. Use AddNodeToParent for the latter.

Guid AddReusableNodeToTrigger(TriggerType triggerType,Guid reusableNodeId)

Adds a node from the reusable node library to specified trigger, returning the ID of the added node. If the node contains actions or conditions which are inappropriate for the trigger, an exception is raised.

void AuditingGeneralControl(bool sendToApplicationEventLog,
                                bool sendToAppSenseEventLog,
                                bool makeEventsAnonymous,
                                bool sendToLocalFileLog)

Sets auditing general control flags. If sendToLocalFileLog is true, the log file format and location may be altered with **AudingSetLocalLog**

void AuditingSetLocalLog(LocalAuditLogFormat localAuditLogFormat, string logFileName="")

Sets log file format (enum) and log file name for the local log. If logFileName is not supplied the default is used. Only effective if local file logging is enabled.

void AuditingRaiseEventsLocally (params int[] eventIDs)

List of event IDs to be raised locally. Only specify existing event IDs (refer to console). Not validated.

void CustomSettingAdd(string settingName, bool useDefault, params object[] values)

Adds a custom setting (also known as an engineering key) to the configuration. The setting name must match an available setting as shown on the console. If "useDefault" is false, a value or list of values must be provided of the correct type. (A list is required only for multistring-type settings, e.g. PrinterErrorCodes).

void Dispose()

Implementation of IDisposable::Dispose(). Releases resources used by the object.

Guid[] FindEntitiesByName(string name, bool ignoreCase=true)

Finds entities (nodes,actions or conditions) by name. Utility function to recover the ID of an entity by its name. Since a name is not required to be unique, this function may return multiple results (hence the array return value). If the name is not found, the returned array will be empty. Useful after importing a template if additional entities need to be added to the template subtree.

Guid ImportPolicyTemplateToNode(Guid parentId, string templateFile)

Imports a policy template file (snippet) to the specified parent node and returns the ID of the topmost node. If the constituent entities in the template are named, their IDs can be retrieved with FindEntitiesByName.

Guid ImportPolicyTemplateToTrigger(TriggerType triggerType, string templateFile)

Imports a policy template file (snippet) to the specified trigger and returns the ID of the topmost node. If the constituent entities in the template are named, their IDs can be retrieved with FindEntitiesByName.

Guid InsertReusableConditionNode(string name = "", string description = "", string notes = "")

Insert a node into the reusable condition library, returning its ID.

If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommended that a meaningful name is used.

The description is optional. If set, the description will appear in the console under the name of the node.

Guid InsertReusableNode(string name = "", string description = "", string notes = "")

Inserts a node into the reusable node library at the top level with the indicated name, and returns the ID of the added node. This ID can be used to add actions/conditions/subnodes to the node.

If no name is supplied, the name is set to the GUID. EM does not require each node to have a unique name, but for documentation purposes it is recommended that a meaningful name is used.

The description is optional. If set, the description will appear in the console under the name of the node.

public bool SetRunAsUserCertificate(string thumbprint)

Sets the certificate to encrypt the RunAs passwords. The parameter is the thumbprint of the certificate, which you can get from the Local Machine Certificate Store.

void InsertRunAsUserLibrary(string friendlyName, string userName, string password)

Inserts an entry into the "Run As User" library.

void NewConfig()

Reinitializes the config to the blank state.

void SaveAsXml(string filename)

Saves just the xml part of the configuration to the specified file. The xml part is the "Configuration.xml" file found in the aemp file. Use for testing only. If no file extension is supplied, ".xml" is used.

void SaveConfig(string filename)

Saves the configuration to an aemp file. If no file extension is supplied, ".aemp" is used.

void EnableLogonSubTriggers()

Enables logon sub-triggers. This allows the use of 'pre-session' and 'desktop-created' triggers. This is the default.

void DisableLogonSubTriggers()

Disables logon sub-triggers. This prevents the use of 'pre-session' and 'desktop-created' triggers (i.e. 8.4 legacy mode).

void SetMidSessionConfigChange(ConfigChangeType changeType)

Specify when to apply changes made to the configuration.

void SetSetGlobalGroupPolicyLocation(string location)

Set the global location for group policy actions.

Void EnableExtraNetworkNotifications()

Enables extra network notifications which means a network connect or disconnect notification will be sent to the client for every adapter that connects or disconnects to or from a network regardless of whether there is already a connection to that network. In EM 8.6 this is the default behaviour.

Void DisableExtraNetworkNotifications()

Disables extra network notifications which means there will only be one network connect and one network disconnect notification for each network connected to. This is the same behaviour as EM 8.5 and earlier.

Void EnableUndoActionSequencing()

Cause actions defined by users on 'undo' triggers (Logoff, SessionDisconnect and ConfigChange) to complete before system-added undo actions are executed. (The system adds undo actions for actions which are reverted at logoff, e.g. Map Printer where 'permanent' is not checked.)

Void DisableUndoActionSequencing()

Legacy sequencing, where user actions and system undo actions run simultaneously. This setting does not affect certain system undo actions which are always run after user actions in a specific sequence. This currently applies to Cache Roaming and Virtual Disk undo actions.

public void SetAutomationSettings(string protocol, string dispatcherHostName, int port, string runAsUser)

You must call this method if you are using Automation actions so that the Console can connect to Ivanti Automation to select the action's Module or Runbook.

void SetAzureActiveDirectoryTenantInformation(string tenantId, string applicationId, string certDisplayName, string certLocation, string certStoreName, string certThumbprint)

Sets the information required by Azure AD conditions. The tenant ID can alternatively be the primary domain name e.g. tenant@onmicrosoft.com. Without valid information, Azure AD conditions may not be correctly evaluated on endpoints. This must be called to add any Azure AD conditions.

Properties

UEM.Configuration Configuration

Returns the base configuration object, so that the configuration may be manipulated directly. For advanced use only - no further documentation provided here.

Advanced Configuration Settings

The "Resume interrupted folder copy operations at next user logon" setting from the Advanced Settings dialog can be appied using:

```
$config.ResumeInterruptedFolderCopyItems = $true
```

EMConfigAPI.Generic.ScheduleType

This class contains public enums and static methods that can be used for AddScheduleNodeToTrigger and AddScheduleNodeToParent.

Enums

enum DaysOfWeek {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}

This enum is only applicable for the following schedule types: Weekly and Monthly (on weeks).

enum DaysOfMonth {Day1, Day2, Day3, Day4, Day5, Day6, Day7, Day8, Day9, Day10, Day11, Day12, Day13, Day14, Day15, Day16, Day17, Day18, Day19, Day20, Day21, Day22, Day23, Day24, Day25, Day26, Day27, Day28, Day29, Day30, Day31, Last}

This enum is only applicable for the following schedule types: Monthly (on days).

enum MonthsOfYear {January, February, March, April, May, June, July, August, September, October, November, December}

This enum is only applicable for the following schedule types: Monthly (on days) and Monthly (on weeks).

enum WeeksOfMonth {First, Second, Third, Fourth, Last}

This enum is only applicable for the following schedule types: Monthly (on weeks).

enum DelayTask {Seconds30, Minute1, Minutes30, Hour1, Hours8, Day1}

This enum is applicable to all schedule types, when the option is enabled.

enum RepeatTask {Minutes5, Minutes10, Minutes15, Minutes30, Hour1}

This enum is applicable to all schedule types, when the option is enabled.

enum DurationTask {Minutes15, Minutes30, Hour1, Hours12, Day1, Indefinite}

This enum is applicable to all schedule types, when the option is enabled.

Methods

UEM.ScheduledNodeObject OneTimeSchedule(string scheduleName, DateTime startDate, bool delayTask = false, bool repeatTask = false, DelayTask delayTaskValue = 0, RepeatTask repeatValue = 0, DurationTask repeatDuration = 0)

This is a static method that creates a one time schedule, which returns a UEM. ScheduledNodeObject.

UEM.ScheduledNodeObject DailySchedule(string scheduleName, DateTime startDate, int interval, bool delayTask = false, bool repeatTask = false, DelayTask delayTaskValue = 0, RepeatTask repeatValue = 0, DurationTask repeatDuration = 0)

This is a static method that creates a daily schedule, which returns a UEM. ScheduledNodeObject.

UEM.ScheduledNodeObject WeeklySchedule(string scheduleName, DateTime startDate, int interval, List<DaysOfWeek> daysOfWeek, bool delayTask = false, bool repeatTask = false, DelayTask delayTaskValue = 0, RepeatTask repeatValue = 0, DurationTask repeatDuration = 0)

This is a static method that creates a weekly schedule, which returns a UEM.ScheduledNodeObject.

UEM.ScheduledNodeObject MonthlyOnDaysSchedule(string scheduleName, DateTime startDate, List<MonthsOfYear> monthsOfYear, List<DaysOfMonth> daysOfMonth, bool delayTask = false, bool repeatTask = false, DelayTask delayTaskValue = 0, RepeatTask repeatValue = 0, DurationTask repeatDuration = 0)

This is a static method that creates a monthly (on days) schedule, which returns a UEM. ScheduledNodeObject.

UEM.ScheduledNodeObject MonthlyOnWeeksSchedule(string scheduleName, DateTime startDate, List<MonthsOfYear> monthsOfYear, List<WeeksOfMonth> weeksOfMonth, List<DaysOfWeek> daysOfWeek, bool delayTask = false, bool repeatTask = false, DelayTask delayTaskValue = 0, RepeatTask repeatValue = 0, DurationTask repeatDuration = 0)

This is a static method that creates a monthly (on weeks) schedule, which returns a UEM.ScheduledNodeObject.

EMConfigAPI.Generic.ActionBase

This is the base class of all conditions and actions. Its most common use is to set the name, description and notes on an action or condition. It can also be instantiated directly and used to "manually" create an action or condition if the full description of low-level format of an action or condition is known. This is beyond the scope of this document.

Basic Properties

string Name

Sets the name of the action or condition. The constructor will normally set the name of an action, but this can be modified here.

string Description

Sets the description of the action/condition (empty by default)

string Notes

Add notes to the action/condition (empty by default)

Advanced Properties

UEM.Action Action

Returns the base action object.

bool IsCached

Gets/sets the IsCached ("EvaluateOnce") flag. This flag is only valid for certain conditions, and never for actions.

bool StopOnFail

Gets/Sets the StopOnFail ("StopIfFail") flag. Validity varies according to condition.

Constructor (Advanced Use)

public ActionBase(string identifier, long actionType)

Constructs action with specified identifier and action type flags.

Methods (Advanced Use)

void AddProperty(string Name, string PropertyType, string Value)

Adds a named property of the specified type to the action properties collection

void AddProperty(string Name, string PropertyType, bool Value)

Overload taking a bool value

void AddProperty(string Name, string PropertyType, int Value)

Overload taking an integer value

void AddRunAsProperties(string runAs, string runAsFriendlyName)

Adds RunAs properties to an action which supports the capability.

void AddDelay(string waitInSeconds)

Adds a delay which is run prior to an action. The duration in seconds is specified by the first parameter which must be a non-negative integer value.

Action and Condition Classes

This section provides a list of all action and condition classes in the API.

The following notes apply to Action/Condition classes:

1. Where enums are required, the class defines its own enum. This is to cater for (legitimate) inconsistencies in enum usage between different classes and actions. In PowerShell it is suggested for brevity that the string value is used - PowerShell will coerce it to the correct enum value.

2. The simplified constructor may not always be valid if the constructed object is used on certain triggers. In these cases the full constructor must be used.

3. There are interdependencies between parameters, typically "query" and "match" - if a condition enum is set to "RegExp", the query parameter should be used. In all cases the corresponding dialog on the console should make clear what the usage should be.

4. Action and condition names are generated in a fairly simplistic way by the constructors (unlike in the console, which generates "intelligent" names). The name can be changed after construction via the **Name** property of the inherited ActionBase base object.

EMConfigAPI.Actions.Automation

Performs an automation task

Constructors

EMConfigAPI.Actions.Automation(string taskName, Guid taskId, string taskType, Boolean stopIfFails)

EMConfigAPI.Actions.Automation(string taskName, Guid taskId, string taskType, Boolean waitForCompletion, string customMessage, Boolean stopIfFails)

Methods

void AddParameter(string Name, string Type, string Value1, string Value2)

Adds an Automation parameter to be set by the action. The type value is a string representing an integer (the following are valid: text=0, List=1, Credential=2, Multi-SelectList=3, Multi-LineText=4 and Password=5). Value1 is always required and Value2 is sometimes required by List, Multi-SelectList and Credential types.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.CacheRoaming

Sets up a redirect so a cache can be roamed. Usually used in conjunction with a vhd action.

Constructors

EMConfigAPI.Actions.CacheRoaming(string applicationName, string originalLocation, string redirectLocation, bool reserved, bool stopIfFails)
Creates a redirect with a cache type of 'standard'. Use SetOneDriveCacheType, SetOutlookSearchCacheType or SetWindowsSearchCacheType to change this

Methods

void SetOneDriveCacheType(bool preventUserChangingCredentials, Guid azureTenantId, bool autoRoamOneDriveCredentials)

Changes the action to have a cache type of OneDrive

void SetOutlookSearchCacheType()

Changes the action to have a cache type of OutlookSearch

void SetWindowsSearchCacheType()

Changes the action to have a cache type of WindowsSearch

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.ClientComputerAadGroup

Creates a condition to check whether the computer being connected to belongs to an Azure Active Directory Group.

Constructors

EMConfigAPI.Actions.ClientComputerAadGroup(Condition condition, string groupName, string groupObjectId)
Group can be specified by name or ID. Valid condition operators are "Equal", "NotEqual", "Query" and "RegExp".

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.Comment

Creates a comment that will be within the actions view.

Constructors

EMConfigAPI.Actions.Comment(string description)
Create a comment inline with a description only

EMConfigAPI.Actions.Comment(string description, string notes)
Create a comment inline with a description and notes

EMConfigAPI.Actions.ComputerAadGroup

Creates a condition to check whether the computer belongs to an Azure Active Directory Group.

Constructors

EMConfigAPI.Actions.ComputerAadGroup(Condition condition, string groupName, string groupObjectId)
Group can be specified by name or ID. Valid condition operators are "Equal", "NotEqual", "Query" and "RegExp".

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.CopyFile

Copies one or more files from one location to another location.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.CopyFile(string from, string to, bool overwrite, Condition condition, ExtList excludedExtList, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)
Copies one or more files from one location to another location. Allows the user to specify that destination is to be overwritten if a file exists, a condition to control which files are copied across, a list of files to be excluded based on their extension, the context the action is to be run under and whether execution should stop if the action fails.

EMConfigAPI.Actions.CopyFile(string from, string to)

EMConfigAPI.Actions.CopyFile(string from, string to, bool overwrite)

EMConfigAPI.Actions.CopyFile(string from, string to, bool overwrite, Condition condition)

EMConfigAPI.Actions.CopyFile(string from, string to, bool overwrite, Condition condition, ExtList excludedExtList)

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

Helper class: EMConfigAPI.FileAndFolder.DateCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileDateProperty** { **Created**, **LastModified**, **LastAccessed** }

Constructors

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, DateTime startDate, DateTime endDate)
Constructs a DateCondition object for the given criteria.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when)
Constructs a DateCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

FileDateProperty DateProperty { get; set; }

DateTime StartDate { get; set; }

DateTime EndDate { get; set; }

Helper class: EMConfigAPI.FileAndFolder.SizeCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileSizeUnits** { **B**, **KB**, **MB**, **GB** }

Constructors

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when, ulong startSize, FileSizeUnits startSizeUnits, ulong endSize, FileSizeUnits EndSizeUnits)
Constructs a SizeCondition object for the given criteria.

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when)
Constructs a SizeCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

ulong StartSize { get; set; }

FileSizeUnits StartSizeUnits { get; set; }

ulong EndSize { get; set; }

FileSizeUnits EndSizeUnits { get; set; }

ulong StartSizeInBytes { get; }

ulong EndSizeInBytes { get; }

Helper class: EMConfigAPI.FileAndFolder.ExtList

Constructors

EMConfigAPI.FileAndFolder.ExtList(params string[] values)
Constructs a list of file extensions which are to be excluded.

EMConfigAPI.Actions.CopyFolder

Copies a folder from one location to another location.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

enum **CopyTypeEnum** { **Merge**, **Sync**, **Mirror**, **Backup**, **Copy** }

Constructors

EMConfigAPI.Actions.CopyFolder(CopyTypeEnum copyType, string from, string to, bool copySubFolders, UInt32 folderLevel, bool continueOnError, bool doNotCopyIfDestinationExists, bool overwriteDestinationFiles, bool onlyCopyNewOrChangedFiles, bool excludeJunctionPoints, bool copyAttributes, bool copyFileSecurity, bool copyFileOwner, bool trackDelete, UInt32 threads, UInt32 throttleDelay, Condition fileCondition, Condition folderCondition, ExtList excludedExtList, ExtList excludedPathList, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)
Copies a folder from one location to another location.

EMConfigAPI.Actions.CopyFolder(string from, string to)

EMConfigAPI.Actions.CopyFolder(CopyTypeEnum copyType, string from, string to)

EMConfigAPI.Actions.CopyFolder(string from, string to, bool copySubFolders, UInt32 folderLevel, bool continueOnError, bool doNotCopyIfDestinationExists, bool overwriteDestinationFiles, bool onlyCopyNewOrChangedFiles, bool excludeJunctionPoints, bool copyAttributes, bool copyFileSecurity, bool copyFileOwner, bool trackDelete, UInt32 threads, UInt32 throttleDelay)

EMConfigAPI.Actions.CopyFolder(CopyTypeEnum copyType, string from, string to, bool copySubFolders, UInt32 folderLevel, bool continueOnError, bool doNotCopyIfDestinationExists, bool overwriteDestinationFiles, bool onlyCopyNewOrChangedFiles, bool excludeJunctionPoints, bool copyAttributes, bool copyFileSecurity, bool copyFileOwner, bool trackDelete, UInt32 threads, UInt32 throttleDelay)

EMConfigAPI.Actions.CopyFolder(string from, string to, bool copySubFolders, UInt32 folderLevel, bool continueOnError, bool doNotCopyIfDestinationExists, bool overwriteDestinationFiles, bool onlyCopyNewOrChangedFiles, bool excludeJunctionPoints, bool copyAttributes, bool copyFileSecurity, bool copyFileOwner, bool trackDelete, UInt32 threads, UInt32 throttleDelay, Condition fileCondition, Condition folderCondition)

EMConfigAPI.Actions.CopyFolder(CopyTypeEnum copyType, string from, string to, bool copySubFolders, UInt32 folderLevel, bool continueOnError, bool doNotCopyIfDestinationExists, bool overwriteDestinationFiles, bool onlyCopyNewOrChangedFiles, bool excludeJunctionPoints, bool copyAttributes, bool copyFileSecurity, bool copyFileOwner, bool trackDelete, UInt32 threads, UInt32 throttleDelay, Condition fileCondition, Condition folderCondition)

EMConfigAPI.Actions.CopyFolder(string from, string to, bool copySubFolders, UInt32 folderLevel, bool continueOnError, bool doNotCopyIfDestinationExists, bool overwriteDestinationFiles, bool onlyCopyNewOrChangedFiles, bool excludeJunctionPoints, bool copyAttributes, bool copyFileSecurity, bool copyFileOwner, bool trackDelete, UInt32 threads, UInt32 throttleDelay, Condition fileCondition, Condition folderCondition, ExtList excludedExtList, ExtList excludedPathList)

EMConfigAPI.Actions.CopyFolder(CopyTypeEnum copyType, string from, string to, bool copySubFolders, UInt32 folderLevel, bool continueOnError, bool doNotCopyIfDestinationExists, bool overwriteDestinationFiles, bool onlyCopyNewOrChangedFiles, bool excludeJunctionPoints, bool copyAttributes, bool copyFileSecurity, bool copyFileOwner, bool trackDelete, UInt32 threads, UInt32 throttleDelay, Condition fileCondition, Condition folderCondition, ExtList excludedExtList, ExtList excludedPathList)

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

Helper class: EMConfigAPI.FileAndFolder.DateCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileDateProperty** { **Created**, **LastModified**, **LastAccessed** }

Constructors

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, DateTime startDate, DateTime endDate)
Constructs a DateCondition object for the given criteria.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when)
Constructs a DateCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

FileDateProperty DateProperty { get; set; }

DateTime StartDate { get; set; }

DateTime EndDate { get; set; }

Helper class: EMConfigAPI.FileAndFolder.SizeCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileSizeUnits** { **B**, **KB**, **MB**, **GB** }

Constructors

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when, ulong startSize, FileSizeUnits startSizeUnits, ulong endSize, FileSizeUnits EndSizeUnits)
Constructs a SizeCondition object for the given criteria.

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when)
Constructs a SizeCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

ulong StartSize { get; set; }

FileSizeUnits StartSizeUnits { get; set; }

ulong EndSize { get; set; }

FileSizeUnits EndSizeUnits { get; set; }

ulong StartSizeInBytes { get; }

ulong EndSizeInBytes { get; }

Helper class: EMConfigAPI.FileAndFolder.ExtList

Constructors

EMConfigAPI.FileAndFolder.ExtList(params string[] values)
Constructs a list of file extensions which are to be excluded.

Helper class: EMConfigAPI.FileAndFolder.PathList

Constructors

EMConfigAPI.FileAndFolder.PathList(params string[] values)
Constructs a list of file paths which are to be excluded.

EMConfigAPI.Actions.CreateFolder

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

CreateFolder(string sourceFolder, RunAs runAs, string userFriendlyName, bool stopIfFails)

CreateFolder(string sourceFolder)

Simplified constructor with the following defaults:

runAs	RunAs.Default
userFriendlyName	""
stopIfFails	false

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.CreateShortcut

Enums

enum **RunType** { **Normal**, **Minimized**, **Maximized** }

Constructors

CreateShortcut(string shortcutFilePath, bool applyPermanently, string target, string startInDirectory, RunType run, string parameters, string comment, string iconFilename, int iconIndexInFile, bool stopIfFails)

CreateShortcut(string shortcutFilePath, bool applyPermanently, string target)

Simplified constructor with the following defaults:

startInDirectory	""
run	RunType.Normal
parameters	""
comment	""
iconFilename	""
iconIndexInFile	0
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.CustomAction

Enums

enum **ScriptType** { **Powershell**, **JScript**, **VBScript** }

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

CustomAction(ScriptType type, string scriptText, bool preventInteractive, bool applyEnvironmentVariables, int runFor, RunAs runAs, string userFriendlyName, bool stopIfFails)

CustomAction(ScriptType type, string scriptText)

Simplified constructor with the following defaults:

preventInteractive	true
applyEnvironmentVariables	false
runFor	0
runAs	RunAs.Default
userFriendlyName	""
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.DataNowCustomSettings

Sets a DataNow Custom Setting the user must supply the full information about the setting including registry key, hive, value name, value type and value

Constructors

EMConfigAPI.Actions.DataNowCustomSettings(bool stopOnFails)
Create a DataNow Custom Settings Action. Allowing "Stop On Fails" to be set

Methods

void AddCustomSetting(string settingName, string hive, string key, string valueName, string valueType, string value)

Adds a custom setting to be set by the action

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.DataNowUser

Constructors

DataNowUser(bool singleSignOn, string mapPoint, string dataNowDocuments, string dataNowPictures, string dataNowVideos, string dataNowMusic, bool stopIfFails)

DataNowUser(bool singleSignOn, string mapPoint, string dataNowDocuments, string dataNowPictures, string dataNowVideos, string dataNowMusic)

Simplified constructor with the following defaults:

stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.Delay

Adds an action which waits a given amount of time in seconds.

Constructors

EMConfigAPI.Actions.Delay(string waitInSeconds)
The time in seconds defining the duration of the delay. This must be a non-negative integer value.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.DeleteFile

Deletes one or more files.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.DeleteFile(string source, bool forceDelete, Condition condition, ExtList excludedExtList, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)
Deletes one or more files. Allows the user to specify that a file can be deleted if it is set to read only, a condition to control which files are copied across, a list of files to be excluded based on their extension, the context the action is to be run under and whether execution should stop if the action fails.

EMConfigAPI.Actions.DeleteFile(string source, bool forceDelete)

EMConfigAPI.Actions.DeleteFile(string source, bool forceDelete, Condition condition)

EMConfigAPI.Actions.DeleteFile(string source, bool forceDelete, Condition condition, ExtList excludedExtList)

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

Helper class: EMConfigAPI.FileAndFolder.DateCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between**, **OlderThan** }

enum **FileDateProperty** { **Created**, **LastModified**, **LastAccessed** }

Constructors

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, DateTime startDate, DateTime endDate)
Constructs a DateCondition object for the given criteria.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when)
Constructs a DateCondition object with the CompareToDestination property implicitly set to true.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, System.Int32 olderThan)
Constructs a DateCondition object with the older than condition enabled.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

FileDateProperty DateProperty { get; set; }

DateTime StartDate { get; set; }

DateTime EndDate { get; set; }

System.Int32 OlderThan { get; set; }

Helper class: EMConfigAPI.FileAndFolder.SizeCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileSizeUnits** { **B**, **KB**, **MB**, **GB** }

Constructors

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when, ulong startSize, FileSizeUnits startSizeUnits, ulong endSize, FileSizeUnits EndSizeUnits)
Constructs a SizeCondition object for the given criteria.

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when)
Constructs a SizeCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

ulong StartSize { get; set; }

FileSizeUnits StartSizeUnits { get; set; }

ulong EndSize { get; set; }

FileSizeUnits EndSizeUnits { get; set; }

ulong StartSizeInBytes { get; }

ulong EndSizeInBytes { get; }

Helper class: EMConfigAPI.FileAndFolder.ExtList

Constructors

EMConfigAPI.FileAndFolder.ExtList(params string[] values)
Constructs a list of file extensions which are to be excluded.

EMConfigAPI.Actions.DeleteFolder

Deletes a folder.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.DeleteFolder(string source, bool forceDelete, Condition condition, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)
Deletes a folder. Allows the user to specify that a file can be deleted if it is set to read only, a condition to control which files are copied across, the context the action is to be run under and whether execution should stop if the action fails.

EMConfigAPI.Actions.DeleteFolder(string source, bool forceDelete)

EMConfigAPI.Actions.DeleteFolder(string source, bool forceDelete, Condition condition)

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

Helper class: EMConfigAPI.FileAndFolder.DateCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between**, **OlderThan** }

enum **FileDateProperty** { **Created**, **LastModified**, **LastAccessed** }

Constructors

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, DateTime startDate, DateTime endDate)
Constructs a DateCondition object for the given criteria.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when)
Constructs a DateCondition object with the CompareToDestination property implicitly set to true.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, System.Int32 olderThan)
Constructs a DateCondition object with the older than condition enabled.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

FileDateProperty DateProperty { get; set; }

DateTime StartDate { get; set; }

DateTime EndDate { get; set; }

System.Int32 OlderThan { get; set; }

Helper class: EMConfigAPI.FileAndFolder.SizeCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileSizeUnits** { **B**, **KB**, **MB**, **GB** }

Constructors

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when, ulong startSize, FileSizeUnits startSizeUnits, ulong endSize, FileSizeUnits EndSizeUnits)
Constructs a SizeCondition object for the given criteria.

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when)
Constructs a SizeCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

ulong StartSize { get; set; }

FileSizeUnits StartSizeUnits { get; set; }

ulong EndSize { get; set; }

FileSizeUnits EndSizeUnits { get; set; }

ulong StartSizeInBytes { get; }

ulong EndSizeInBytes { get; }

Helper class: EMConfigAPI.FileAndFolder.ExtList

Constructors

EMConfigAPI.FileAndFolder.ExtList(params string[] values)
Constructs a list of file extensions which are to be excluded.

EMConfigAPI.Actions.DeleteShortcut

Delete a shortcut from the path specified.

Constructors

EMConfigAPI.Actions.DeleteShortcut(string linkPath)
Delete a shortcut using default values for stopIfFails

EMConfigAPI.Actions.DeleteShortcut(string linkPath, bool stopIfFails)
Deletes a shortcut

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.DriveMap

Enums

enum **ConnectAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.DriveMap(string driveLetter, string remotePath, bool unMapAtLogoff, ConnectAs connectAs, string userFriendlyName, bool stopIfFails, string mapFriendlyName, string excludeDriveLetterList, bool hideDriveAfterMapping, bool overrideExistingMapping)

EMConfigAPI.Actions.DriveMap(string driveLetter, string remotePath, bool unMapAtLogoff, ConnectAs connectAs, string userFriendlyName, bool stopIfFails)

EMConfigAPI.Actions.DriveMap(string driveLetter, string remotePath, bool unMapAtLogoff)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.DriveUnmap

Enums

enum **ConnectAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

DriveUnmap(string driveLetter, ConnectAs connectAs, string userFriendlyName, bool stopIfFails)

DriveUnmap(string driveLetter)

Simplified constructor with the following defaults:

connectAs	ConnectAs.Default
userFriendlyName	""
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.EnvironmentDeleteSessionVariable

Constructors

EnvironmentDeleteSessionVariable(string variableName, bool stopIfFails)

EnvironmentDeleteSessionVariable(string variableName)

Simplified constructor with the following defaults:

stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.EnvironmentSetSessionVariable

Constructors

EnvironmentSetSessionVariable(string variableName, string variableValue, bool stopIfFails)

EnvironmentSetSessionVariable(string variableName, string variableValue)

Simplified constructor with the following defaults:

stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.EnvironmentVariableAppend

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

EnvironmentVariableAppend(string variableName, string variableValue, string separator, RunAs runAs, bool stopIfFails)

EnvironmentVariableAppend(string variableName, string variableValue, string separator)

Simplified constructor with the following defaults:

runAs	RunAs.Default
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.EnvironmentVariableDelete

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

EnvironmentVariableDelete(string variableName, RunAs runAs, bool stopIfFails)

EnvironmentVariableDelete(string variableName)

Simplified constructor with the following defaults:

runAs	RunAs.Default
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.EnvironmentVariableSet

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

EnvironmentVariableSet(string variableName, string variableValue, RunAs runAs, bool stopIfFails)

EnvironmentVariableSet(string variableName, string variableValue)

Simplified constructor with the following defaults:

runAs	RunAs.Default
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.Execute

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

Execute(string fileName, string workingDirectory, string parameters, bool doNotExecuteChildrenUntilExited, bool doNotCreateWindow, RunAs runAs, string userFriendlyName, int Timeout, bool stopIfFails)

Execute(string fileName, string workingDirectory, string parameters)

Simplified constructor with the following defaults:

doNotExecuteChildrenUntilExited	false
doNotCreateWindow	false
runAs	RunAs.Default
userFriendlyName	""
Timeout	0
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.FileRename

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

FileRename(string renameFrom, string renameTo, RunAs runAs, string userFriendlyName, bool stopIfFails)

FileRename(string renameFrom, string renameTo)

Simplified constructor with the following defaults:

runAs	RunAs.Default
userFriendlyName	""

stopIfFails	false

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.FileTextFileCreate

Creates a text file.

Enums

enum **TextFileCreateFileEncoding** { **Ansi**, **Utf8**, **Unicode** }

Constructors

FileTextFileCreate(string fileName, bool failIfExists, bool overwriteIfExists, string fileContents, bool readOnly, bool hidden, bool system, bool archive, bool temporary, TextFileCreateFileEncoding fileEncoding, bool stopIfFails)

FileTextFileCreate(string fileName, string fileContents)

Simplified constructor with the following defaults:

failIfExists	false
overwriteIfExists	false
readOnly	false
hidden	false

system	false
archive	true
temporary	false
fileEncoding	TextFileCreateFileEncoding.Ansi
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.FileTextFileUpdate

Update the contents of a text file. When multi-line mode is enabled, use '\n' to represent a newline character and '\\' to represent a backslash

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

FileTextFileUpdate(string file, string searchFor, string replaceWith, bool matchCase, bool useRegularExpressions, bool multiLineMode, RunAs runAs, string userFriendlyName, bool stopIfFails)

FileTextFileUpdate(string file, string searchFor, string replaceWith)

Simplified constructor with the following defaults:

matchCase	false
useRegularExpressions	false

multiLineMode	false
runAs	RunAs.Default
userFriendlyName	""
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.FileTypeAssociations

Associate file types with an application.

Constructors

EMConfigAPI.Actions.FileTypeAssociations(string application, string extensionList)
Associates a comma-separated list of extensions with the application

EMConfigAPI.Actions.FileTypeAssociations(string uwpApplicationName, string uwpApplicationUserModelId, string extensionList)
Associates a comma-separated list of extensions with the uwp application. The uwpApplicationName is only for console display. The agent only uses uwpApplicationUserModelId

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.FolderRedirection

Redirects a known folder from one location to another.

Enums

enum **RedirectionCopyType** { **DoNotCopy**, **UseFolderRedirectionCopy**, **UseEmCopy** }

Constructors

EMConfigAPI.Actions.FolderRedirection(string source, string destination, RedirectionCopyType copyType, bool copyModifiedOnly, bool noRedirectOnFail, bool copyContents, bool deleteOriginalFolder, bool onlyUserHasAccessPermission, bool maintainAccessPermissions, bool availableOffline, bool unapply, bool stopIfFails)

EMConfigAPI.Actions.FolderRedirection(string source, string destination, bool copyContents, bool deleteOriginalFolder, bool onlyUserHasAccessPermission, bool maintainAccessPermissions, bool availableOffline, bool unapply, bool stopIfFails)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.GroupPolicySetADM

Sets an ADM (administrative template for Server2003/XP/2000) group policy on the endpoint. Behavior is exactly the same as the GroupPolicySetADMX class. Refer to that class for more detail.

Enums

enum **PolicyClass** { **Machine**, **User** }

Constructors

EMConfigAPI.Actions.GroupPolicySetADM(PolicyClass policyClass, string policyFolder, bool stopIfFails)
Create a policy of the specified class, using the policyFolder to find ADM files.

EMConfigAPI.Actions.GroupPolicySetADM(PolicyClass policyClass, bool stopIfFails)
Create a policy of the specified class, using the default location for ADM files.

Methods

void EnablePolicy(string[] categoryPath, string policyName, params object[] elementValues)

See **GroupPolicySetADMX.EnablePolicy**

void DisablePolicy(string[] categoryPath, string policyName)

See **GroupPolicySetADMX.DisablePolicy**

Properties

bool ApplyPolicySettingsPermanently { get; set; }

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

Helper class: EMConfigAPI.Actions.ADMList

Constructors

EMConfigAPI.Actions.ADMList(params string[] values)
Constructs a single column list from a string array for ADM/ADMX actions with list values

Helper class: EMConfigAPI.Actions.ADMKeyValueList

Constructors

EMConfigAPI.Actions.ADMKeyValueList(params string[] values)
Constructs a two-column list (keys,values) from a string array for ADM/ADMX actions with list values. The source array consists of alternate key-value pairs and must have an even number of elements.

EMConfigAPI.Actions.ADMKeyValueList(List<string> keys, List<string> values)
Constructs a two-column list (keys,values)from two separate lists.

EMConfigAPI.Actions.GroupPolicySetADMX

Sets an ADMX (administrative template for Server2008/Vista and later) group policy on the endpoint. After construction, policies are added to the action via the EnablePolicy and DisablePolicy methods.

Enums

enum **PolicyClass** { **Machine**, **User** }

Constructors

EMConfigAPI.Actions.GroupPolicySetADMX(PolicyClass policyClass, string policyFolder, bool stopIfFails)
Create a policy of the specified class, using the policyFolder to find ADMX files.

EMConfigAPI.Actions.GroupPolicySetADMX(PolicyClass policyClass, bool stopIfFails)
Create a policy of the specified class, using the default location for ADMX files.

Methods

void EnablePolicy(string[] categoryPath, string policyName, params object[] elementValues)

Enables a policy setting located by the categoryPath. The category path is an array of category names which correspond to the tree structure in the console and group policy editor. So for example the category "Windows\Backup\Client" would be coded (in PowerShell) as @("Windows","Backup","Client").

policyName is the verbose name of the policy as shown in the right hand pane of the console and group policy editor, e.g. "Hide specified control panel items".

Where a policy requires further values they are supplied as a list in elementValues.

Each value in the list must match the expected type for the policy. Two helper classes are provided for list-type elements (see below).

void DisablePolicy(string categoryPath, string policyName)

Disables a policy setting located by the categoryPath and the policyName (long name of the policy). If the policy has previously been added to the object it will be disabled.

Properties

bool ApplyPolicySettingsPermanently { get; set; }

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

Helper class: EMConfigAPI.Actions.ADMList

Constructors

EMConfigAPI.Actions.ADMList(params string[] values)
Constructs a single column list from a string array for ADM/ADMX actions with list values

Helper class: EMConfigAPI.Actions.ADMKeyValueList

Constructors

EMConfigAPI.Actions.ADMKeyValueList(params string[] values)
Constructs a two-column list (keys,values) from a string array for ADM/ADMX actions with list values. The source array consists of alternate key-value pairs and must have an even number of elements.

EMConfigAPI.Actions.ADMKeyValueList(List<string> keys, List<string> values)
Constructs a two-column list (keys,values)from two separate lists.

EMConfigAPI.Actions.LogonLogoffMessage

Constructors

LogonLogoffMessage(string Text, bool stopIfFails)

LogonLogoffMessage(string Text)

Simplified constructor with the following defaults:

stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.ModifyAttributes

Modifies the attributes of a file.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

enum **AttributeSettings** { **Ignore**, **Set**, **Unset** }

Constructors

EMConfigAPI.Actions.ModifyAttributes(string file, AttributeSettings readOnly, AttributeSettings hidden, AttributeSettings system, AttributeSettings archive, AttributeSettings temporary, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)

EMConfigAPI.Actions.ModifyAttributes(string file, AttributeSettings readOnly, AttributeSettings hidden, AttributeSettings system, AttributeSettings archive, AttributeSettings temporary)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.MoveFile

Moves one or more files from one location to another location.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.MoveFile(string from, string to, bool overwrite, Condition condition, ExtList excludedExtList, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)
Moves one or more files from one location to another location. Allows the user to specify that destination is to be overwritten if a file exists, a condition to control which files are copied across, a list of files to be excluded based on their extension, the context the action is to be run under and whether execution should stop if the action fails.

EMConfigAPI.Actions.MoveFile(string from, string to)

EMConfigAPI.Actions.MoveFile(string from, string to, bool overwrite)

EMConfigAPI.Actions.MoveFile(string from, string to, bool overwrite, Condition condition)

EMConfigAPI.Actions.MoveFile(string from, string to, bool overwrite, Condition condition, ExtList excludedExtList)

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

Helper class: EMConfigAPI.FileAndFolder.DateCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileDateProperty** { **Created**, **LastModified**, **LastAccessed** }

Constructors

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, DateTime startDate, DateTime endDate)
Constructs a DateCondition object for the given criteria.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when)
Constructs a DateCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

FileDateProperty DateProperty { get; set; }

DateTime StartDate { get; set; }

DateTime EndDate { get; set; }

Helper class: EMConfigAPI.FileAndFolder.SizeCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileSizeUnits** { **B**, **KB**, **MB**, **GB** }

Constructors

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when, ulong startSize, FileSizeUnits startSizeUnits, ulong endSize, FileSizeUnits EndSizeUnits)
Constructs a SizeCondition object for the given criteria.

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when)
Constructs a SizeCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

ulong StartSize { get; set; }

FileSizeUnits StartSizeUnits { get; set; }

ulong EndSize { get; set; }

FileSizeUnits EndSizeUnits { get; set; }

ulong StartSizeInBytes { get; }

ulong EndSizeInBytes { get; }

Helper class: EMConfigAPI.FileAndFolder.ExtList

Constructors

EMConfigAPI.FileAndFolder.ExtList(params string[] values)
Constructs a list of file extensions which are to be excluded.

EMConfigAPI.Actions.ODBCAmend

Amends an ODBC connection.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.ODBCAmend(string connectionName, string driverType, Hashtable ODBCDataPairs, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)

EMConfigAPI.Actions.ODBCAmend(string connectionName, string driverType, Hashtable ODBCDataPairs)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.ODBCCreate

Creates an ODBC connection.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.ODBCCreate(string connectionName, string driverType, Hashtable ODBCDataPairs, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)

EMConfigAPI.Actions.ODBCCreate(string connectionName, string driverType, Hashtable ODBCDataPairs)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.ODBCDelete

Deletes an ODBC connection.

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.ODBCDelete(string connectionName, string driverType, RunAs runAs, RunAs userFriendlyName, bool stopIfFails)

EMConfigAPI.Actions.ODBCDelete(string connectionName, string driverType)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.OutlookCreateProfile

Creates a new Outlook Profile.

Constructors

EMConfigAPI.Actions.OutlookCreateProfile(string profileName, string exchangeServer)
Constructs an OutlookCreateProfile object with a default value for mailboxName/stopOnFail.

EMConfigAPI.Actions.OutlookCreateProfile(string profileName, string exchangeServer, bool stopOnFail)
Constructs an OutlookCreateProfile object with a default value for mailboxName.

EMConfigAPI.Actions.OutlookCreateProfile(string profileName, string mailboxName, string exchangeServer)
Constructs an OutlookCreateProfile object with a default value for stopOnFail.

EMConfigAPI.Actions.OutlookCreateProfile(string profileName, string mailboxName, string exchangeServer, bool stopOnFail)
Constructs an OutlookCreateProfile object.

Methods

void AddMailbox(string displayName, string mailboxDistinguishedName)

Adds an item to the "Open these additional mailboxes" list.

void EnableCachedMode(bool enabled, bool downloadSharedFolders, bool downloadPublicFavourites, string ostFileName, string addressBookPath, System.Int32 offlinePeriod)

Enables cached exchange mode.

void EnableOutlookAnywhere(string url, bool useHttpOnFastNetworks, bool useHttpOnSlowNetworks, bool ntlmAuthenticate)

Enables Outlook Anywhere with default values for useSSL/mutuallyAuthenticate/principalUrl.

void EnableOutlookAnywhere(string url, bool useSSL, bool mutuallyAuthenticate, string principalUrl, bool useHttpOnFastNetworks, bool useHttpOnSlowNetworks, bool ntlmAuthenticate)

Enables Outlook Anywhere.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.OutlookCreateProfile365

Creates a new Outlook 365 Profile.

Constructors

EMConfigAPI.Actions.OutlookCreateProfile365(string profileName, string userEmail)
Constructs an OutlookCreateProfile365 object with a default value for stopOnFail.

EMConfigAPI.Actions.OutlookCreateProfile365(string profileName, string userEmail, bool stopOnFail)
Constructs an OutlookCreateProfile365 object.

Methods

void AddMailbox(string displayName, string mailboxDistinguishedName)

Adds an item to the "Open these additional mailboxes" list.

void EnableCachedMode(bool enabled, bool downloadSharedFolders, bool downloadPublicFavourites, string ostFileName, string addressBookPath, System.Int32 offlinePeriod)

Enables cached exchange mode.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.OutlookEmailSignature

Creates a new Outlook signature.

Constructors

EMConfigAPI.Actions.OutlookEmailSignature(string name)
Constructs an OutlookEmailSignature object with a default value for stopOnFail.

EMConfigAPI.Actions.OutlookEmailSignature(string name, bool stopOnFail)
Constructs an OutlookEmailSignature object.

Methods

void LoadNewMessagesFromRtfFile(string fileName)

Loads **OutlookEmailSignature.NewMessages** from a file.

void LoadRepliesFromRtfFile(string fileName)

Loads **OutlookEmailSignature.Replies** from a file.

Properties

string NewMessages { get; set; }

bool UseNewMessagesForReplies { get; set; }

string Replies { get; set; }

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.OutlookUpdateMailboxes

Creates/Removes Outlook mailboxes.

Constructors

EMConfigAPI.Actions.OutlookUpdateMailboxes(string profileName)
Constructs an OutlookUpdateMailboxes object with a default value for stopOnFail.

EMConfigAPI.Actions.OutlookUpdateMailboxes(string profileName, bool stopOnFail)
Constructs an OutlookUpdateMailboxes object.

Methods

void AddMailbox(string displayName, string mailboxDistinguishedName)

Adds an item to the "Open these additional mailboxes" list.

void RemoveMailbox(string displayName)

Adds an item to the "Remove these additional mailboxes" list.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.Pin

Sets an application to be pinned/ unpinned to the windows Start Menu and/or TaskBar.

Constructors

EMConfigAPI.Actions.Pin(string fileName, bool pin, bool startMenu, bool taskBar, bool stopIfFails)
Creates an pin/unpin action with a default value for stopOnFail.

EMConfigAPI.Actions.Pin(string fileName, bool pin, bool startMenu, bool taskBar)
Creates an pin/unpin action.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.PrinterMap

Enums

enum **ConnectAs** { **CurrentUser**, **AsUser**, **Default** }

Constructors

PrinterMap(string printerPath, bool setAsDefault, bool unMapAtLogoff, string friendyName, int retries, int timeout, string driverLocation, string driverName, ConnectAs connectAs, string userFriendlyName, bool stopIfFails)

PrinterMap(string printerPath)

Simplified constructor with the following defaults:

setAsDefault	false
unMapAtLogoff	false
friendyName	""
retries	0
timeout	1
driverLocation	""
driverName	""
connectAs	ConnectAs.Default
userFriendlyName	""
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.PrinterSetDefault

Enums

enum **ConnectAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

PrinterSetDefault(string remotePrinterPath, bool unMapAtLogoff, ConnectAs connectAs, string userFriendlyName, bool stopIfFails)

PrinterSetDefault(string remotePrinterPath, bool unMapAtLogoff)

Simplified constructor with the following defaults:

connectAs	ConnectAs.Default
userFriendlyName	""
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.PrinterUnmap

Unmaps a specific printer, or all printers

Enums

enum **RunAs** { **CurrentUser**, **System**, **AsUser**, **Default** }

Constructors

EMConfigAPI.Actions.PrinterUnmap(string path)
Unmap a printer using default values for for runAs, userFriendlyname and userFriendlyname

EMConfigAPI.Actions.PrinterUnmap(string path, RunAs runAs, string userFriendlyname, bool stopOnFail)
Unmap a printer

EMConfigAPI.Actions.PrinterUnmap()
Unmap all printers using default values for for runAs, userFriendlyname and userFriendlyname

EMConfigAPI.Actions.PrinterUnmap(RunAs runAs, string userFriendlyname, bool stopOnFail)
Unmap all printers

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.RegistryCreateKey

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

RegistryCreateKey(string mainKey, string subKey, RunAs runAs, bool personalizationOverride, bool stopIfFails)

RegistryCreateKey(string mainKey, string subKey)

Simplified constructor with the following defaults:

runAs	RunAs.Default
personalizationOverride	false

stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.RegistryDeleteKey

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

RegistryDeleteKey(string mainKey, string subKey, RunAs runAs, bool stopIfFails)

RegistryDeleteKey(string mainKey, string subKey)

Simplified constructor with the following defaults:

runAs	RunAs.Default
stopIfFails	false

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.RegistryDeleteValue

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

RegistryDeleteValue(string mainKey, string subKey, string valueName, RunAs runAs, bool stopIfFails)

RegistryDeleteValue(string mainKey, string subKey, string valueName)

Simplified constructor with the following defaults:

runAs	RunAs.Default
stopIfFails	false

Properties

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.RegistryHiving

Hives one or more items from or to the registry.

Enums

enum **ImportExport** { **Import**, **Export** }

enum **RestoreFormat** { **Replace**, **Merge** }

Constructors

RegistryHiving(string title, string location, ImportExport importOrExport, bool stopIfFails)

RegistryHiving(string title, string location, ImportExport importOrExport)

Simplified constructor with the following defaults:

stopIfFails	false

Methods

void AddHiveItem(String RegistryKey, String RegistryValue, Boolean UseDefault, RestoreFormat RestoreFormat, Boolean UseRegistryKeyAsFileName, String FileName)

Use this function to add the items that are to be hived from or to a file.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.RegistrySetDefaultValue

Sets the (Default) value in a registry key. The constructor sets REG_SZ values only. Use the ValueXXX properties to set values of other types.

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

RegistrySetDefaultValue(string mainKey, string subKey, string value, RunAs runAs, bool stopIfFails)

RegistrySetDefaultValue(string mainKey, string subKey, string value)

Simplified constructor with the following defaults:

runAs	RunAs.Default
stopIfFails	false

Properties

UInt32 ValueDWORD { set; }
Set a DWORD value (overriding previous value)

UInt64 ValueQWORD { set; }
Set a QWORD value (overriding previous value)

string ValueEXPAND_SZ { set; }
Set a REG_EXPAND_SZ value (overriding previous value)

string[] ValueMULTI_SZ { set; }
Supply an array of strings to set a REG_MULTI_SZ value (overriding previous value)

byte[] ValueBINARY { set; }
Supply an array of bytes to set a REG_BINARY value (overriding previous value)

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.RegistrySetValue

Sets a registry value. The constructor sets REG_SZ values only. Use the ValueXXX properties to set values of other types.

Enums

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

RegistrySetValue(string mainKey, string subKey, string valueName, string value, RunAs runAs, bool stopIfFails)

RegistrySetValue(string mainKey, string subKey, string valueName, string value)

Simplified constructor with the following defaults:

runAs	RunAs.Default
stopIfFails	false

Properties

UInt32 ValueDWORD { set; }
Set a DWORD value (overriding previous value)

UInt64 ValueQWORD { set; }
Set a QWORD value (overriding previous value)

string ValueEXPAND_SZ { set; }
Set a REG_EXPAND_SZ value (overriding previous value)

string[] ValueMULTI_SZ { set; }
Supply an array of strings to set a REG_MULTI_SZ value (overriding previous value)

byte[] ValueBINARY { set; }
Supply an array of bytes to set a REG_BINARY value (overriding previous value)

bool PersonalizationOverride { set; }
Set this property to allow policy configuration to take precedence over user personalization

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.ScreenText

Constructors

ScreenText(string text, bool stopIfFails)

ScreenText(string text)

Simplified constructor with the following defaults:

stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.SelfHealFile

Constructors

SelfHealFile(string filePathName, bool ensureFileExists, bool ensureFileIsNotChanged, bool stopIfFails)

SelfHealFile(string filePathName, bool ensureFileExists)

Simplified constructor with the following defaults:

ensureFileIsNotChanged	false
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.SelfHealProcess

Constructors

SelfHealProcess(string processName, string processDirectory, string parameters, bool runAsSystem, bool stopIfFails)

SelfHealProcess(string processName, string processDirectory)

Simplified constructor with the following defaults:

parameters	""
runAsSystem	false
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.SelfHealRegistry

Constructors

SelfHealRegistry(string mainKey, string subKey, string valueName, bool useDefaultValue, bool stopIfFails)

SelfHealRegistry(string mainKey, string subKey, string valueName)

Simplified constructor with the following defaults:

useDefaultValue	false
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.SelfHealService

Enums

enum **ServiceStatus** { **EnsureAlwaysRunning**, **EnsureNeverRuns** }

Constructors

SelfHealService(string displayName, string serviceName, ServiceStatus serviceStatus, string startupParameters, bool stopIfFails)

SelfHealService(string displayName, string serviceName)

Simplified constructor with the following defaults:

serviceStatus	ServiceStatus.EnsureAlwaysRunning
startupParameters	""

stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.SetDesktopWallpaper

Sets the image to be shown as the desktop wallpaper.

Enums

enum **PositionEnum** { **Fill**, **Fit**, **Span**, **Stretch**, **Tile**, **Center** }

Constructors

SetDesktopWallpaper(string image, PositionEnum position, bool copyLocal, string destinationFolder, bool stopIfFails)

SetDesktopWallpaper(string image)

Simplified constructor with the following defaults:

position	PositionEnum.Center
copyLocal	true
destinationFolder	"%USERPROFILE%\\Pictures"
stopIfFails	false

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.UserAadGroupMembership

Creates a condition to check whether the user belongs to an Azure Active Directory Group.

Constructors

EMConfigAPI.Actions.UserAadGroupMembership(Condition condition, string groupName, string groupObjectId)
Group can be specified by name or ID. Valid condition operators are "Equal", "NotEqual", "Query" and "RegExp".

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.UserAadName

Creates a condition to check whether the user is an Azure Active Directory user.

Constructors

EMConfigAPI.Actions.UserAadName(Condition condition, string userName, string objectId)
User can be specified by name or ID. Valid condition operators are "Equal", "NotEqual", "Query" and "RegExp".

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.VHD

Attaches to and optionally creates a VHD(X).

Enums

enum **VhdFormat** { **Expandable**, **Fixed** }

enum **VhdUnits** { **MB**, **GB** }

enum **VhdStorageTypes** { **VHD**, **VHDX** }

Constructors

EMConfigAPI.Actions.VHD(string vhdLocation, string workingDirectory, bool autoCreate, VhdFormats vhdType, string size, VhdUnits bit, bool undoAction, bool stopIfFails, VhdStorageTypes storageType)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Actions.Windows10StartMenu

Creates Windows 10 Start Menu action on user logon triggers. The AddUwpApplication and AddWin32Application methods add applications to the application list, and the TileLayoutXml property receives the xml defining the tile area. As with the console, a start menu action must have at least one application and one tile. This is validated when the action is added to the configuration.

Constructors

EMConfigAPI.Actions.Windows10StartMenu(string actionName, bool preventuserChange, bool stopIfFails)
The actionName is required, other parameters default to false

Methods

void AddUwpApplication(string appUserModelId, string displayName)

Adds a Uwp application to the application list. The appUserModelId is required. displayName is required by the console to display the application if it is not in the available applications list. It has no effect on the endpoint.

void AddWin32Application(string exePath, string shortcutPath, string arguments, string workingDirectory, string description, Windows10Menu.ShowCommand showCommand, int iconIndex, string iconLocation, string parentDisplayName)

Adds a Win32 application to the application list. Parameters other than exePath or shortcutPath may be omitted as defaults are applied. Only trailing parameters can be defaulted in a powershell script.
Parameters as follows:

exePath: path to executable on endpoint. Environment variables should be used for such folders as Program Files and Windows etc.

shortcutPath: path to the shortcut to be created. When harvesting applications on the machine, the strings %ALLUSERSPROFILE% and %PROGRAMDATA% are replaced with %APPDATA%. This ensures shortcuts that pertain to all users on a machine are not modified.

arguments: Argument list for running the program.

workingDirectory: Start directory for the program. Default is "".

description: Description for the shortcut. Default is "".

showCommand: Initial window state. Enum values are Normal, Minimized, Maximized. Default is Normal.

iconIndex: index of icon in file containing icon (icon location). Default 0.

iconLocation: path to file containing icon. If the default of "" is applied, the endpoint assumes exePath

parentDisplayName: used in console only to indicate the folder the shortcut can be found in. This is only used to retrieve the correct icon on the console wizard. If defaulted to null, the action will use the first component of the shortcut path below the base "Programs" folder. For folders with a 'localized name' (e.g. 'System Tools' is localized to 'Windows System' on English systems), the correct localized name is required or the console will not show an icon.

Properties

string TileLayoutXml { set; }
Set the tile layout xml as a string

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the action

string Description { set; }
Sets the action description

string Notes { set; }
Sets the action notes

EMConfigAPI.Conditions.CitrixClientSettings

Compares the Citrix settings for a connecting client.

Enums

enum **CitrixClientConditionOperator** { **Any**, **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual** }

Constructors

Methods

void SetClientOS(string OperatingSystem)

Specify the operating system of the connecting client. The valid options are Any, Android, BlackBerry, BlackBerry Playbook, DOS32, EPOC, HTML5, iOS, Java, Macintosh, OS/2, UNIX/Linux, Windows CE, Windows Mobile, Windows Phone 8/WinRT.

void SetClientType(string ClientType)

You can specify what the device type is for the connecting client. The valid options are Any, Phone, Tablet, Other, Unknown.

void SetNetScalerPolicies(string NetScalerPolicy)

Specify the Netscaler policy inforce for the connecting client

void SetNetScalerHostName(string NetScalerHostName)

Specify the NetScaler hostname of the connecting client

void SetClientVersion(operator CitrixClientConditionOperator, string ClientVersion)

Specify the version of the connecting client.

void SetClientEncryption(operator CitrixClientConditionOperator, string ClientEncryptionType)

Specify the encryption level of the connecting client. The valid options are Basic, LogonOnly, 40bits, 56bits, 128bits.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientComputerDomainMembership

Enums

enum **LookupType** { **DNS**, **Windows** }

enum **Condition** { **Equal**, **NotEqual**, **Query**, **RegExp** }

Constructors

ClientComputerDomainMembership(LookupType lookupType, Condition condition, string match, string query, bool matchCase, bool evaluateOnce, bool stopIfFails)

ClientComputerDomainMembership(LookupType lookupType, Condition condition, string match)

Simplified constructor with the following defaults:

query	""
matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientComputerGroup

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query** }

Constructors

ClientComputerGroup(Condition condition, string match, string query, bool searchNestedGroups, bool evaluateOnce, bool stopIfFails)

ClientComputerGroup(Condition condition, string match)

Simplified constructor with the following defaults:

query	""
searchNestedGroups	true
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientConnectionProtocol

Enums

enum **Condition** { **Equal**, **NotEqual** }

enum **Protocol** { **RDP**, **ICA**, **Console**, **PCoIP** }

Constructors

ClientConnectionProtocol(Condition condition, Protocol match, bool evaluateOnce, bool stopIfFails)

ClientConnectionProtocol(Condition condition, Protocol match)

Simplified constructor with the following defaults:

evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientIPAddress

Enums

enum **Condition** { **Equal**, **NotEqual**, **Between** }

Constructors

ClientIPAddress(Condition condition, string address, string address2, bool evaluateOnce, bool stopIfFails)

ClientIPAddress(Condition condition, string address)

Simplified constructor with the following defaults:

address2	""
evaluateOnce	false
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientNETBiosName

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query**, **RegExp** }

Constructors

ClientNETBiosName(Condition condition, string match, string query, bool matchCase, bool evaluateOnce, bool stopIfFails)

ClientNETBiosName(Condition condition, string match)

Simplified constructor with the following defaults:

query	""
matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientPublishedApplicationName

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query**, **RegExp** }

Constructors

ClientPublishedApplicationName(Condition condition, string match, bool matchCase, bool evaluateOnce, bool stopIfFails)

ClientPublishedApplicationName(Condition condition, string match)

Simplified constructor with the following defaults:

matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientScreenColorDepth

Enums

enum **Condition** { **Equal**, **NotEqual**, **Between**, **From** }

enum **Colors** { **Colors16**, **Colors256**, **Colors32768**, **Colors65536**, **Colors16M** }

enum **Colors2** { **Colors16**, **Colors256**, **Colors32768**, **Colors65536**, **Colors16M** }

Constructors

ClientScreenColorDepth(Condition condition, Colors colors, Colors2 colors2, bool evaluateOnce, bool stopIfFails)

ClientScreenColorDepth(Condition condition, Colors colors)

Simplified constructor with the following defaults:

colors2	Colors2.Colors16
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ClientScreenResolution

Enums

enum **Condition** { **Equal**, **NotEqual**, **Between**, **GreaterThanOrEqual** }

Constructors

ClientScreenResolution(Condition resolutionIs, int X, int Y, int X2, int Y2, bool evaluateOnce, bool stopIfFails)

ClientScreenResolution(Condition resolutionIs, int X, int Y)

Simplified constructor with the following defaults:

X2	0
Y2	0
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerDomain

Enums

enum **LookupType** { **DNS**, **Windows** }

enum **Condition** { **Equal**, **NotEqual**, **Query**, **RegExp** }

Constructors

ComputerDomain(LookupType lookupType, Condition condition, string match, string query, bool matchCase, bool evaluateOnce, bool stopIfFails)

ComputerDomain(LookupType lookupType, Condition condition, string match)

Simplified constructor with the following defaults:

query	""
matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerGroup

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query** }

Constructors

ComputerGroup(Condition condition, string match, string query, bool searchNestedGroups, bool evaluateOnce, bool stopIfFails)

ComputerGroup(Condition condition, string match)

Simplified constructor with the following defaults:

query	""
searchNestedGroups	true
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerIPAddress

Enums

enum **Condition** { **Equal**, **NotEqual**, **Between** }

Constructors

ComputerIPAddress(Condition condition, string address, string address2, bool evaluateOnce, bool stopIfFails)

ComputerIPAddress(Condition condition, string address)

Simplified constructor with the following defaults:

address2	""
evaluateOnce	false
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerIsVDI

Constructors

ComputerIsVDI(bool isVdi, bool evaluateOnce, bool stopIfFails)

ComputerIsVDI()

Simplified constructor with the following defaults:

isVdi	true
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerMACAddress

Enums

enum **Condition** { **Equal**, **NotEqual** }

Constructors

ComputerMACAddress(Condition condition, string match, bool evaluateOnce, bool stopIfFails)

ComputerMACAddress(Condition condition, string match)

Simplified constructor with the following defaults:

evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerName

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query**, **RegExp** }

Constructors

ComputerName(Condition condition, string match, string query, bool matchCase, bool evaluateOnce, bool stopIfFails)

ComputerName(Condition condition, string match)

Simplified constructor with the following defaults:

query	""
matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerNETBiosName

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query**, **RegExp** }

Constructors

ComputerNETBiosName(Condition condition, string match, string query, bool matchCase, bool evaluateOnce, bool stopIfFails)

ComputerNETBiosName(Condition condition, string match)

Simplified constructor with the following defaults:

query	""
matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerOperatingSystem

Enums

enum **Condition** { **Equal**, **NotEqual** }

enum **Version** { **Any**, **Windows7**, **Windows8**, **Windows81**, **Windows10**, **WindowsServer2008R2**, **WindowsServer2012**, **WindowsServer2012R2**, **WindowsServer2016** }

enum **ServicePack** { **Any**, **None**, **SP1**, **SP2**, **SP3**, **SP4**, **SP5** }

enum **BuildNumberCondition** { **None**, **Equal**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual** }

enum **Edition** { **Any**, **Standard**, **Professional**, **Enterprise**, **Business**, **Ultimate** }

enum **CPUArchitecture** { **Any**, **Bit32**, **Bit64** }

enum **TSEnabled** { **Any**, **Yes**, **No** }

Constructors

EMConfigAPI.Conditions.ComputerOperatingSystem(Condition condition, Version version, ServicePack servicePack, Edition edition, CPUArchitecture cpuArchitecture, TSEnabled tsEnabled, bool stopIfFails)
Constructs a ComputerOperatingSystem object.

EMConfigAPI.Conditions.ComputerOperatingSystem(Condition condition, Version version, ServicePack servicePack, BuildNumberCondition buildNumberCondition, int buildNumber, Edition edition, CPUArchitecture cpuArchitecture, TSEnabled tsEnabled, bool stopIfFails)
Constructs a ComputerOperatingSystem object.

EMConfigAPI.Conditions.ComputerOperatingSystem(Condition condition, Version version)
Constructs a ComputerOperatingSystem object for a given Version and true for stopIfFails.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.ComputerProcessName

Enums

enum **Condition** { **Equal**, **NotEqual**, **RegExp** }

Constructors

ComputerProcessName(Condition condition, string match, string parameters, bool matchParameters, bool matchCase, bool stopIfFails)

ComputerProcessName(Condition condition, string match)

Simplified constructor with the following defaults:

parameters	""
matchParameters	false
matchCase	false
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.Counter

Constructors

Counter(int runChildrenTimesPerSession, bool stopIfFails)

Counter(int runChildrenTimesPerSession)

Simplified constructor with the following defaults:

stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.CustomCondition

Enums

enum **ScriptType** { **Powershell**, **JScript**, **VBScript** }

enum **RunAs** { **CurrentUser**, **System**, **Default** }

Constructors

CustomCondition(ScriptType type, string scriptText, RunAs runAs, int runFor, bool evaluateOnce, bool stopIfFails)

CustomCondition(ScriptType type, string scriptText)

Simplified constructor with the following defaults:

runAs	RunAs.Default
runFor	0

evaluateOnce	false
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.DirectoryClientComputerOUMembership

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query** }

Constructors

DirectoryClientComputerOUMembership(Condition condition, string match, bool includeSubOUsInMatch, string query, bool evaluateOnce, bool stopIfFails)

DirectoryClientComputerOUMembership(Condition condition, string match)

Simplified constructor with the following defaults:

includeSubOUsInMatch	true
query	""
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.DirectoryComputerOUMembership

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query** }

Constructors

DirectoryComputerOUMembership(Condition condition, string match, bool includeSubOUsInMatch, string query, bool evaluateOnce, bool stopIfFails)

DirectoryComputerOUMembership(Condition condition, string match)

Simplified constructor with the following defaults:

includeSubOUsInMatch	true
query	""
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.DirectorySiteMembership

Enums

enum **Condition** { **Equal**, **NotEqual**, **RegExp** }

Constructors

DirectorySiteMembership(Condition condition, string match, bool matchCase, bool evaluateOnce, bool stopIfFails)

DirectorySiteMembership(Condition condition, string match)

Simplified constructor with the following defaults:

matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.DirectoryUserOUMembership

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query** }

Constructors

DirectoryUserOUMembership(Condition condition, string match, bool includeSubOUsInMatch, string query, bool evaluateOnce, bool stopIfFails)

DirectoryUserOUMembership(Condition condition, string match)

Simplified constructor with the following defaults:

includeSubOUsInMatch	true
query	""
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.EnvironmentDateAndTime

Specifies various date and time conditions. The conditions required must be set by the methods after constructing the class. Multiple conditions (DaysOfWeek, Date and Time) may be set on the same condition if they are compatible.

Enums

enum **DayOfWeek** { **Sunday**, **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday** }

enum **When** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual** }

Constructors

EMConfigAPI.Conditions.EnvironmentDateAndTime(bool stopIfFails)

EMConfigAPI.Conditions.EnvironmentDateAndTime()
Construct with stopIfFails true.

Methods

void SetDaysOfWeek(params DayOfWeek[] daysOfWeek)

Sets the days of week condition. Accepts a list of days of week.

void SetDate(When when, DateTime date)

Sets a date condition. From PowerShell, specify the date with the month name, e.g. "25-Dec-2001". Numeric dates (e.g. 1/25/2001) are interpreted in the US format. This date condition is incompatible with a days of week condition.

The time part of the date is ignored.

void SetBetweenDates(DateTime date1, DateTime date2)

Makes the condition true between the indicated dates. This format is compatible with a days of week condition.

void SetTime(When when, DateTime date)

Sets a time condition. From PowerShell, specify as "hh:mm:ss"

The date part of the time is ignored.

void SetBetweenTimes(DateTime time1, DateTime time2)

Makes the condition true between the indicated times.

The date part of the time is ignored.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.EnvironmentSessionVariable

Compares a session variable with a given value or values.

Enums

enum **Condition** { **Equal**, **NotEqual**, **RegExp**, **Query**, **Between**, **NotBetween** }

Constructors

EMConfigAPI.Conditions.EnvironmentSessionVariable(Condition condition, string name, string value, string value2, bool stopIfFails)
Constructs an EnvironmentSessionVariable object.

EMConfigAPI.Conditions.EnvironmentSessionVariable(Condition condition, string name, string value, bool stopIfFails)
Constructs an EnvironmentSessionVariable object with an empty string for value2. Value2 is used to specify a second value when using the Between and NotBetween operators.

EMConfigAPI.Conditions.EnvironmentSessionVariable(Condition condition, string name, string value)
Constructs an EnvironmentSessionVariable object with an empty string for value2 and true for stopIfFails.

EMConfigAPI.Conditions.EnvironmentSessionVariable(Condition Condition, string name, string value, string value2)
Constructs an EnvironmentSessionVariable object to compare a session variable to a range, passing true to stopIfFails.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.EnvironmentVariable

Enums

enum **Condition** { **Equal**, **NotEqual**, **Exist**, **NotExist**, **Contains** }

Constructors

EnvironmentVariable(Condition condition, string name, string value, bool stopIfFails)

EnvironmentVariable(Condition condition, string name, string value)

Simplified constructor with the following defaults:

stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.FileExistsCondition

Validates the existence of a file.

Constructors

EMConfigAPI.Conditions.FileExistsCondition(string exists, bool path, DateCondition dateCondition, SizeCondition sizeCondition, bool stopIfFails)

EMConfigAPI.Conditions.FileExistsCondition(string exists, bool path, DateCondition dateCondition, SizeCondition sizeCondition)

EMConfigAPI.Conditions.FileExistsCondition(string exists, bool path)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

Helper class: EMConfigAPI.FileAndFolder.DateCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileDateProperty** { **Created**, **LastModified**, **LastAccessed** }

Constructors

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, DateTime startDate, DateTime endDate)
Constructs a DateCondition object for the given criteria.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when)
Constructs a DateCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

FileDateProperty DateProperty { get; set; }

DateTime StartDate { get; set; }

DateTime EndDate { get; set; }

Helper class: EMConfigAPI.FileAndFolder.SizeCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileSizeUnits** { **B**, **KB**, **MB**, **GB** }

Constructors

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when, ulong startSize, FileSizeUnits startSizeUnits, ulong endSize, FileSizeUnits EndSizeUnits)
Constructs a SizeCondition object for the given criteria.

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when)
Constructs a SizeCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

ulong StartSize { get; set; }

FileSizeUnits StartSizeUnits { get; set; }

ulong EndSize { get; set; }

FileSizeUnits EndSizeUnits { get; set; }

ulong StartSizeInBytes { get; }

ulong EndSizeInBytes { get; }

EMConfigAPI.Conditions.FileTextFileSearch

Enums

enum **Condition** { **Contains**, **NotContains** }

Constructors

FileTextFileSearch(string file, Condition condition, string text, bool useRegularExpressions, bool matchCase, bool stopIfFails)

FileTextFileSearch(string file, Condition condition, string text)

Simplified constructor with the following defaults:

useRegularExpressions	false

matchCase	false
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.FolderExistsCondition

Validates the existence of a folder.

Constructors

EMConfigAPI.Conditions.FolderExistsCondition(string exists, bool path, DateCondition dateCondition, SizeCondition sizeCondition, bool stopIfFails)

EMConfigAPI.Conditions.FolderExistsCondition(string exists, bool path, DateCondition dateCondition, SizeCondition sizeCondition)

EMConfigAPI.Conditions.FolderExistsCondition(string exists, bool path)

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

Helper class: EMConfigAPI.FileAndFolder.DateCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileDateProperty** { **Created**, **LastModified**, **LastAccessed** }

Constructors

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when, DateTime startDate, DateTime endDate)
Constructs a DateCondition object for the given criteria.

EMConfigAPI.FileAndFolder.DateCondition(FileDateProperty property, ConditionOperator when)
Constructs a DateCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

FileDateProperty DateProperty { get; set; }

DateTime StartDate { get; set; }

DateTime EndDate { get; set; }

Helper class: EMConfigAPI.FileAndFolder.SizeCondition

A Condition object, based on date criteria, which can be passed to the Action to only process files which match the condition.

Enums

enum **ConditionOperator** { **Equal**, **NotEqual**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual**, **Between** }

enum **FileSizeUnits** { **B**, **KB**, **MB**, **GB** }

Constructors

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when, ulong startSize, FileSizeUnits startSizeUnits, ulong endSize, FileSizeUnits EndSizeUnits)
Constructs a SizeCondition object for the given criteria.

EMConfigAPI.FileAndFolder.SizeCondition(ConditionOperator when)
Constructs a SizeCondition object with the CompareToDestination property implicitly set to true.

Properties

ConditionOperator When { get; set; }

bool CompareToDestination { get; set; }

ulong StartSize { get; set; }

FileSizeUnits StartSizeUnits { get; set; }

ulong EndSize { get; set; }

FileSizeUnits EndSizeUnits { get; set; }

ulong StartSizeInBytes { get; }

ulong EndSizeInBytes { get; }

EMConfigAPI.Conditions.IsLaptop

Constructors

IsLaptop(bool isLaptop, bool evaluateOnce, bool stopIfFails)

IsLaptop()

Simplified constructor with the following defaults:

isLaptop	true
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.IsvDisk

Checks the vDisk activity of a connecting client.

Enums

enum **VDiskMode** { **Any**, **Standard**, **Private** }

Constructors

EMConfigAPI.Conditions.IsvDisk(bool IsActive, VDiskMode Mode)
Creates a condition based on whether or not an endpoint has a vDisk active and is in a specified mode

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.RegistryKeyExists

Enums

enum **Condition** { **Exist**, **NotExist**, **Contains** }

Constructors

RegistryKeyExists(string hive, string key, Condition comparison, bool stopIfFails)

RegistryKeyExists(string hive, string key, Condition comparison)

Simplified constructor with the following defaults:

stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.RegistryValue

Tests a registry value. For all types, if the comparison is Exist, NotExist or ValueTypeExist, the Value property is ignored. To set the default value, specify ValueName as an empty string (or use the UseDefault property). The constructor tests only REG_SZ values - use the properties to test other types of values. Valid comparison values for REG_SZ are 'Equal, NotEqual'.

Enums

enum **Comparison** { **Equal**, **NotEqual**, **Exist**, **NotExist**, **ValueTypeExist**, **Contains**, **NotContains**, **LessThan**, **LessThanOrEqual**, **GreaterThan**, **GreaterThanOrEqual** }

Constructors

RegistryValue(string hive, string key, string valueName, Comparison comparison, string value, bool stopIfFails)

RegistryValue(string hive, string key, string valueName, Comparison comparison, string value)

Simplified constructor with the following defaults:

stopIfFails	true

Properties

bool UseDefault { set; }
Set to test the (Default) value rather than the named value. Same as setting ValueName to empty.

UInt32 ValueDWORD { set; }
Set a DWORD value (overriding previous value) to test against. Valid comparison values are 'Equal, NotEqual, LessThan, LessThanOrEqual, GreaterThan, GreaterThanOrEqual'.

UInt64 ValueQWORD { set; }
Set a QWORD value (overriding previous value) to test against. Valid comparison values are 'Equal, NotEqual, LessThan, LessThanOrEqual, GreaterThan, GreaterThanOrEqual'.

string ValueEXPAND_SZ { set; }
Set a REG_EXPAND_SZ value (overriding previous value) to test against. Valid comparison values are 'Equal, NotEqual'.

string[] ValueMULTI_SZ { set; }
Supply an array of strings to set a REG_MULTI_SZ value (overriding previous value) to test against. Valid comparison values are 'Equal, NotEqual, Contains, NotContains'.

byte[] ValueBINARY { set; }
Supply an array of bytes to set a REG_BINARY value (overriding previous value). Valid comparison values are 'Equal, NotEqual'.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.UserGroupMembership

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query** }

Constructors

UserGroupMembership(Condition condition, string match, string query, string groupSID, bool evaluateOnce, bool stopIfFails)

UserGroupMembership(Condition condition, string match, string query)

Simplified constructor with the following defaults:

groupSID	""
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.UserIsAdministrator

Constructors

UserIsAdministrator(bool isAdministrator, bool evaluateOnce, bool stopIfFails)

UserIsAdministrator(bool isAdministrator)

Simplified constructor with the following defaults:

evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.UserName

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query**, **RegExp** }

Constructors

UserName(Condition condition, string match, string query, bool matchCase, bool evaluateOnce, bool stopIfFails)

UserName(Condition condition, string match)

Simplified constructor with the following defaults:

query	""
matchCase	false
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.UserPrimaryGroup

Enums

enum **Condition** { **Equal**, **NotEqual**, **Query** }

Constructors

UserPrimaryGroup(Condition condition, string match, string query, string groupSID, bool evaluateOnce, bool stopIfFails)

UserPrimaryGroup(Condition condition, string match, string query)

Simplified constructor with the following defaults:

groupSID	""
evaluateOnce	true
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.UserProcessName

Enums

enum **Condition** { **Equal**, **NotEqual**, **RegExp** }

Constructors

UserProcessName(Condition condition, string match, string parameters, bool matchParameters, bool matchCase, bool stopIfFails)

UserProcessName(Condition condition, string match)

Simplified constructor with the following defaults:

parameters	""
matchParameters	false
matchCase	false
stopIfFails	true

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes

EMConfigAPI.Conditions.VMwareVariables

Compares one or more VMware variables. To specify a variable to be compared, supply the operator to use for the comparison and the value that the variable should be compared to. Set the operator to None to ignore (and not compare) the variable. The condition will evaluate to true if all of the individual variables compare successfully.

Enums

enum **BrokerUrlCondition** { **None**, **Equal**, **NotEqual**, **Contains** }

enum **BrokerDNSNameCondition** { **None**, **Equal**, **NotEqual**, **Contains** }

enum **BrokerRemoteIPAddressCondition** { **None**, **Equal**, **NotEqual**, **Between**, **NotBetween** }

enum **BrokerTunnelUrlCondition** { **None**, **Equal**, **NotEqual**, **Contains** }

enum **BrokerIsTunneledCondition** { **None**, **Equal** }

enum **ClientIpAddressCondition** { **None**, **Equal**, **NotEqual**, **Between**, **NotBetween** }

enum **ClientLoggedOnDomainNameCondition** { **None**, **Equal**, **NotEqual**, **Query** }

enum **ClientLoggedOnUserNameCondition** { **None**, **Equal**, **NotEqual**, **Query** }

enum **ClientMacAddressCondition** { **None**, **Equal** }

enum **ClientMachineNameCondition** { **None**, **Equal**, **NotEqual**, **Query** }

enum **ClientTypeCondition** { **None**, **Equal**, **NotEqual** }

Constructors

Methods

void SetBrokerUrl(BrokerUrlCondition operator, System.String value)

Compares the Broker Url variable using the 'operator' to the given 'value'.

void SetBrokerDNSName(BrokerDNSNameCondition operator, System.String value)

Compares the Broker DNS Name variable using the 'operator' to the given 'value'.

void SetBrokerRemoteIPAddress(BrokerRemoteIPAddressCondition operator, System.String value, System.String endValue)

Compares the Broker Remote IP Address variable using the 'operator' to the given 'value'. When using the 'Between' operator, an endValue should be supplied.

void SetBrokerTunnelUrl(BrokerTunnelUrlCondition operator, System.String value)

Compares the Broker Tunnel Url variable using the 'operator' to the given 'value'.

void SetBrokerIsTunneled(BrokerIsTunneledCondition operator, System.Boolean value)

Compares the Broker Is Tunneled variable using the 'operator' to the given 'value'.

void SetClientIpAddress(ClientIpAddressCondition operator, System.String value, System.String endValue)

Compares the Client Ip Address variable using the 'operator' to the given 'value'. When using the 'Between' operator, an endValue should be supplied.

void SetClientLoggedOnDomainName(ClientLoggedOnDomainNameCondition operator, System.String value)

Compares the Client Logged On Domain Name variable using the 'operator' to the given 'value'.

void SetClientLoggedOnUserName(ClientLoggedOnUserNameCondition operator, System.String value)

Compares the Client Logged On User Name variable using the 'operator' to the given 'value'.

void SetClientMacAddress(ClientMacAddressCondition operator, System.String value)

Compares the Client Mac Address variable using the 'operator' to the given 'value'.

void SetClientMachineName(ClientMachineNameCondition operator, System.String value)

Compares the Client Machine Name variable using the 'operator' to the given 'value'.

void SetClientType(ClientTypeCondition operator, System.String value)

Compares the Client Type variable using the 'operator' to the given 'value'.

Inherited ActionBase Properties

string Name { set; }
Overwrites the default name of the condition

string Description { set; }
Sets the condition description

string Notes { set; }
Sets the condition notes