**SCOM Management Pack Creator Guide**

**By Dujon Walsham**

Contents

[Introduction 4](#_Toc531767994)

[How does it Work 4](#_Toc531767995)

[How to use the script 5](#_Toc531767996)

[PowerShell Script Steps 5](#_Toc531767997)

[Create Visual Studios Project Steps 8](#_Toc531767998)

[Functions 9](#_Toc531767999)

[**New-SCOMMPClass** 9](#_Toc531768000)

[**Add-SCOMMPClass** 9](#_Toc531768001)

[Switches 9](#_Toc531768002)

[**Add-SCOMMPClassProperty** 9](#_Toc531768003)

[Switches 10](#_Toc531768004)

[**Add-SCOMMPRunAsAccount** 10](#_Toc531768005)

[Switches 10](#_Toc531768006)

[**Edit-SCOMMPClass** 10](#_Toc531768007)

[**New-SCOMMPRelationship** 10](#_Toc531768008)

[**Add-SCOMMPRelationship** 10](#_Toc531768009)

[Switches 11](#_Toc531768010)

[**New-SCOMMPDiscovery** 11](#_Toc531768011)

[Switches 11](#_Toc531768012)

[**Add-SCOMMPPowerShellDiscovery** 11](#_Toc531768013)

[Switches 11](#_Toc531768014)

[**Add-SCOMMPRegistryDiscovery** 12](#_Toc531768015)

[Switches 12](#_Toc531768016)

[**Add-SCOMMPRegistryKey** 13](#_Toc531768017)

[Switches 13](#_Toc531768018)

[**Edit-SCOMMPRegistry** 13](#_Toc531768019)

[**Add-SCOMMPWMIDiscovery** 13](#_Toc531768020)

[Switches 13](#_Toc531768021)

[**Add-SCOMMPVBScriptDiscovery** 14](#_Toc531768022)

[Switches 14](#_Toc531768023)

[**Add-SCOMMPComputerGroupDiscovery** 14](#_Toc531768024)

[Switches 15](#_Toc531768025)

[**Add-SCOMMPInstanceGroupDiscovery** 15](#_Toc531768026)

[Switches 15](#_Toc531768027)

[**Create-PowerShellScript** 15](#_Toc531768028)

[Switches 15](#_Toc531768029)

[**Create-VBScript** 16](#_Toc531768030)

[Switches 16](#_Toc531768031)

[Edit-SCOMMPDiscovery 16](#_Toc531768032)

[**New-SCOMMPView** 16](#_Toc531768033)

[Switches 16](#_Toc531768034)

[**Add-SCOMMPView** 16](#_Toc531768035)

[Switches 16](#_Toc531768036)

[**New-SCOMMPFolder** 17](#_Toc531768037)

[Switches 17](#_Toc531768038)

[**Add-SCOMMPFolder** 17](#_Toc531768039)

[Switches 17](#_Toc531768040)

[**Edit-SCOMMPViewsFolders** 17](#_Toc531768041)

[**New-SCOMMPMonitorRule** 17](#_Toc531768042)

[**New-SCOMMPCustomProbeAction** 17](#_Toc531768043)

[**Add-SCOMMPCustomProbeAction** 17](#_Toc531768044)

[Switches 18](#_Toc531768045)

[**Add-SCOMMPWindowsEventMonitor** 18](#_Toc531768046)

[Switches 18](#_Toc531768047)

[**Add-SCOMMPWindowsServiceMonitor** 19](#_Toc531768048)

[Switches 19](#_Toc531768049)

[**Add-SCOMMPWindowsServicePerformanceMonitor** 20](#_Toc531768050)

[Switches 20](#_Toc531768051)

[**Add-SCOMMPWindowsGenericLogMonitor** 21](#_Toc531768052)

[Switches 21](#_Toc531768053)

[**Add-SCOMMPPerformanceMonitor** 22](#_Toc531768054)

[Switches 22](#_Toc531768055)

[**Add-SCOMMPWindowsEventRule** 23](#_Toc531768056)

[Switches 23](#_Toc531768057)

[**Add-SCOMMPWindowsPowerShellScriptRule** 24](#_Toc531768058)

[Switches 24](#_Toc531768059)

[**Add-SCOMMPPerfomanceRule** 24](#_Toc531768060)

[Switches 24](#_Toc531768061)

[**Edit-SCOMMPMonitorRule** 26](#_Toc531768062)

[**Reload-SCOManagementPack** 26](#_Toc531768063)

[Functions 26](#_Toc531768064)

# Introduction

I had created many customized management packs using Visual Studios within combination of the VSAE Authoring extensions application.

Writing an entire management can take a lot of time depending on the complexity requirements such as

* Classes
* Discoveries
* Views
* Monitors
* Rules

There have been methods to speed up and aid in the productivity such as using the VSAE authoring guide available on TechNet, opening a sealed management pack to look at the XML code and also using the existing codes from a management pack which you yourself had created.

One of the issues I sometimes faced was creating Monitors, Rules or discoveries using the templates as sometimes if you make edits directly to the XML file which can control anyone of these it will then be overwritten if you attempt to change anything inside of the template files used to create any of those specific objects within your management pack.

This lead me to develop my own solution which would end up being a PowerShell script called the “**SCOM Management Pack Creation Script**”

# How does it Work

This script has the capability to build an entire SCOM management pack from scratch using its XML codes and intellisense answers.

Here are the objects in which can be created

·        Classes

·        Properties for classes (additional attributes)

·        Secure References (Run As Accounts)

·        Discoveries

·        Script for discoveries (PowerShell, VBScript, Registry & WMI)

·        Views & Folders

·        Monitors & Rules

The script has the ability to format the entire XML code the exact same way you would see it shown within the Visual Studios application when creating a new project to create a management Pack.

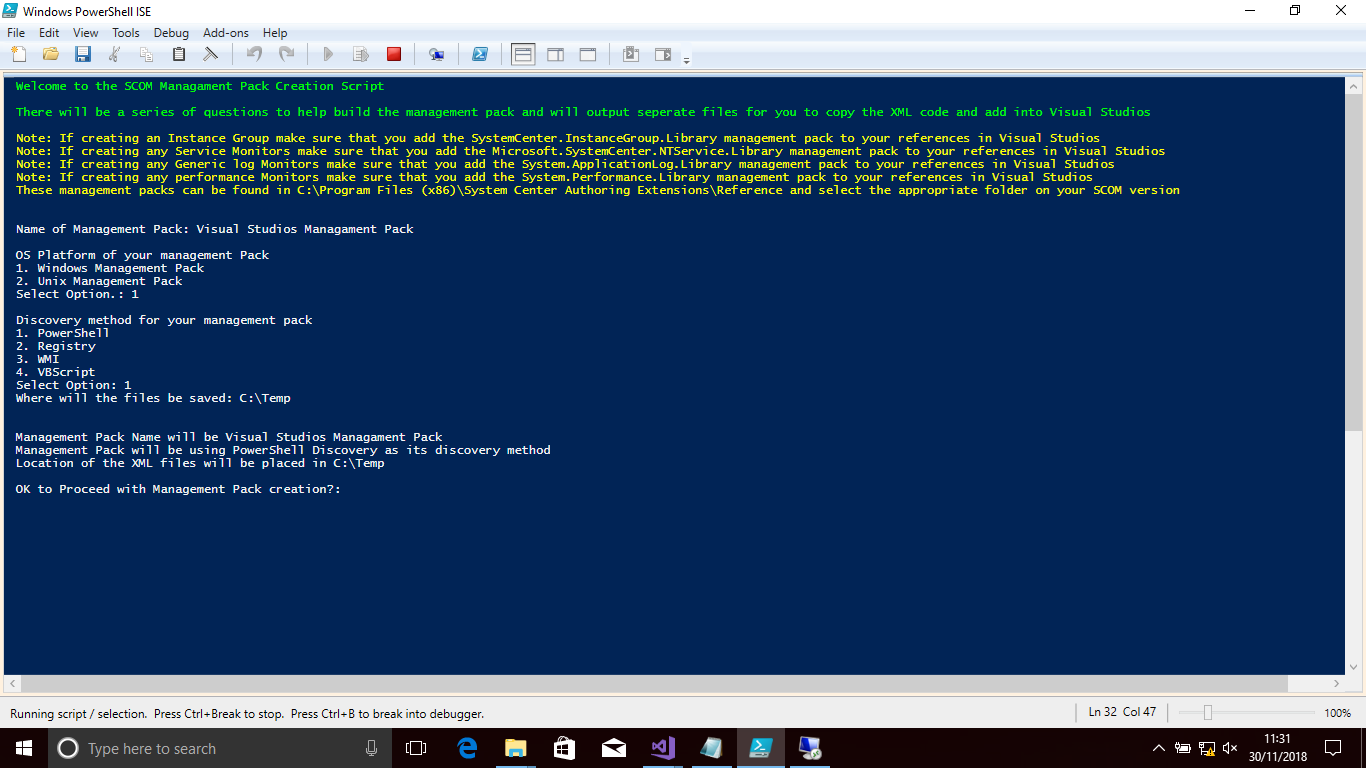
After the script has ran its course and you have answered all of the questions it will output all of the XML files depending on the type of object you have created and you will be able to copy all of the information from the XML file into a new Visual Studio project which will be able to build the management pack.

# How to use the script

Once you run the management pack this is how the process will go

(**Note:** Ensure when you are creating any classes or anything make sure you go through the wizard and enter it again ***exactly as you did it*** as it is case sensitive and it can stop the management pack from building successfully)

### PowerShell Script Steps



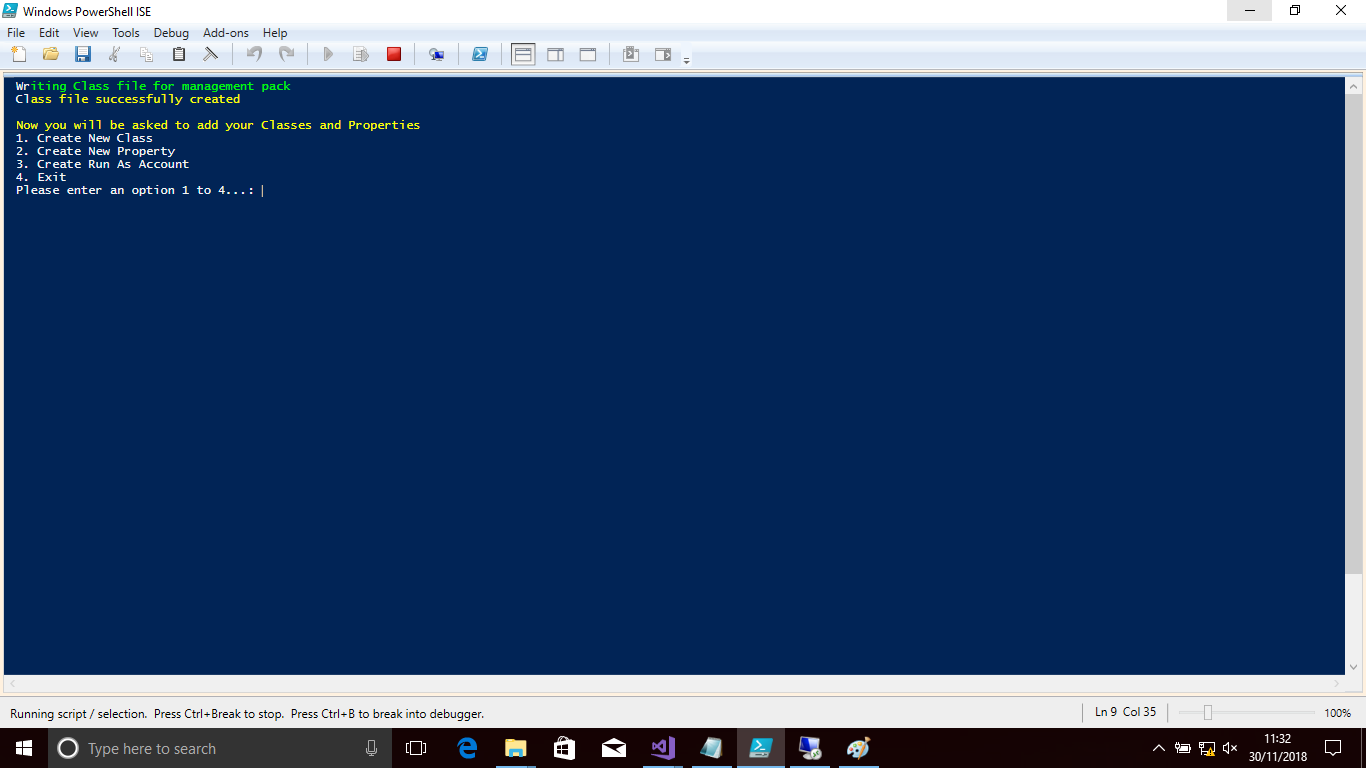
**Figure 1.1 –** Starting page of the PowerShell Script which gives you the following questionnaire

1.      Enter the name of the Management Pack

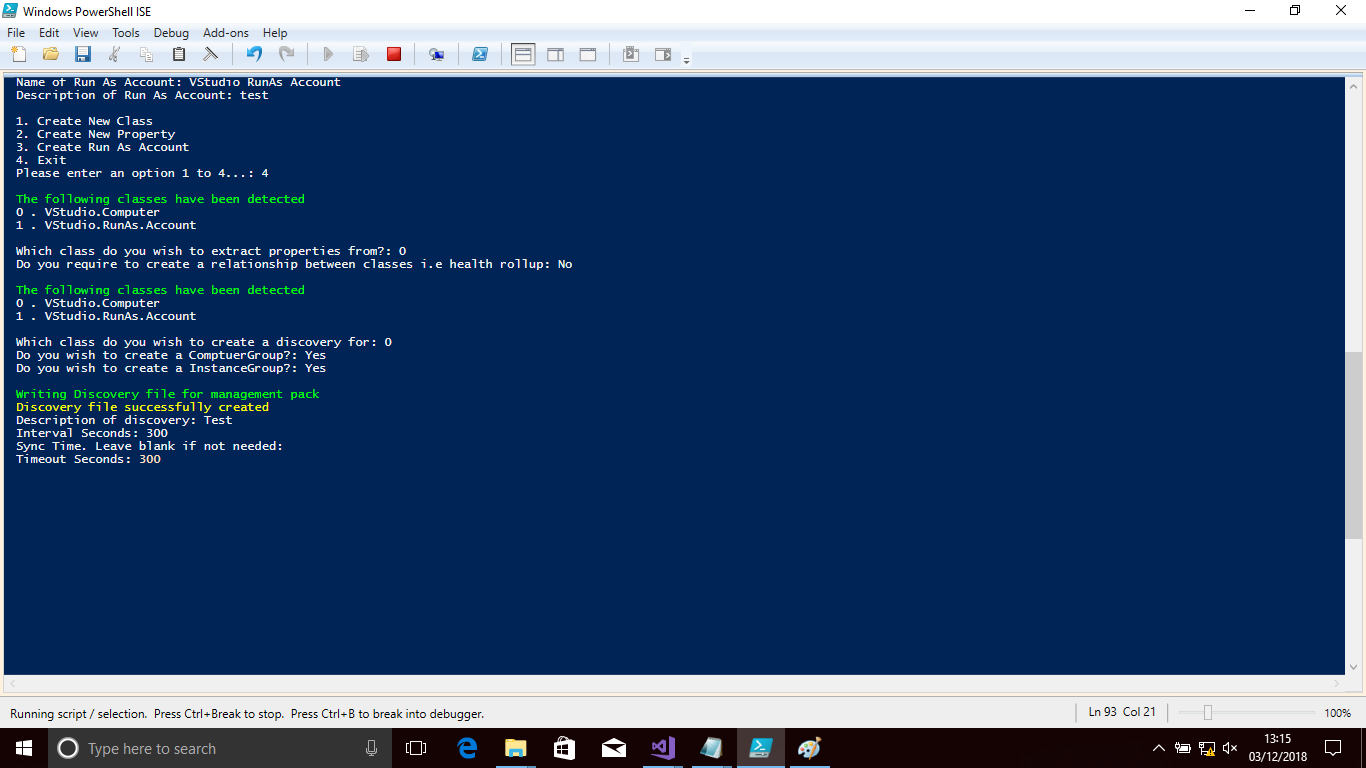
2.      Enter the platform type which is either a “Windows Management Pack” or “Unix Management Pack”

3.      Enter the discovery method which is either “PowerShell”, “VBScript”, “WMI” or “Registry”

4.      Enter the location of where all of the XML files will be created. And then you will be presented with a summary which will show how your management pack will be configured overall. Type yes



**Figure 1.2 –** Displaying the options to create your classes. A result from **Edit-SCOMMPClass**

5.      You will go through the creating of the class object, which you can add multiple classes, properties for each class and Run As Accounts.  
  
  
**Figure 1.3 –** Shows wizard going through the Classes and Discovery options

6.      Asked to extract properties from a specific class to build the discovery around (**Note:** Depending on the discovery method you chose it will create the template for it i.e. PowerShell, it will create the PowerShell lines to capture the information but you would need to enter your initial script into it for it to be able to work)

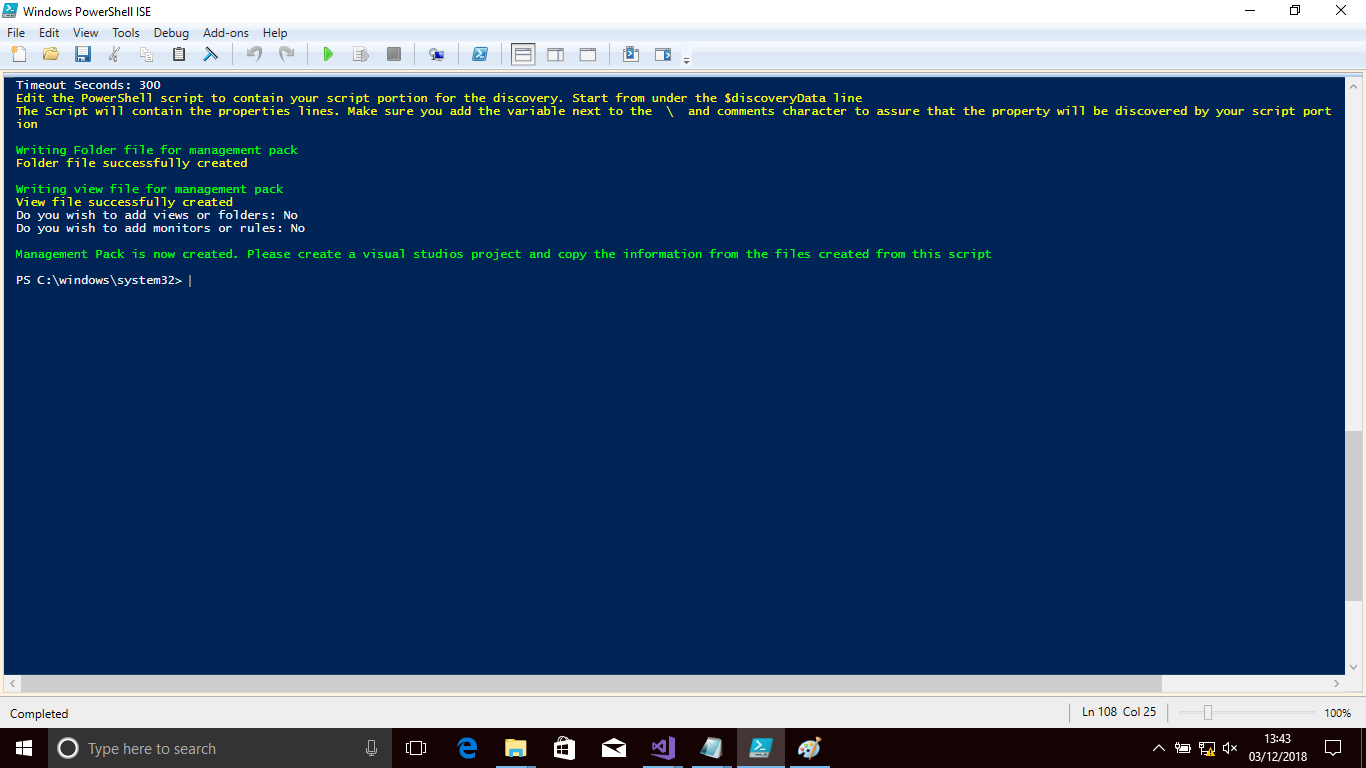
7.      Asks to create a ComputerGroup for your class

8.      Asks to create an InstanceGroup for your class

9.      Creates the Management Pack folder

10.  Enter details to create the views

11.  Asks if you wish to add more folders or views

12.  Asks if you wish to create any Monitors or Rules  
  
  
**Figure 1.4 –** Shows wizard going through the Views, Folders and completion

13.  If yes it will ask to create a Monitor or Rule and then go into the questions for it

14.  Message states the management pack is now created

### Create Visual Studios Project Steps

Once this is done you will need to create a new Visual Studios project if you haven’t done so already and do the following to create the objects.

* Right click your project and choose to create a new object
* Choose the Empty Fragment file and give it an appropriate name according to the XML files you created
* Open the XML files I.e. class file and copy the XML from this and into the Empty Fragment file which you have created
* §Create the script file (if PowerShell or VBScript) in your Visual Studios project and name it exactly as it is on your XML output.
* Copy the information from your outputted script file into the new script you created for your new management pack.
* Build the management pack when ready

# Functions

Here are a list of all of the functions alongside its switches which can be used to build out the management pack.

(**Note:** The Script solution already has a pre-configured questionnaire, so these functions not only allow you to understand each function, but also if you wish to run the functions individually or to edit certain parts in your XML files)

## **New-SCOMMPClass**

This function is used to create the XML file which will contain all of the information for your SCOM management pack Classes, properties and Run as Accounts

## **Add-SCOMMPClass**

This functions allow you to create a new class within the class file which would have been created from the New-SCOMMPClass function

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| ClassName | Name of the class | String value i.e. Visual Studio Computer |
| ClassType | Type of class | List of numerical options given to select type |
| ClassDescription | Description of the class | String value i.e. computer which has visual studios installed |
| Abstract | Will this class be abstract | Boolean value – true or false. |
| Hosted | Will this class be a hosted class | Boolean value – true or false. |
| Singleton | Will this class be a singleton class | Boolean value – true or false. |
| MPClassFile | Location of where the XML file will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPClassProperty**

This function allows you to create properties for the class which you have created which will be the CI attribute information that would be captured alongside its discovery

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| PropertyName | Name of the Property | Name of the property i.e. Version |
| PropertyType | Type of property | List of numerical options to choose the type of property data |
| KeyValue | Is the value a key value | Boolean value – true or false |
| PropertyDescription | Description of the property | String value i.e. Version of Visual Studios installed |
| AffectedClassID | The class which this property is linked to | List of numerical options to choose from the classes you had just created |
| MPClassFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPRunAsAccount**

This function will add a Run as Account to your class file

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| SecureReferenceName | Name of the Run As Account | String value i.e. Visual Studio RunAs Account |
| SecureReferenceDescription | Description of the Run As Account | String value i.e. Run As Account used to discover Visual Studios |
| MPClassFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Edit-SCOMMPClass**

This functions provides a switch list of all of the functions which are used to formulate your classes in your SCOM Management pack. It will list the following functions

* Create New Class (Add-SCOMMPClass)
* Create New Property (Add-SCOMMPClassProperty)
* Create Run As Account (Add-SCOMMPRunAsAccount)
* Exit

## **New-SCOMMPRelationship**

This function creates a relationship between two classes which is needed if you are looking to have nested classes which will be used to reference others when creating further discoveries or monitors

## **Add-SCOMMPRelationship**

This function adds a relationship class to your relationship container that you would have created on the New-SCOMMPRelationship function

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| RelationshipName | Name of the Relationship | String value i.e. Visual Studio Computer User Relationship |
| RelationshipDescription | Description of the relationship class | String value i.e. Relationship between Visual Studio computer and user |
| Abstract | Will this class be abstract | Boolean value – true or false |
| Accessibility | What is the accessibility of the class | String value – Internal or Public |
| SourceType |  | List of numerical options to choose from the classes you had just created |
| TargetType |  | List of numerical options to choose from the classes you had just created |
| MPRelationshipFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **New-SCOMMPDiscovery**

This function creates the discovery class for which you will use to discover all of your objects within your SCOM Management Pack

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| DiscoveryTarget | ClassID which the discovery will run for | String value i.e. Visual.Studio.Computer |
| MPDiscoveryFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPPowerShellDiscovery**

Creates a PowerShell based discovery for your discovery container which would use a PowerShell script to perform the object discovery within your management pack.

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| DiscoveryName | Name of the discovery | String value |
| DiscoveryTarget | Which class will the discovery run for | List of numerical options to choose from the classes you had just created |
| DiscoveryDescription | Description of the discovery | String value |
| DiscoveryClass | Discovery Class Name | List of numerical options to choose from the classes you had just created |
| DiscoveryRunAsAccount | Run As account used to run the discovery | List of numerical options to choose from the classes you had just created |
| IntervalSeconds | Time to run the discovery | Integer i.e. 300 |
| SyncTime | Specifcy a time to run at (optional) | Time i.e. 03:00 |
| ScriptName | Name of the script | String value i.e. VisualStudioPSDiscovery.ps1 |
| ScriptBody | Script body content |  |
| TimeoutSeconds | Time it takes to no longer run | Integer i.e. 300 |
| MPClassFile | Location of the class file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPDiscoveryFile | Location of where the discovery file will be created | String value i.e. C:\Temp\Visualstudioclass.xml |
| ClassID | Pass through details of the class used | List of numerical options to choose from the classes you had just created |

## **Add-SCOMMPRegistryDiscovery**

This function creates a Registry discovery which uses a Registry key to identify if an object belongs to the class you have created for your SCOM Management Pack

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| DiscoveryName | Name of the discovery | String value |
| DiscoveryTarget | Which class will the discovery run for | List of numerical options to choose from the classes you had just created |
| DiscoveryDescription | Description of the discovery | String value |
| DiscoveryClass | Discovery Class Name | List of numerical options to choose from the classes you had just created |
| DiscoveryRunAsAccount | Run As account used to run the discovery | List of numerical options to choose from the classes you had just created |
| Frequency | Time to run the discovery | Integer i.e. 300 |
| MPClassFile | Location of the class file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPDiscoveryFile | Location of where the discovery file will be created | String value i.e. C:\Temp\Visualstudioclass.xml |
| ClassID | Pass through details of the class used | List of numerical options to choose from the classes you had just created |

## **Add-SCOMMPRegistryKey**

This function creates the Registry key which will be added to your Registry Discovery which will be used to interrogate the classes’ registry to see if there is a match

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| AttributeName | Name of the discovery | String Value. Visual Studio KeyExists |
| RegistryPath | Path of where the Key is in the registry | String Value i.e. HKLM\Software |
| PathType | Type of path discovery | Integer value – 0 for key existing and 1 for discovering attribute |
| AttributeType | Attribute Type | List of numerical options to choose from the different view types |
| ClassID | Class which the Registry discovery will be ran against | String value i.e. Visual Studio ClassID |

## **Edit-SCOMMPRegistry**

This functions gives you the option to add Registry Attributes to your Registry discovery as you will want to add multiple if looking to discover these for your properties which are outlined in the class which you had created in your class file.

## **Add-SCOMMPWMIDiscovery**

This functions creates a WMI discovery which will utilise a WMI query that will be used against the class object WMI to discover the object for the designated class

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| DiscoveryName | Name of the discovery | String value |
| DiscoveryTarget | Which class will the discovery run for | List of numerical options to choose from the classes you had just created |
| DiscoveryDescription | Description of the discovery | String value |
| DiscoveryClass | Discovery Class Name | List of numerical options to choose from the classes you had just created |
| DiscoveryRunAsAccount | Run As account used to run the discovery | List of numerical options to choose from the classes you had just created |
| Namespace | WMI namespace to connect to | String value i.e. \\root\cimv2 |
| Query | Query to use for WMI | String value i.e. select \* from win32\_operatingsystem |
| Frequency | Time to run the discovery | Integer value i.e. 300 |
| ClassID | Pass through details of the class used | String value i.e. Visual Studio ClassID |

## **Add-SCOMMPVBScriptDiscovery**

This function creates a VB Script discovery which will use a VB Script to run the discovery for your designated class

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| ClassID | Pass through details of the class used | String value i.e. Visual Studio ClassID |
| DiscoveryName | Name of the discovery | String value i.e. Visual Studio Discovery |
| DiscoveryTarget | Which class will the discovery run for | List of numerical options to choose from the classes you had just created |
| DiscoveryDescription | Description of the discovery | String value i.e. Discovery using VBScript |
| DiscoveryClass | Discovery Class Name | List of numerical options to choose from the classes you had just created |
| DiscoveryRunAsAccount | Run As account used to run the discovery | List of numerical options to choose from the classes you had just created |
| IntervalSeconds | Time to run the discovery | Integer i.e. 300 |
| SyncTime | Specifcy a time to run at (optional) | Date Time i.e. 03:00 |
| MPClassFile | Location of the class file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPDiscoveryFile | Location of where the discovery file will be created | String value i.e. C:\Temp\Visualstudioclass.xml |
| ScriptName | Name of the script | String value i.e. Vbscript.vbs |
| TImeoutSeconds | Time it takes to no longer run | Integer i.e. 300 |

## **Add-SCOMMPComputerGroupDiscovery**

This function creates a computer group for all of the objects which have been discovered by your SCOM Management Pack.

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| DiscoveryName | Name of the discovery | String value i.e. Visual Studio Discovery |
| DiscoveryTarget | Which class will the discovery run for | List of numerical options to choose from the classes you had just created |
| DiscoveryDescription | Description of the discovery | String value i.e. Discovery using VBScript |
| ClassID | Pass through details of the class used | String value i.e. Visual Studio ClassID |
| DiscoveryFile | Discovery file location where the computer group discovery will be added | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPInstanceGroupDiscovery**

This function creates an instance group for all of the instances which have been discovered by your SCOM Management Pack.

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| DiscoveryName | Name of the discovery | String value i.e. Visual Studio Discovery |
| DiscoveryTarget | Which class will the discovery run for | List of numerical options to choose from the classes you had just created |
| DiscoveryDescription | Description of the discovery | String value i.e. Discovery using VBScript |
| ClassID | Pass through details of the class used | String value i.e. Visual Studio ClassID |
| DiscoveryFile | Discovery file location where the computer group discovery will be added | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Create-PowerShellScript**

This function is used to create a PowerShell script for your PowerShell discovery. It adds all of the lines required for the discovery template script to work.

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| ScriptName | Name of the script | String value i.e. VisualStudioPSDiscovery.ps1 |
| MPClassFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Create-VBScript**

This function is used to create a VB script for your VBScript discovery. It adds all of the lines required for the discovery template script to work.

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| ScriptName | ClassID which the discovery will run for | String value i.e. VisualStudioPSDiscovery.vbs |
| MPClassFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## Edit-SCOMMPDiscovery

This function is used to add additional discoveries

## **New-SCOMMPView**

This function creates the View XML file so that you are able to create views for your classes in your SCOM Management pack.

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MPViewFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPView**

This function creates new views and adds them to your view XML file which was created using the New-SCOMMPView function

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| ViewName | Name of the View | String value i.e. Visual Studio Alerts View |
| ViewTarget | Class which the view will be created for | List of numerical options to choose from the classes you had just created |
| ViewType | Type of view to create | List of numerical options to choose from the different view types |
| FolderID | Folder ID for where the view will be placed | List of numerical options to choose from the folders you had just created |
| MPViewFile | Output location of the View XML file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **New-SCOMMPFolder**

This function creates folders xml file for where you can store your different views within you SCOM Management Pack

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MPFolderFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPFolder**

This function allows you to create different views and store them in your views XML which you had just created with the New-SCOMMPFolder function

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| FolderName | Name of the folder to create | String value i.e. Visual Studio Views |
| FolderParent | The folder ID of where the folder will be placed. By default it will go to the Root of the SCOM Console | List of numerical options to choose from the folders you had just created |
| MPFolderFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Edit-SCOMMPViewsFolders**

This function provides a list of the following functions

* Create New View (Add-SCOMMPView)
* Create New Folder (Add-SCOMMPFolder)

## **New-SCOMMPMonitorRule**

This function creates the Monitor/Rules XML file which allows you to add additional Monitors and Rules to

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MPMonitorRuleFile | Location of where the XML files will be created | String value i.e. C:\Temp\Visualstudioclass.xml |

## **New-SCOMMPCustomProbeAction**

This function is used to create the XML file for a Custom probe action monitor. Primarily used in this script as a PowerShell Script Monitor

## **Add-SCOMMPCustomProbeAction**

This function adds the following

* Probe Action Type
* Data Source Module
* Unit Monitor Type

These allow you to use a PowerShell script which can be used within a monitor.

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| CustomModuleName | Name of the custom monitor you will use | String value i.e. Visual Studio PowerShellMonitor |
| MonitorTarget | ID of your Class that the monitor will be used with | List of numerical options to choose from the classes you had just created |
| RunAsAccount | ID of the Run As Account that will be used | List of numerical options to choose from the classes you had just created |
| AlertName | Name of the alert | String value i.e. ComputerIssue |
| AlertMessage | Message of the alert | String value i.e. Computer is not working |
| TimeoutSeconds | Timeout in Seconds | Integer i.e. 300 |

## **Add-SCOMMPWindowsEventMonitor**

This function allows you to create a Windows Event 2 state monitor

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MontiorName | Name of monitor | String value i.e. Visual Studio EventMonitor |
| MonitorEnabled | If monitored is enabled by default or not “true” or “false” | Boolean value – true or false |
| MonitorTarget | Class ID which the monitor will be targeted to | List of numerical options to choose from the classes you had just created |
| MonitorRunAsAccount | Run as account which will run the monitor | List of numerical options to choose from the classes you had just created |
| AlertOnState | If it should alert on Error or Warning | String Value – Error or Warning |
| AlertSeverity | Alert Severit | String Value – Error or Warning |
| AlertPriority | Alert Priority, High, Normal or Low | String Value – Error or Warning |
| Alert Name | Name of the Alert | String value i.e. Visual Studio Event Monitor Bad Health |
| AlertMessage | Message shown when alert is created | String value i.e. Visual Studio Event Monitor in bad health call administrator. |
| UnHealthyLogName | Log Name for unhealthy status | String Value i.e Application |
| UnHealthyPublisherName | Publisher name for unhealthy status | String Value i.e. VisualStudio |
| UnHealthyEventDisplayNumber | Event Number for unhealthy status | Integer Value i.e. VisualStudio |
| HealthyLogName | Log Name for healthy status | String Value i.e Application |
| HealthyPublisherName | Publisher name for healthy status | String Value i.e. VisualStudio |
| HealthyEventDisplayNumber |  | Integer Value i.e. VisualStudio |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPWindowsServiceMonitor**

This function will create a Windows Service Monitor that will monitor if a windows service is up or down

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MontiorName | Name of monitor | String value i.e. Visual Studio EventMonitor |
| MonitorEnabled | If monitored is enabled by default or not “true” or “false” | Boolean value – true or false |
| MonitorTarget | (Options being WindowsComputer, UnixComputer, ComputerGroup, InstanceGroup, WindowsApplicationComponent or WindowsLocalApplication) | List of numerical options to choose from the classes you had just created |
| MonitorRunAsAccount | Run as account which will run the monitor | List of numerical options to choose from the classes you had just created |
| AlertOnState | If it should alert on Error or Warning | String Value – Error or Warning |
| AlertSeverity | Alert Severity, Error or Warning | String Value – Error or Warning |
| AlertPriority | Alert Priority, High, Normal or Low | String Value – Error or Warning |
| Alert Name | Name of the Alert | String value i.e. Visual Studio Event Monitor Bad Health |
| AlertMessage | Message shown when alert is created | String value i.e. Visual Studio Event Monitor in bad health call administrator. |
| ServiceName | Name of the service to monitor | String value – i.e. AppInfo |
| AlertOnAuto | If it will monitor an automatic service | Boolean value – true or false |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPWindowsServicePerformanceMonitor**

This function will create a Windows Service performance monitor which will monitor the resources that a Windows Service uses i.e. CPU, Memory, Logical disk etc

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MontiorName | Name of monitor | String value i.e. Visual Studio EventMonitor |
| MonitorEnabled | If monitored is enabled by default or not “true” or “false” | Boolean value – true or false |
| MonitorTarget | (Options being WindowsComputer, UnixComputer, ComputerGroup, InstanceGroup, WindowsApplicationComponent or WindowsLocalApplication) | List of numerical options to choose from the classes you had just created |
| MonitorRunAsAccount | Run as account which will run the monitor | List of numerical options to choose from the classes you had just created |
| AlertOnState | If it should alert on Error or Warning | String Value – Error or Warning |
| AlertSeverity | Alert Severity, Error or Warning | String Value – Error or Warning |
| AlertPriority | Alert Priority, High, Normal or Low | String Value – Error or Warning |
| Alert Name | Name of the Alert | String value i.e. Visual Studio Event Monitor Bad Health |
| AlertMessage | Message shown when alert is created | String value i.e. Visual Studio Event Monitor in bad health call administrator. |
| ServiceName | Name of the service to monitor | String value – i.e. AppInfo |
| AlertOnAuto | If it will monitor an automatic service | Boolean value – true or false |
| CounterName | Counter name of the resource to monitor |  |
| ObjectName | Object name of the resource to monitor | String value i.e. LogicalDisk |
| Frequency | Frequency to which the monitor will run | Integer value i.e. 300 |
| Threshold | Threshold max it should reach before alerting | Interger value i.e. 50 |
| NumSamples | Number of samples to compare with | Integer value i.e. 5 |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPWindowsGenericLogMonitor**

This function will create a Windows generic log file monitor which can monitor for patterns on log files which can alert if found

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MontiorName | Name of monitor | String value i.e. Visual Studio EventMonitor |
| MonitorEnabled | If monitored is enabled by default or not “true” or “false” | Boolean value – true or false |
| MonitorTarget | (Options being WindowsComputer, UnixComputer, ComputerGroup, InstanceGroup, WindowsApplicationComponent or WindowsLocalApplication) | List of numerical options to choose from the classes you had just created |
| MonitorRunAsAccount | Run as account which will run the monitor | List of numerical options to choose from the classes you had just created |
| AlertOnState | If it should alert on Error or Warning | String Value – Error or Warning |
| AlertSeverity | Alert Severity, Error or Warning | String Value – Error or Warning |
| AlertPriority | Alert Priority, High, Normal or Low | String Value – Error or Warning |
| Alert Name | Name of the Alert | String value i.e. Visual Studio Event Monitor Bad Health |
| AlertMessage | Message shown when alert is created | String value i.e. Visual Studio Event Monitor in bad health call administrator. |
| LogFileDirectory | Location of the log files | String value i.e. C:\Temp\Logs |
| LogPattern | Log file extension files | Strng value i.e. \*.log |
| LogIsUTF8 | Is the log format of UTF8 | Boolean value – true or false |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPPerformanceMonitor**

This function will create a Windows performance monitor which will monitor the resources such as i.e. CPU, Memory, Logical disk etc

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MontiorName | Name of monitor | String value i.e. Visual Studio EventMonitor |
| MonitorEnabled | If monitored is enabled by default or not “true” or “false” | Boolean value – true or false |
| MonitorTarget | (Options being WindowsComputer, UnixComputer, ComputerGroup, InstanceGroup, WindowsApplicationComponent or WindowsLocalApplication) | List of numerical options to choose from the classes you had just created |
| MonitorRunAsAccount | Run as account which will run the monitor | List of numerical options to choose from the classes you had just created |
| AlertOnState | If it should alert on Error or Warning | String Value – Error or Warning |
| AlertSeverity | Alert Severity, Error or Warning | String Value – Error or Warning |
| AlertPriority | Alert Priority, High, Normal or Low | String Value – Error or Warning |
| Alert Name | Name of the Alert | String value i.e. Visual Studio Event Monitor Bad Health |
| AlertMessage | Message shown when alert is created | String value i.e. Visual Studio Event Monitor in bad health call administrator. |
| ComputerName | Where to get computer name value | String value i.e. $Target/Property[Type=”MicrosoftWindowsLibrary!Microsoft.Windows.Computer”]/NetworkName$ |
| ObjectName | Name of object to resource monitor | String value i.e. LogicalDisk |
| AllInstances | Monitor all instances of the object/counter | Boolean value – true or false |
| Frequency | Frequency the monitor runs | Integer value i.e. 30 |
| Threshold | Threshold in percentage | Integer value i.e. 40 |
| InstanceName |  |  |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPWindowsEventRule**

This function will create a Windows Event Rule. This has the ability to suppress alerts depending on the variables

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| RuleName | Name of Rule | String value i.e. Visual Studio PowerShellRule |
| RuleEnabled | If the rule is enabled by default | Boolean value – true or false |
| RuleTarget | Class ID to which the rule will apply to | List of numerical options to choose from the classes you had just created |
| RuleRunAsAccount | Run as account which will run the Rule | List of numerical options to choose from the classes you had just created |
| AlertOnState | If it should alert on Error or Warning | String Value – Error or Warning |
| AlertSeverity | Alert Severity, Error or Warning | String Value – Error or Warning |
| AlertPriority | Alert Priority, High, Normal or Low | String Value – Error or Warning |
| Alert Name | Name of the Alert | String value i.e. Visual Studio Event Monitor Bad Health |
| AlertMessage | Message shown when alert is created | String value i.e. Visual Studio Event Monitor in bad health call administrator. |
| ServiceName | Name of the service to monitor | String value – i.e. AppInfo |
| AlertOnAuto | If it will monitor an automatic service | Boolean value – true or false |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPWindowsPowerShellScriptRule**

This function will create a Windows PowerShell Script rule

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| RuleName | Name of Rule | String value i.e. Visual Studio PowerShellRule |
| RuleEnabled | If the rule is enabled by default | Boolean value – true or false |
| RuleTarget | Class ID to which the rule will apply to | List of numerical options to choose from the classes you had just created |
| RuleRunAsAccount | Run as account which will run the Rule | List of numerical options to choose from the classes you had just created |
| Alert Name | Name of the Alert | String value i.e. Visual Studio Event Monitor Bad Health |
| Alert Message | Message which the alert will display | String value i.e. Visual Studio Event Monitor in bad health call administrator. |
| Severity | Alert Severity | List of numerical options to choose from the classes you had just created |
| Priority | Alert Priority | List of numerical options to choose from the classes you had just created |
| Interval Seconds | Interval which the rule will run | Integer value i.e. 300 |
| SyncTime | Time in which the rule will run | Time value i.e. 03:00 |
| TimeoutSeconds | Interval which the rule will run | Integer value i.e. 300 |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Add-SCOMMPPerfomanceRule**

This function will create a Windows Performance Rule

### Switches

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| RuleName | Name of Rule | String value i.e. Visual Studio PowerShellRule |
| RuleEnabled | If the rule is enabled by default | Boolean value – true or false |
| RuleTarget | Class ID to which the rule will apply to | List of numerical options to choose from the classes you had just created |
| RuleRunAsAccount | Run as account which will run the Rule | List of numerical options to choose from the classes you had just created |
| Alert Name | Name of the Alert | String Value i.e Visual Studio AlertTooHigh |
| AlertMessage | Message shown when alert is created | String Value ie. Alert has breached threshold |
| ComputerName | Where to obtain computer name | String value i.e. $Target/Property[Type=”MicrosoftWindowsLibrary!Microsoft.Windows.Computer”]/NetworkName$ |
| CounterName | Name of counter | String value i.e. % Disk Time |
| ObjectName | Name of object to resource monitor | String value i.e. LogicalDisk |
| AllInstances | Monitor all instances of the object/counter | Boolean value – true or false |
| Frequency | Frequency the monitor runs | Integer value i.e. 30 |
| Threshold | Threshold of performance monitor | Integer value i.e. 300 |
| Tolerance | Tolerance based on percentage | Integer value i.e 10 |
| MaxSampleSeparation | Maximum samples to compare with | Integer value i.e. 4 |
| MPMonitorRuleFile | Output for XML for Monitor file | String value i.e. C:\Temp\Visualstudioclass.xml |

## **Edit-SCOMMPMonitorRule**

This function will give a list of all of the Monitors and Rules which you can create so that you can add additional ones each time.

## **Reload-SCOManagementPack**

This function allows you to reload your entire management pack by the location of your files so that it can provide you with a list of options to add additions to your SCOM Management Pack.

### Functions

|  |  |  |
| --- | --- | --- |
| Name | Description | Expected Value |
| MPClassFile | Location of class XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPDiscoveryFile | Location of discover XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPFolderFile | Location of Folder XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPViewFile | Location of View XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPMonitorRuleFile | Location of Monitor Rule XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPRelationShipFle | Location of Relationship XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPCustomProbeActionFile | Location of Custom Probe Action XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPCustomDataSourceFile | Location of Custom Data Source XML file | String value i.e. C:\Temp\Visualstudioclass.xml |
| MPCustomMonitorTypeFile | Location of Custom Monitor Type XML file | String value i.e. C:\Temp\Visualstudioclass.xml |