Lab LC09: Automating SMS Tasks with Monad Scripting

Objectives
These labs show you how to use Monad to automate software distribution and other SMS operational tasks. You will learn basic Monad commands and apply them to the SMS 2003 WMI, COM and .NET managed SMS site server interfaces to complete various SMS tasks.

After completing these exercises, you will be able to write Monad script code to perform the following tasks:

- Create a package, program and advertisement.
- Read, modify and delete a package.
- List advertisements and programs.
- View and run queries.
- Modify SMS site control settings.
- Summarize a site’s health.
- View and modify the advanced client cache.
- Check the status of the advanced client software distribution agent.
- Use WMI CIM Studio to view and modify object data.

Prerequisites
Prior to starting this lab, a Microsoft Windows Server 2003 computer installed as an SMS primary site server. Optionally, a Microsoft Windows XP Professional computer installed as an Advanced Client in the SMS site.

The SMS site code used as an example in this lab will be MCM. If using a different site code please make sure to replace MCM throughout the exercises.

Important: The Virtual Images used for these labs have been pre-configured to allow the use of the managed SMS site server interfaces documented in the SMS SDK 3.1 documentation.

If you are running these labs on another machine:


2. Run the following command-line in the Monad shell to load the assembly in the shell each time that the shell is started:
Estimated time to complete this lab: 75 minutes
Exercise 0
Preparing the Virtual Computer Clients for the Lab

In this exercise, you will update the collection membership of the All Systems collection. When you do, you will see the SMSClient computer appear twice in the membership list. This is due to Virtual PC 2004 virtualizing the SMBios serial number of the host computer, which SMS detects as being different than the original SMBios serial number. Because of this, SMS generates a new GUID for the client, which causes a new record to be generated.

Note  Complete this procedure from the primary site server computer only.

atoiTo update the collection membership

1. Log on as administrator with a password of password.
2. On the Start menu, click SMS Administrator Console.
   The SMS Administrator Console window appears.
3. In the console tree, expand Site Database, expand Collections, and then click All Systems.
   The members of the All Systems collection appear in the details pane. Notice that the site server computer (SMSServer) and the Windows XP Professional client computer (SMSClient) appear as members.
4. On the Action menu, point to All Tasks, and then click Update Collection Membership.
   The All Systems message box appears prompting to update subcollection membership.
5. Click OK, and then on the Action menu, click Refresh.
   The collection membership is updated, and the current membership of the All Systems collection is displayed. Notice that the SMSClient computer is now displayed twice. Notice also that one of the instances is listed as being Obsolete and inactive. This is the old reference of the client.
6. In the details pane, click the topmost record for the SMSClient computer, which should be listed as an Obsolete client (scroll to the right in the details pane) and then on the Action menu, click Delete.
   A Confirm Delete message box appears prompting to delete the record.
7. Click Yes.
   The collection membership is updated, and the current membership of the All Systems collection is displayed. Notice that the SMSClient computer is now displayed only once.
8. Delete any other obsolete records from the All Systems collection.
9. Update the membership of the All Windows XP Systems collection.
   This collection will be used later in this lab for targeting of an advertisement.
   You have now prepared your images for the lab and may proceed to Exercise 1.
Exercise 1
Connecting to the SMS Provider

In this exercise, you will write Monad script to connect to the SMS Provider for an SMS Site. The script connects to the SMS Provider using the SMS SDK Managed Site Server Interfaces. The SMSProvider object is used to make a connection to the SMS Provider. SMSProvider is used to access SMS Objects such as programs, packages and queries. You will use this script to make a connection for the following exercises.

To Connect to the SMS Provider:

1. Open Notepad and enter the following lines of code to capture the username and password from the command-line:

   # Get username and password from the command-line, provide error if nothing is entered
   param(  
       [string] $UserName = $(throw "Enter your user name"),  
       [string] $Password = $(throw "Enter your password")  
   )

2. Enter the following line of code to connect to the local SMS site server:

   # Reference the namespaces to use.
   $oProvider = new-object Microsoft.SystemsManagementServer.Automation.SMSProvider ".", $UserName, $Password

3. Temporarily add some test code to verify the connection by pulling some data from the local site. Enter the following code at the end of the script:

   # Test code to verify connection to the local site
   # Remove test code once basic connection code is verified
   #
   # Get the site settings for the local site.
   $oSite = $oProvider.GetSiteSettings()
   # Display the site code of the site.
   write-host "SiteCode="$oSite.SiteCode

4. Save the file in c:\scripts with the name get-connection.msh.
5. Click Start->Run, type msh and click OK to start a command window.
6. Type CD c:\scripts to enter the script directory
7. Type c:\scripts\get-connection.msh and press Enter to run the script.
8. If you added the additional test code, you should see SiteCode = MCM in the command window.
9. If there are any error messages fix them before continuing. You will use this code in several of the following procedures.
10. Once `get-connection.msh` runs correctly, remove the test code and save the script.
Exercise 2
Create a Package, Program and Advertisement

In this exercise, you use the SMSProvider object that you obtained in Exercise 1 to create a package (Test Package), program (Test Program) and advertisement (Test Advertisement). You will use the SMS Administrators console to see that the package, program and advertisement have been created.

Note  Complete this procedure from the primary site server computer

To create a new package
1. Create a new text file with Notepad.
2. Enter the code from exercise 1 to connect to the SMS Provider.
3. Enter the following code to create the package (Test Package).
   # Create a package.
   $oPackage = $oProvider.Packages.Create("Test Package")
   # Set some basic settings on the package.
   $oPackage.Description = "Created by script"
   $oPackage.PkgSourcePath = "C:\\PkgSrc"
   $oPackage.UseCompressedSource = $false
   $oPackage.Save()
   # Write status back to console
   write-host "Created package"

To create a new program
4. Enter the following code to create the program (Test Program).
   # Create a program.
   $oProgram = $oPackage.Programs.Create("Test Program", "notepad.exe")
   # Set some basic settings on the program.
   $oProgram.Comment = "Contact helpdesk for this program"
   $oProgram.RunOnAnyPlatform = $true
   $oProgram.Save()
   # Write status back to console
   write-host "Created program"

To create a new advertisement
5. Enter the following code to create the advertisement (Test Advertisement).
   # Get the All Systems Collection
   $oCollection = $oProvider.Collections.Get("SMS00001")

   # Create an advertisement.
$oAdvert = $oProvider.Advertisements.Create("Test Advertisement", $oCollection, $oProgram)

# Set some basic settings on the advertisement.
$oAdvert.AllowUserToRunIndependent = $true
$oAdvert.RunASAP = $true
$oAdvert.Save()

# Write status back to console
write-host "Created advertisement"

6. Save the file as create-packageprogramadvertisement.msh in C:\scripts.

7. In the command window opened in the previous procedure enter
   c:\scripts\create-packageprogramadvertisement.msh to run the script.

8. Open the SMS Administrator console

9. Click on the **Packages** node. You should see the package you created under
   the **Packages** node. You should also see the package in the right pane. If
   you do not see the package try Right-Clicking the **Packages** node and select
   **Refresh**

10. In the left tree view Right Click the newly created package and select **Properties**. In the dialog box displayed you should see the name of the
    package in the **Name** edit box and the description in the **Comment** box.

11. Click **Cancel** to close the properties dialog box.

12. In the right hand pane Double Click **Programs**.

13. You should see the program you created listed. Right Click the program
    name and select **Properties**.

14. In the properties dialog box displayed you should see the name and other
    properties that you set in the program creation script.

15. Click **Cancel** to close the properties dialog box.

16. Click on the **Advertisement** node. You should see the advertisement you
    created under the **Advertisement** node. You should also see the
    advertisement in the right pane. If you do not see the advertisement try
    Right-Clicking the **Advertisement** node and select **Refresh**

17. In the left tree view Right Click the newly created package and select **Properties**.

18. In the properties dialog box displayed you should see the name and other
    properties that you set in the advertisement creation script.

19. Click **Cancel** to close the properties dialog box.
Exercise 3
Read, Modify and Delete Packages

In this exercise, you use the SMSProvider object that you obtained in Exercise 1 to view available packages, modify a package and delete a package.

In the following procedure you will view the collection of package objects.

**Note** Complete this procedure from the primary site server computer.

 выполняется:

---

**To read available packages**

1. Create a new text file with Notepad.
2. Enter the code from exercise 1 to connect to the SMS Provider.
3. Enter the following code to get the SMS package objects (**SMS_Package**)
   ```powershell
   # Get packages.
   $oPackages = $oProvider.Packages
   ```
4. Enter the following code to enumerate through the collection of program objects and display the program name and other information:
   ```powershell
   Foreach ($oPackage in $oPackages)
   {
       write-host $oPackage.PackageID
       write-host $oPackage.Name
       write-host $oPackage.Description
   }
   ```
5. Save the file as **get-packages.msh** in **C:\scripts**.
6. In the command window type **c:\scripts\get-packages.msh** and press enter to run the script.
7. You should see the available packages listed.
8. Make a note of the **PackageID** property displayed. You will need it in the next procedure.
In the following procedure, you will modify the previously created package.

**Note** Complete this procedure from the primary site server computer.

.lock

**To modify a package**

1. Create a new text file with Notepad.
2. Enter the code from exercise 1 to connect to the SMS Provider.
3. Enter the following code to get an SMS package using the PackageID saved from the previous procedure.
4. Enter the following code that opens the previously created package object. Replace `PACKAGEID` with the package ID you wrote down in the previous procedure.
   
   ```powershell
   # Get package.
   $oPackages = $oProvider.Packages
   $oPackage=$oProvider.Packages.Get("PACKAGEID")
   ```
5. Enter the following code to change the package description.
   
   ```powershell
   # Change package description
   $oPackage.Description="Modify Package Description"
   ```
6. Enter the following code to save the package changes.
   
   ```powershell
   # Save package changes
   $oPackage.save()
   ```
7. Save the script in C:\scripts as a file named `modify-package.msh`.
8. In the command window type `c:\scripts\modify-package.msh` and press enter to run the script.
9. Open the SMS Administrator console
10. Click on the Packages node. You should see the package you created under the Packages node. You should also see the package in the right pane. If you do not see the package try Right-Clicking the Packages node and select Refresh
11. In the left tree view Right Click the package and select Properties. In the dialog box displayed you should see the updated description in the Comment box.
12. Click Cancel to close the properties dialog box.
13. Run the script, `get-packages.msh` that you created in the previous procedure. You should see that package description has changed.
In the following procedure, you will delete the previously created package.

**Note** Complete this procedure from the primary site server computer.

라면 To delete a package

1. Create a new text file with Notepad.
2. Enter the code from exercise 1 to connect to the SMS Provider.
3. Enter the following code to get an SMS package using the PackageID saved from the previous procedure.
   ```
   # Get package.
   $oPackages = $oProvider.Packages
   $oPackage=$oProvider.Packages.Get("PACKAGEID")
   ```
4. Enter the following code to delete the package.
   ```
   # Delete package.
   $oPackage.delete()
   ```
5. Save the script in C:\scripts in a file named “delete-package.msh”
6. In the command window type c:\scripts\delete-package.msh and press enter to run the script.
7. Run the script.
8. Open the SMS Administrator console
9. Right Click on the Packages node and press Refresh. You should see that the package has been deleted.
10. Run the script, get-packages.msh, that you created previously. You should see that the package has been deleted.
Exercise 4
List Programs and Advertisements

In this exercise, you use the Monad `get-wmiobject` cmdlet to display available programs and advertisements.

**Note**  Complete this procedure from the primary site server computer

 Kafka To list available programs

1. Create a new text file with Notepad.
2. Enter the following code to get the SMS program objects (SMS_Program) by using the `get-wmiobject` cmdlet.

   ```powershell
   # Set computer variable to local and get SMS_Program object
   $strComputer = ".";
   $oPrograms = get-wmiobject -class "SMS_Program" -namespace "root\sms\site_mcm" -computername $strComputer
   ```

3. Enter the following code to enumerate through the collection of program objects and display the program name:

   ```powershell
   # List each program
   foreach ($oProgram in $oPrograms)
   {
   write-host $oProgram.ProgramName
   }
   ```

4. Save the file as `get-programs.msh` in C:\scripts.
5. In the command window type `c:\scripts\get-programs.msh` and press enter to run the script.
6. You should see the available programs listed.
7. Try changing the script to display other properties set when the program was created.
To list available advertisements

8. Create a new text file with **Notepad**.

9. Enter the following code to get the SMS advertisement objects (SMS_Advertisement) by using the **get-wmiobject** cmdlet.
   
   ```powershell
   # Set computer variable to local and get SMS_Advertisement object
   $strComputer = "."
   $oAdvertisements = get-wmiobject -class "SMS_Advertisement" -namespace "root\sms\site_mcm" -computername $strComputer
   ```

10. Enter the following code to enumerate through the collection of program objects and display the program name:

   ```powershell
   # List each program
   foreach ($oAdvertisement in $oAdvertisements)
   {
       write-host "Advertisement Name:" $oAdvertisement.AdvertisementName
   }
   ```

11. Save the file as **get-advertisements.msh** in **C:\scripts**.

12. In the command window type **c:\scripts\get-advertisements.msh** and press enter to run the script.

13. You should see the available advertisements listed.

14. Try changing the script to display other properties set when the advertisement was created.
Exercise 4
Query SMS

In this exercise, you use the SMSProvider object that you obtained in Exercise 1 to get a list of available queries and run a query on the SMS site.

**Note**  Complete this exercise from the primary site server.

❑ **To view available queries**
1. Create a new file with Notepad.
2. Enter the code from exercise 1 to connect to the SMS Provider.
3. Enter the following code to get a list of available queries (Queries).
   
   # Get queries.
   
   $oQueries = $oProvider.Queries

4. Enter the following code to display each available query.
   
   foreach ($oQuery in $oQueries)
   {
      write-host $oQuery.QueryID
      write-host $oQuery.Name
   }

5. Save the script in C:\scripts to a file named get-queries.msh
6. In the command window type c:\scripts\get-queries.msh and press enter to run the script.
7. You should see a list of available queries displayed.
8. Make a note of a few QueryID values. You will use them in the next procedure.

**Note**  Complete this exercise from the primary site server.

❑ **To run a specific query**
1. Create a new file with Notepad.
2. Enter the code from exercise 1 to connect to the SMS Provider.
3. Enter the following code to get the required query. The string is the identifier for the query. SMS001 represents All Systems
   
   # Get query.
   
   $oQueries = $oProvider.Queries
   $oQuery = $oProvider.Queries.Get("SMS001")

4. Enter the following code to run the query.
   
   # Execute query.
$oConnection = $oProvider.Connection
$collQueryResults = $oConnection.ExecuteQuery($oQuery.SQLExpression)

5. Enter the following code to list the results of the query.
   # List results of query.
   Foreach ($oResult in $collQueryResults)
   {
       write-host $oResult.Name
   }

6. Save the script to a file named `run-query.msh`

7. In the command window type `c:\scripts\run-query.msh` and press enter to
    run the script.

8. You should see a list of all systems displayed.

9. Change `strQueryID` to the one of the `QueryID` values you noted in the
    previous procedure.

10. Run the query script again to see the query run.
Exercise 5
Modify SMS Site Control File Settings

In this exercise, you use the **SMSProvider** object that you obtained in Exercise 1 to modify the SMS Site Control File.

**Note** Complete this exercise from the primary site server.

⚠️ To change site control file settings

1. Create a new file with **Notepad**.
2. Enter the code from exercise 1 to connect to the SMS Provider.
3. Enter the following to get the site settings.
   ```powershell
   # Get the site settings for the local site.
   $oSite = $oProvider.GetSiteSettings()
   ```
4. Enter the following code to display the site code.
   ```powershell
   # Display the site code of the site.
   Write-Host "SiteCode=" $oSite.SiteCode
   ```
5. Enter the following change the comment for the site server.
   ```powershell
   # Change the comment for the site server.
   $oSite.Comment = "Modified Comment"
   ```
6. Enter the following to add an Active Directory site boundary to the site server.
   ```powershell
   # Check to see if the site server already contains the CustomADSite Active Directory site boundary
   if($oSite.SiteBoundaries.ADSites.Contains("CustomADSite") -eq $false)
   {
     # Add the site boundary to the site server if it doesn’t already exist.
     $ADSite = new-object Microsoft.SystemsManagementServer.Automation.SMSSiteBoundaryADSite "CustomADSite"
     $oSite.SiteBoundaries.ADSites.Add($ADSite)
   }
   # Save the changes to the site boundaries.
   $oSite.SiteBoundaries.SaveSettings()
   }
7. Save the script in **C:\scripts** to a file named **modify-sitecontrolfilessettings.msh**
8. In the command window type **c:\scripts\modify-sitecontrolfilessettings.msh** and press enter to run the script.
9. Verify the site control file changes by inspecting the site control file, or reviewing the relevant setting in the Admin UI.
Exercise 6
Determining SMS Site Health

In this exercise, you use the Monad `get-wmiobject` cmdlet to display the health of the SMS site.

Note Complete this procedure from the primary site server computer.

네 To determine the site’s health
1. Create a new file with Notepad.
2. Enter the following code to get the SMS site status by using the `get-wmiobject` cmdlet.
   ```powershell
   # Set computer variable to local and get SMS_Program object
   $strComputer = "."
   $colSummarizerSiteStatus = get-wmiobject -class "SMS_SummarizerSiteStatus" -namespace "root\sms\site_mcm" -computername $strComputer
   ```
3. Enter the following code to display the overall site health summary
   ```powershell
   foreach ($objSiteSummary in $colSummarizerSiteStatus) {
       write-host "Site code: " $objSiteSummary.SiteCode
       if ($objSiteSummary.Status -eq 0) {
           write-host "Site is healthy"
       }
       if ($objSiteSummary.Status -eq 1) {
           write-host "There are warnings for the site"
       }
       if ($objSiteSummary.Status -eq 2) {
           write-host "Site health is critical"
       }
   }
   ```
4. Save the script in `C:\scripts` to a file named `get-sitehealth.msh`
5. In the command window type `c:\scripts\get-sitehealth.msh` and press enter to run the script.
6. The site’s overall health will be displayed.
Exercise 7
Using the Advanced Client Automation Objects

In this exercise you will use the SMS COM Automation Objects to view and adjust Advanced Client Settings.

Note  Complete this exercise from the primary site server or advanced client computer.

To view the size of the Advanced Client cache
1. Create a new file with Notepad.
2. Enter the following code to get the cache object from the software distribution object.
   ```ps
   $oUIResManager = new-object -com UIResource.UIResourceMgr
   $oCache = $oUIResManager.GetCacheInfo()
   ```
3. Enter the following code to display the current size of the cache and the amount of free space in the cache.
   ```ps
   write-host "Free Cache Space: ", $oCache.FreeSize "MB"
   write-host "Total Cache Size: ", $oCache.TotalSize "MB"
   ```
4. Save the script to C:\scripts in a file named get-cache.msh
5. In the command window type c:\scripts\get-cache.msh and press enter to run the script.

In the following procedure you will use the Advanced Client Automation cache object to set the size of the cache.

Note  Complete this exercise from the primary site server or advanced client computer.

To set the size of the Advanced Client cache
1. Create a new file with Notepad.
2. Enter the following code to get the cache object from the software distribution object.
   ```ps
   $oUIResManager = new-object -com UIResource.UIResourceMgr
   $oCache = $oUIResManager.GetCacheInfo()
   ```
3. Enter the following code to change the size of the Advanced Client cache to 300mb.
   ```ps
   $oCache.TotalSize = 300
   ```
4. Enter the following code to list the new cache size.
   ```ps
   # List new cache size
   write-host "New Cache Size: ", $oCache.TotalSize "MB"
   ```
5. Save the script in C:\scripts to a file named set-cache.msh
6. In the command window type `c:\scripts\set-cache.msh` and press enter to run the script.

7. Click **Start->Control Panel** to start the control panel.

8. Double Click on the Systems Management Icon


10. Note the size of the cache.

11. Try running the script with different cache sizes.

In the following procedure you will use the Advanced Client Automation software distribution client settings object to get the state of the client software distribution agent.

**Note** Complete this exercise from the primary site server or advanced client computer.

Lisa to the state of the software distribution agent

1. Create a new file with **Notepad**.

2. Enter the following code to get the software distribution client settings object from the software distribution object.

   ```powershell
   $oUIResManager = new-object -com UIResource.UIResourceMgr
   $oSWDist = $oUIResManager.GetSWDistClientSettings()
   ```

3. Enter the following code to determine is the software distribution agent is enabled.

   ```powershell
   if ($oSWDist.SWDistEnabled -eq 1)
   {
       write-host "Software Distribution: Enabled"
   }
   else
   {
       write-host "Software Distribution: Disabled"
   }
   ```

4. Save the script in `C:\scripts` to a file named `get-swdstate.msh`

5. In the command window type `c:\scripts\get-swdstate.msh` and press enter to run the script.

6. Note the state of the software distribution agent.
Exercise 8
Using CIM Studio to view and modify WMI Data

In this exercise you will use the WMI CIM Studio tool to view WMI classes and namespaces in the CIM repository.

**Note**  Complete this exercise from the primary site server.

✓ **To connect to the primary site’s namespace in CIM Studio**
1. Run WMI CIM Studio by going to Start/Programs/WMI Tools/WMI CIM Studio.
2. In the **Connect to namespace** box type `root\sms\site_mcm` and click **OK**.
3. In the **WMI CIM Studio Logon** box click **OK** to login as the current user.

✓ **To view created packages**
1. To find the SMS_Packages class click on the **Search for Class** icon.
2. Type **SMS_Package** in the **Search for Class** text box and click **Go!**
3. Several items will appear in the **Search results** box. Double click on the **SMS_Package** item.
   
   Note: Notice that the SMS_Package class is listed in the left pane under the SMS_BaseClass node. Searching is one way to quickly find classes or they can be found by navigating the trees in the left pane. Also notice that in the right pane all of the objects for the given class are listed.
4. To list all Instances for the SMS_Package object click on the **Instances** icon. All package instances should now appear in the right pane.
5. Double click on one of the packages to view all properties and associated values for the specific package. Make a note of the **PackageID** property value for the package.

✓ **To modify an existing package**
6. Find the **Description** property in the right pane and click in the cell that contains the **Description** value. Change the current value to “created in CIM Studio”
7. Find the Name property in the right pane and click in the cell that contains the Name value. Change the current value to “CIM Studio modified package”
8. To save the package properties changes made click on the Save icon.
9. Open the SMS Administrator console
10. Click on the Packages node. You should see the package with the PackageID that you noted in the previous procedure.
11. In the left tree view Right Click this package and select **Properties**. In the dialog box displayed you should see the updated package name in the **Name** box and the updated description in the **Comment** box.

**To view how objects are associated to each other**

1. Open WMI CIM Studio and connect to the `root\sms\site_mcm` namespace.
2. Find the **SMS_Advertisement** object in the left pane and click on it.
3. The Properties tab is displayed by default. Click on the **Associations** tab to graphically see how the SMS_Advertisement object is associated to other objects. Notice that the SMS_Advertisement object is associated to the SMS_Collection, SMS_Site, and SMS_Package objects.
4. Now click on the Instances icon to show all instances of the SMS_Advertisement object. All advertisement instances should now appear in the right pane.
5. Double click on one of the advertisements to view all properties and associated values for that specific advertisement.
6. Click on the **Associations** tab to graphically see how the specific advertisement is associated to a specific collection, site code, and package ID.

Look at other classes, how they are associated, and even connect to other namespaces used by SMS (root\ccm, root\sms, root\cimv2\sms, etc.) and go through some of the previous steps to get a better feel for how the WMI CIM Studio can really be another useful tool for the SMS Administrator.