

Shelling MOM 2005: Part 1 – Creating a MOM-Friendly Shell

By Pete Zerger
MCSE(Messaging), MVP - MOM
July 2006

With PowerShell on the horizon as the standard management interface for future Microsoft Server applications, I've been reading, looking around and asking questions and looking at production scenarios I can leverage in getting acquainted with PowerShell on current applications... namely MOM 2005. Here's a basic tutorial to get those new to PowerShell or the non-programmatically inclined started down the road in preparation for Ops Mgr 2007.

You've no doubt seen the post from The [PowerShelled Guru](#) on how to leverage the MOM 2005 .NET class libraries in PowerShell. That being said, understand you cannot perform all MOM management functions from the Shell, only what is available through the .NET class libraries and/or MOM WMI namespace. Given that all the SDK examples are in C#, PowerShell offers an option that the novice developer may find somewhat more user friendly.

In this post, I am going to work from that base and provide some basic guidance in how to:

1. How to configure PowerShell script security and
2. How to pre-load the MOM .dlls in your default Shell session and then finally, I'll
3. Provide a sample script, starting with this article, and then on a regular basis going forward.

NOTES:

This and all future samples can be downloaded from the script downloads section at momresources.org. Look for the new "PowerShell" section on the site later today.

This requires installing and running PowerShell on the MOM Management Server, as it is not removable from what I've seen in the MSDN and SDK documentation..

Steps for Creating the MOM-Friendly Shell

1. Configure PowerShell Script Security

By default, script security in PowerShell is set at „Restricted”, which means any script (any .ps1 file) cannot be executed from within PowerShell. This includes profile scripts (which I’ll get to in a moment), so for this example we’ll configure the shell to run scripts “Unrestricted”.

From a Shell prompt, type

Set ExecutionPolicy Unrestricted

NOTE: This is not recommended best practice, but less cumbersome for demonstration purposes here.

2. Configure Pre-loading of MOM .NET Class Libraries

PowerShell allows special scripts, called **Profiles**, to be run during Shell startup to allow for user customization of the shell environment. The files that can be used (and are read in this order):

1. Documents and Settings\All Users\Documents\PSConfiguration\Profile.ps1
2. Documents and Settings\All Users\Documents\PSConfiguration\Microsoft.PowerShell_Profile.ps1
3. \$HOME\My Documents\PSConfiguration\profile.ps1
4. \$HOME\My Documents\PSConfiguration\Microsoft.PowerShell_profile.ps1

We’re going to use #2, since this will allow us to customize the shell environment to make it MOM friendly for all MOM administrators that launch PowerShell on the Mgmt Server.

Perform the following actions on the MOM Management Server:

1. If it does not already exist, create a file called **Microsoft.PowerShell_Profile.ps1** in the **Documents and Settings\All Users\Documents\PS** directory
2. Cut-and-paste the following code into this file, which will automatically load the MOM .NET Class Libraries and instantiate a MOM administration object.

*****start copy below this line*****

```
[System.Reflection.Assembly]::LoadFile("$($env:ProgramFiles)\Microsoft Operations Manager 2005\SDK Bin\Microsoft.Mom.Sdk.dll") > $null
```

```
[System.Reflection.Assembly]::LoadFile("$($env:ProgramFiles)\Microsoft Operations Manager 2005\SDK Bin\mom.context.dll") > $null
```

```
$mom = [Microsoft.EnterpriseManagement.Mom.Administration]::GetAdministrationObject()
```

*****end copy above this line*****

NOTE: The > \$null basically prevents the loading of these additional components from being echoed to the console.

3. Save the file and launch the PowerShell console on your Management Server. To confirm the dlls loaded successfully, run the following command from a prompt.

Mom\$ | gm

Through reflection, this statement simply enumerates the properties and methods of this object, giving you some idea of what operations are possible. See the MOM SDK for more information on the MOM .NET Class Libraries.

The output should look something like this:

```
TypeName: Microsoft.EnterpriseManagement.Mom.Administration
Name          MemberType      Definition
-----
GetHashCode    Method          System.Int32 GetHashCode()
GetManagementGroupName Method          System.String GetManagementGroupName()
****truncated for brevity ****
```

3. Create and Run a Script

Below is a sample script I threw together. Copy this into a notepad file and save to a local directory on your management server. This script will return a list of management pack names, the management pack version and current enabled/disabled status.

```
*****start copy below this line*****
$rulegroups = $mom.getrulegroups()
foreach ($rulegroup in $rulegroups) {
    $version=$rulegroup.GetVersion()
    write-host "MP Name:" $rulegroup.Name " MPVersion:" $version " Enabled:" $rulegroup.Enabled
}
*****end copy above this line*****
```

Save the file and **mpsummary.sp1**.

To run the script, launch PowerShell on your Management Server and type the following

.\mpsummary.sp1

NOTE: .\ is required, even when you're in the local directory, due to a security feature within PowerShell, which as I understand it is to prevent someone from naming a script to match that of a native command.

The output should look something like this:

```
Mode                LastWriteTime         Length Name
----                -
-a---          7/18/2006  1:26 PM           216 rulegroups.ps1
```

```
PS C:\scripts\powershell> .\rulegroups.ps1
MP Name: Custom    MPVersion: 00.0.0000.0000  Enabled: True
MP Name: Dell OpenManage  MPVersion: 00.0.0000.0000  Enabled: True
MP Name: Microsoft Exchange Intelligent Message Filter  MPVersion: 00.0.0000.0000
MP Name: Microsoft Exchange Server  MPVersion: 06.5.7385.0000  Enabled: True
```

For More Info:

See more on PowerShell and MOM 2005 at the PowerShelled blog

<http://mow001.blogspot.com/2006/06/PowerShell-and-mom-2005.html>

<http://mow001.blogspot.com/2006/07/PowerShell-and-mom2005-part-2-updating.html>