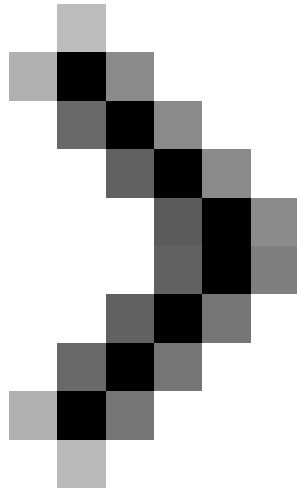


# Microsoft Windows Power Shell and SQL Server 2005 SMO - Part 4

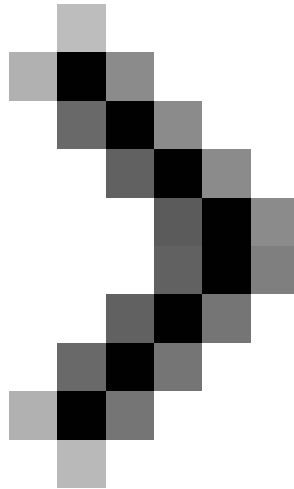
Part I and Part II of this series showed simple Power Shell settings, SMO and WMI cmdlets. Part III instructs writing PowerShell and connecting to SQL Server. Part 4 will show you how to use PowerShell code to iterate file content and connect to other servers.



**Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 1**



## **Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 2**



## Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 3

### *The MAK*

**Part I and Part II of this series showed simple Power Shell settings, SMO and WMI cmdlets. Part III instructs writing PowerShell and connecting to SQL Server. Part 4 will show you how to use PowerShell code to iterate file content and connect to other servers.**

Imagine that we want to connect to different SQL Servers and collect the correct information like server name, version and all database names.

### *Step 1*

We need to create a PowerShell script that displays all of the above information for a server. Create connect2.ps1 as shown below. [Figure 1.0]

```
param (  
[string] $ SQLSERVER  
)  
$ SqlConnection = New-Object System.Data.SqlClient.SqlConnection  
$ SqlConnection.ConnectionString =  
"Server = $ SQLSERVER; Database = master; Integrated Security = True"  
$ SqlCommand = New-Object System.Data.SqlClient.SqlCommand  
$ SqlCommand.CommandText = "select 'Servername:"
```

```
'+ @@servername as Result union Select' Version: '+
@@ version as Result union select 'Database:' + name from sysdatabases as Result
order by Result desc ''
$ SqlCmd.Connection = $ SqlConnection
$ SqlAdapter = New-Object System.Data.SqlClient.SqlDataAdapter
$ SqlAdapter.SelectCommand = $ SqlCmd
$ DataSet = New-Object System.Data.DataSet
$ SqlAdapter.Fill ($ DataSet)
$ SqlConnection.Close ()
$ DataSet.Tables [0]
```

```
connect2.ps1 - Notepad
File Edit Format View Help
param (
    [string] $SQLSERVER
)

$SqlConnection = New-Object System.Data.SqlClient.SqlConnection
$SqlConnection.ConnectionString = "Server=$SQLSERVER;Database=master;Integrated Security=True"

$SqlCommand = New-Object System.Data.SqlClient.SqlCommand
$SqlCommand.CommandText = 'select 'Servername: '+@@servername as Result union Select 'Version: '+
@@version as Result union select 'Database:' +name from sysdatabases as Result
order by Result desc ''
$SqlCommand.Connection = $SqlConnection

$SqlAdapter = New-Object System.Data.SqlClient.SqlDataAdapter
$SqlAdapter.SelectCommand = $SqlCommand

$DataSet = New-Object System.Data.DataSet
$SqlAdapter.Fill($DataSet)
$SqlConnection.Close()
$DataSet.Tables[0]
```

Figure 1.0

**Step 2**

Execute the following code: [Figure 1.1]

```
./connect2 "HOME"
```

Note : HOME is the server name. Please replace your server name.

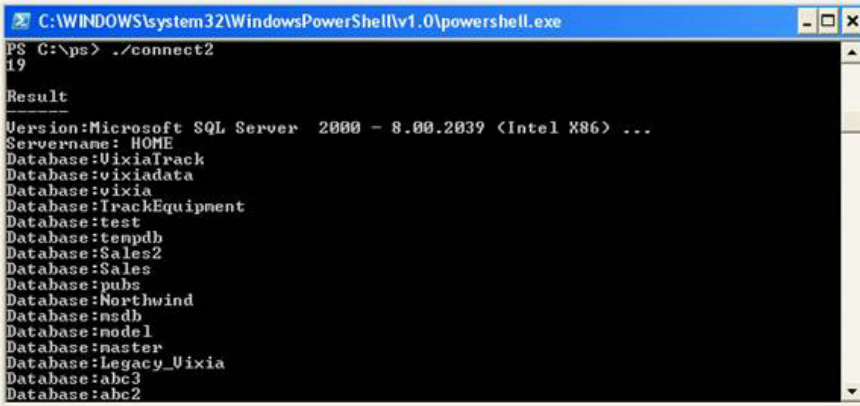
Result

**Result**

-----

```
Version: Microsoft SQL Server 2000 - 8.00.2039 (Intel X86) .
Servername: HOME
Database: VixiaTrack
```

Database: vixiadata  
Database: vixia  
Database: TrackEquipment  
Database: test  
Database: tempdb  
Database: Sales2  
Database: Sales  
Database: pubs  
Database: Northwind  
Database: msdb  
Database: model  
Database: master  
Database: Legacy\_Vixia  
Database: abc3  
Database: abc2  
Database: abc



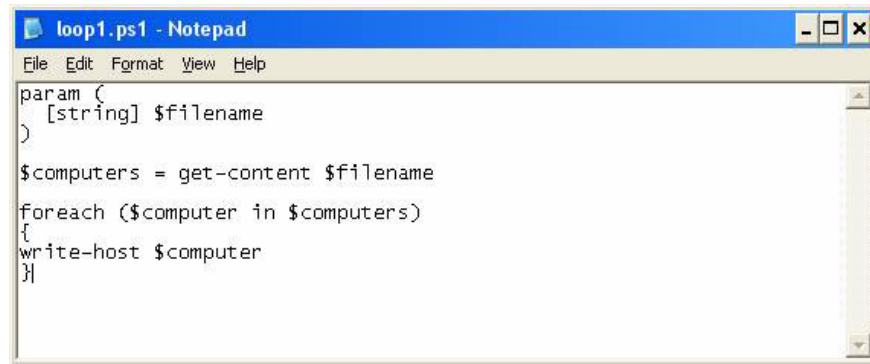
```
C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe
PS C:\ps> ./connect2
19
Result
Version:Microsoft SQL Server 2008 - 8.00.2039 (Intel X86) ...
Servername: HOME
Database:UixiaTrack
Database:vixiadata
Database:vixia
Database:TrackEquipment
Database:test
Database:tempdb
Database:Sales2
Database:Sales
Database:pubs
Database:Northwind
Database:msdb
Database:model
Database:master
Database:Legacy_Uixia
Database:abc3
Database:abc2
```

Figure 1.1

### Step 3

Repeat through a file and display the contents of the file. Create the following loop1.ps1 code: [Figure 1.2] Also create the serverlist.txt file. [Figure 1.3]

```
param ([string] $ filename) $ computers = get-content $ filename foreach ($
```

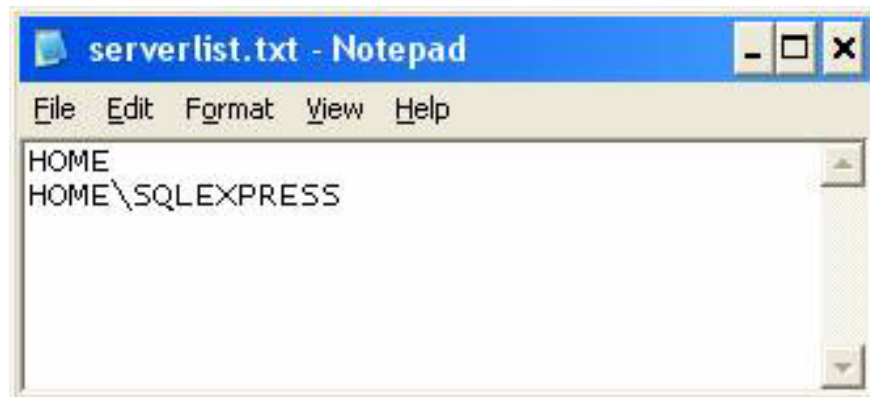


```
loop1.ps1 - Notepad
File Edit Format View Help
param (
  [string] $filename
)
$computers = get-content $filename
foreach ($computer in $computers)
{
write-host $computer
}
```

Figure 1.2

**Serverlist.txt**

HOME  
HOMESQLEXPRESS



```
serverlist.txt - Notepad
File Edit Format View Help
HOME
HOME\SQLEXPRESS
```

Figure 1.3

Please execute the loop1.ps1 code. [Figure 1.4]

**`./loop1 serverlist.txt`**

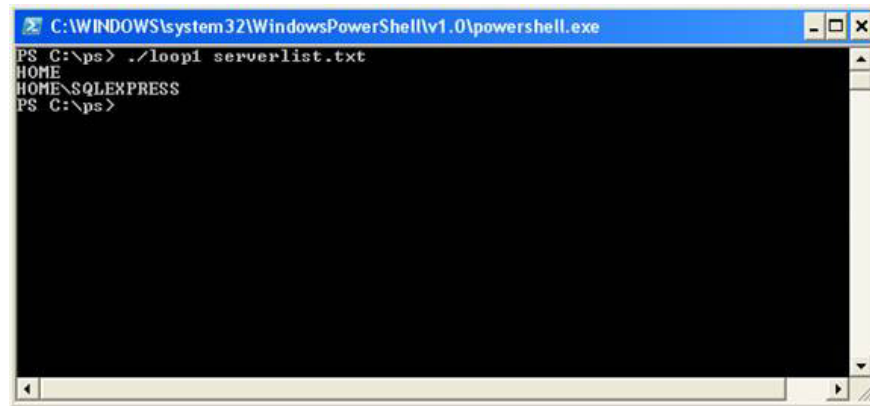


Figure 1.4

#### *Step 4*

Now combine connect2.ps1 and loop1.ps1 to obtain the names of SQL Server, SQL Server version and all database names listed on serverlist.txt.

Create connect3.ps1 as follows: [Figure 1.5]

```
param ([string] $ filename) $ SqlConnection = New-Object System.Data.SqlClient

$SqlConnection.ConnectionString = "Server=$computer;Database=master;IntegratedSecurity=true"

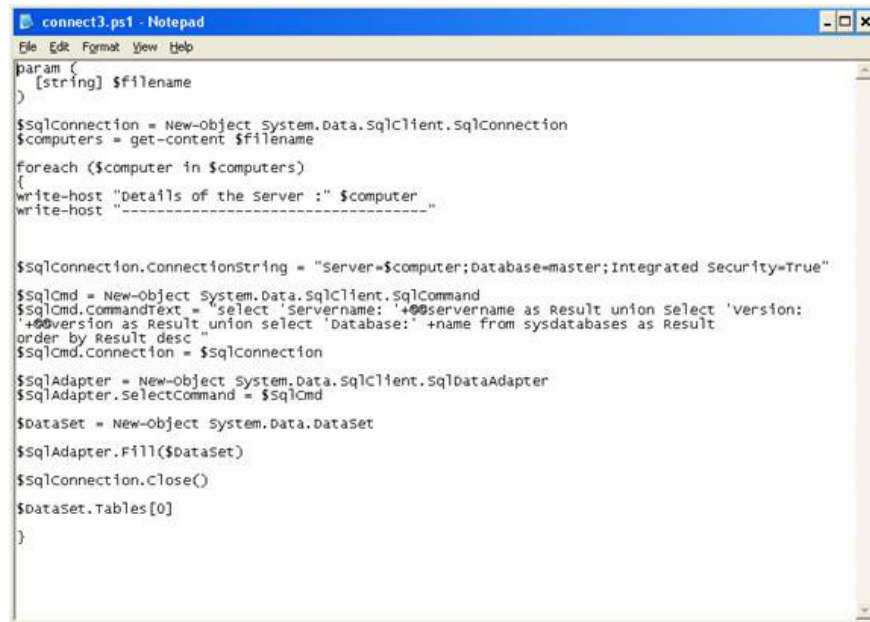
$ SqlConnection.ConnectionString = "Server = $ computer; Database = master;

+@@version as Result union select 'Database:' +name from sysdatabases as Res

+ @@ version as Result union select 'Database:' + name from sysdatabases as

??

SqlAdapter.Fill ($ DataSet) $ SqlConnection.Close () $ DataSet.Tables [0]}
```



```
connect3.ps1 - Notepad
File Edit Format View Help
param (
    [string] $filename
)

$sqlconnection = New-Object System.Data.SqlClient.SqlConnection
$computers = get-content $filename

foreach ($computer in $computers)
{
    write-host "details of the Server :" $computer
    write-host "-----"

    $sqlconnection.ConnectionString = "Server=$computer;database=master;integrated security=true"

    $sqlcmd = New-Object System.Data.SqlClient.SqlCommand
    $sqlcmd.CommandText = 'select 'servername: '+@@servername as Result union select 'version:
    '+@@version as Result union select 'database:' +name from sysdatabases as Result
    order by Result desc '
    $sqlcmd.Connection = $sqlconnection

    $sqladapter = New-Object System.Data.SqlClient.SqlDataAdapter
    $sqladapter.SelectCommand = $sqlcmd

    $dataset = New-Object System.Data.DataSet
    $sqladapter.Fill($dataset)

    $sqlconnection.Close()

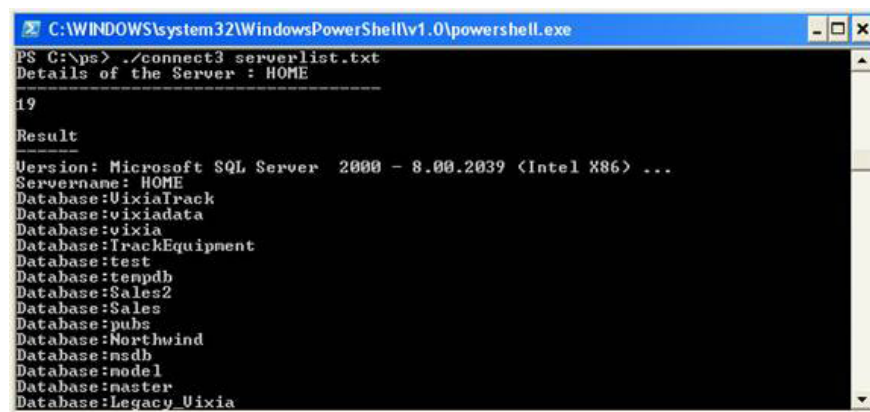
    $dataset.Tables[0]
}
}
```

Figure 1.5

**Step 5**

Please execute the following code: [Figure 1.6]

**./connect3 serverlist.txt**



```
C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe
PS C:\> ./connect3 serverlist.txt
Details of the Server : HOME
-----
19
Result
Version: Microsoft SQL Server 2000 - 8.00.2039 (Intel X86) ...
Servername: HOME
Database:UixiaTrack
Database:vixiadata
Database:vixia
Database:TrackEquipment
Database:test
Database:tempdb
Database:Sales2
Database:Sales
Database:pubs
Database:Northwind
Database:msdb
Database:model
Database:master
Database:Legacy_Uixia
```

Figure 1.6

Result:

**Details of the Server: HOME**

-----

**19**

## Result

-----

**Version: Microsoft SQL Server 2000 - 8.00.2039 (Intel X86) .**

**Servename: HOME**

**Database: VixiaTrack**

**Database: vixiadata**

**Database: vixia**

**Database: TrackEquipment**

**Database: test**

**Database: tempdb**

**Database: Sales2**

**Database: Sales**

**Database: pubs**

**Database: Northwind**

**Database: msdb**

**Database: model**

**Database: master**

**Database: Legacy\_Vixia**

**Database: abc3**

**Database: abc2**

**Database: abc**

**Details of the Server: HOMESQLEXPRESS**

-----

**11**

**Version: Microsoft SQL Server 2005 - 9.00.2047.00 (Intel X86) .**

**Servename: HOMESQLEXPRESS**

**Database: XMLTest**

**Database: VixiaTrack**

**Database: test**

**Database: tempdb**

**Database: msdb**

**Database: model**

**Database: master**

**Database: AdventureWorks**

**Database: admin**

Now create a similar PowerShell code, using SQL Server SMO.

### *Step 1*

Generate code connect4.ps1 PowerShell [Figure 1.7]

```
param ([string] $ filename) [reflection.assembly] :: LoadWithPartialName ("M
```

```

connect4.ps1 - Notepad
File Edit Format View Help
param
(
    [string] $filename
)
[reflection.assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo")
$SqlConnection = New-Object System.Data.SqlClient.SqlConnection
$computers = get-content $filename
foreach ($computer in $computers)
{
    write-host "details of the Server : " $computer
    write-host "-----"
    $Server = new-object ("Microsoft.SqlServer.Management.Smo.Server") "$computer"
    write-host "Server Version: " $Server.ServerVersion
    write-host "Server Name: " $Server.Information.VersionString
    foreach($database in $Server.databases)
    {
        write-host "Database:" $database.name
    }
}

```

Figure 1.7

**Step 2**

Execute the code connect4.ps1 created above [Figure 1.8]

**./connect4 serverlist.txt**

```

C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe
PS C:\ps> ./connect4 serverlist.txt
GAC    Version    Location
True   v2.0.50727  C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.Smo\9
-----
Details of the Server : HOME
Server Version:
Server Name: 8.00.2039
Database: abc
Database: abc2
Database: abc3
Database: Legacy_Uixia
Database: master

```

Figure 1.8

**Result**

```

PS C: ps> ./connect4 serverlist.txt
GAC Version Location
-----
True v2.0.50727 C: WINDOWSassemblyGAC_MSILMicrosoft.SqlServer.Smo9
Details of the Server: HOME
-----

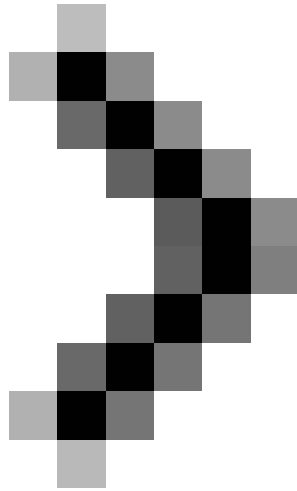
```

**Server Version:**  
**Server Name: 8.00.2039**  
**Database: abc**  
**Database: abc2**  
**Database: abc3**  
**Database: Legacy\_Vixia**  
**Database: master**  
**Database: model**  
**Database: msdb**  
**Database: Northwind**  
**Database: pubs**  
**Database: Sales**  
**Database: Sales2**  
**Database: tempdb**  
**Database: test**  
**Database: TrackEquipment**  
**Database: vixia**  
**Database: vixiadata**  
**Database: VixiaTrack**  
**Details of the Server: HOMESQLEXPRESS**

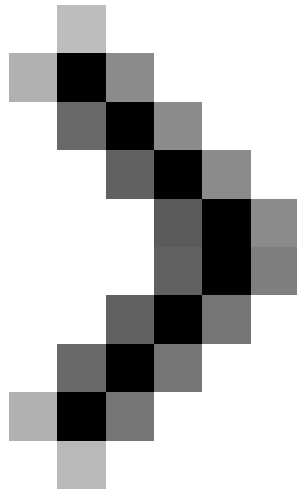
-----  
**Server Version:**  
**Server Name: 9.00.2047.00**  
**Database: admin**  
**Database: AdventureWorks**  
**Database: master**  
**Database: model**  
**Database: msdb**  
**Database: tempdb**  
**Database: test**  
**Database: VixiaTrack**  
**Database: XMLTest**

## **Conclude**

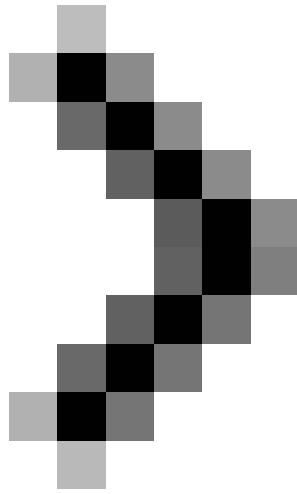
Part IV introduces how to use the PowerShell script to loop through the contents of a file and connect to different servers. This section also describes how to do the same using SQL Server SMO.



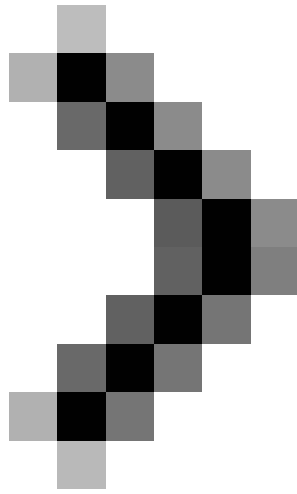
**Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 5**



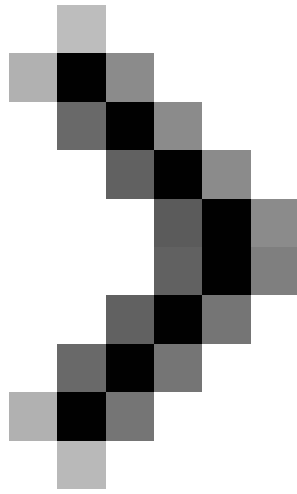
## **Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 6**



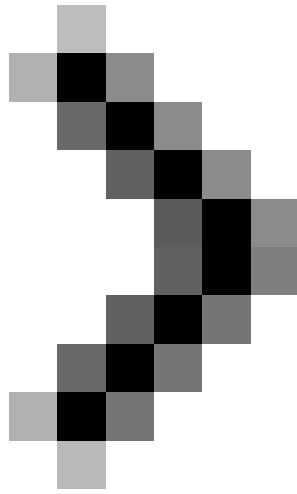
**Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 7**



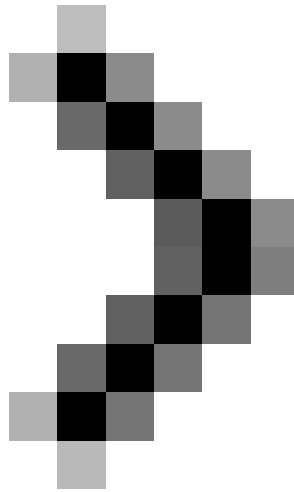
**Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 8**



**Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 9**



## **Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 10**



## **Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 11**

You finished reading the article "**Microsoft Windows Power Shell and SQL Server 2005 SMO - Part 4**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

---