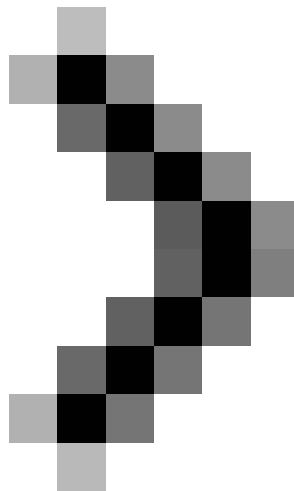
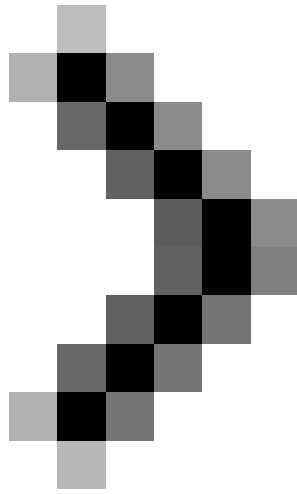


# Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 6

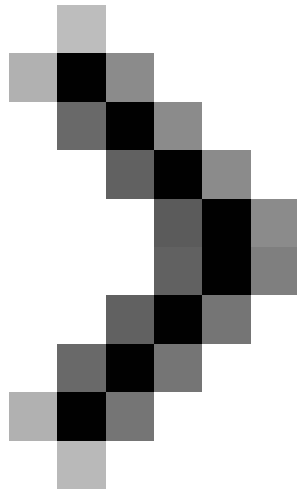
In this article, I will show you how to use PowerShell and scripts to perform a complete backup - Full Backup, Transaction Log backup and Differential backup for the database with the file name already pasted. external stamp.



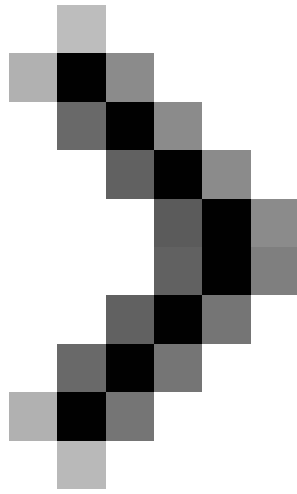
**Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 1**



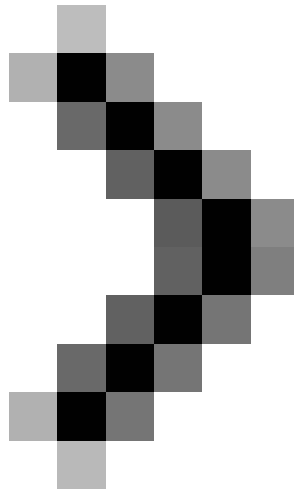
## **Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 2**



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



## **Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 4**



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

### *The MAK*

**In Part 6, I will show you how to use PowerShell and scripts to back up the database.**

### **Method 1**

Let's assume we have a database 'MyDatabase' on the 'HOME' server. Now let's assume that we want to make a full backup of the database to the **c: test** directory.

Execute each cmdlet one by one. Refer to Figure 1.0

```
[System.Reflection.Assembly] :: LoadWithPartialName ("Microsoft.SqlServer.Smo") | out-null
[System.IO.Directory] :: CreateDirectory ("C: test") | out-null
$ srv = New-Object "Microsoft.SqlServer.Management.Smo.Server" "HOME"
$ bck = new-object "Microsoft.SqlServer.Management.Smo.Backup"
$ bck.Action = 'Database'
$ fil = new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
$ fil.DeviceType = 'File'
$ fil.Name = [System.IO.Path] :: Combine ("C: test", "MyDatabase" + ". bak")
$ bck.Devices.Add ($ fil)
$ bck.Database = "MyDatabase"
```

```
$ bck.SqlBackup ($ srv)
write-host "Backup of MyDatabase done"
```

```
C:\WINDOWS\system32\cmd.exe - powershell
PS C:\> [System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo") | out-null
PS C:\> [System.IO.Directory]::CreateDirectory("C:\test") | out-null
PS C:\> $srv=New-Object "Microsoft.SqlServer.Management.Smo.Server" "HOME"
PS C:\> $bck=new-object "Microsoft.SqlServer.Management.Smo.Backup"
PS C:\> $bck.Action = 'Database'
PS C:\> $fil=new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
PS C:\> $fil.DeviceType='File'
PS C:\> $fil.Name=[System.IO.Path]::Combine("C:\test", "MyDatabase" + ".bak")
PS C:\> $bck.Devices.Add($fil)
PS C:\> $bck.Database="MyDatabase"
PS C:\> $bck.SqlBackup($srv)
PS C:\> write-host "Backup of MyDatabase done"
Backup of MyDatabase done
PS C:\>
```

Figure 1.0

This script will create a full backup of the MyDatabase database to the c: test directory as shown below. (Refer to Figure 1.1)

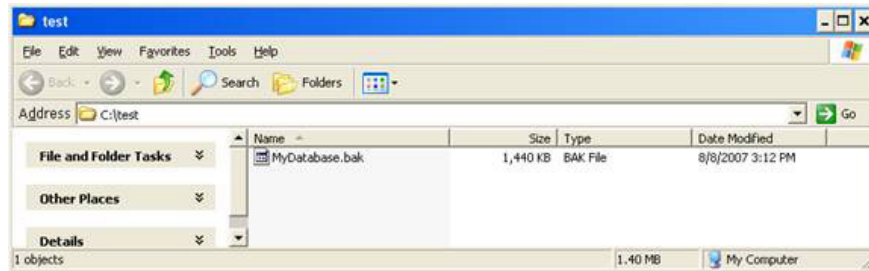


Figure 1.1

## Method 2

Let's assume we have a database 'MyDatabase' on the 'HOME' server and want to perform a Transaction log backup of the databases into the c: test directory. Execute the cmdlet below, times the amount of each instruction (refer to Figure 1.2).

```
[System.Reflection.Assembly] :: LoadWithPartialName ("Microsoft.SqlServer.Smo") | out-null
[System.IO.Directory] :: CreateDirectory ("C: test") | out-null
$ srv = New-Object "Microsoft.SqlServer.Management.Smo.Server" "HOME"
$ bck = new-object "Microsoft.SqlServer.Management.Smo.Backup"
$ bck.Action = 'Log'
$ fil = new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
$ fil.DeviceType = 'File'
$ fil.Name = [System.IO.Path] :: Combine ("C: test", "MyDatabase" + ". trn")
$ bck.Devices.Add ($ fil)
$ bck.Database = "MyDatabase"
$ bck.SqlBackup ($ srv)
write-host "Log Backup of MyDatabase done"
```

```

C:\WINDOWS\system32\cmd.exe - powershell
PS C:\> [System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo") | out-null
PS C:\> [System.IO.Directory]::CreateDirectory("C:\test") | out-null
PS C:\> $srv=New-Object "Microsoft.SqlServer.Management.Smo.Server" "HOME"
PS C:\> $bck=new-object "Microsoft.SqlServer.Management.Smo.Backup"
PS C:\> $bck.Action = 'Log'
PS C:\> $fil=new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
PS C:\> $fil.DeviceType='File'
PS C:\> $fil.Name=[System.IO.Path]::Combine("C:\test", "MyDatabase" + ".trn")
PS C:\> $bck.Devices.Add($fil)
PS C:\> $bck.Database="MyDatabase"
PS C:\> $bck.SqlBackup($srv)
PS C:\> write-host "Log Backup of MyDatabase done"
Log Backup of MyDatabase done
PS C:\>

```

Figure 1.2

This scenario will perform a Transaction log backup of the MyDatabase database to the c: test directory, as shown in the figure below (refer to Figure 1.3).

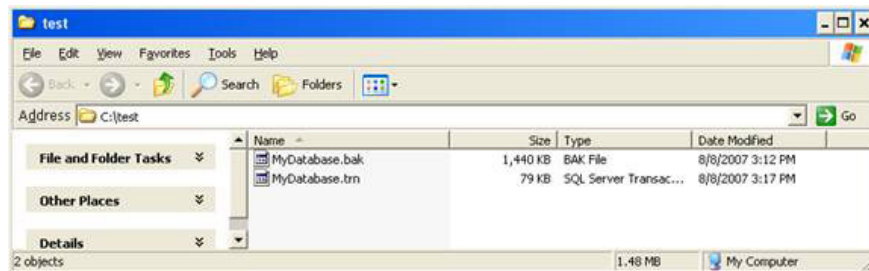


Figure 1.3

### Method 3

Let's assume we have a database 'MyDatabase' on the 'HOME' server and want to make a Differential backup for the databases into the c: test directory. Follow the commands below, one by one (refer to Figure 1.4).

```

[System.Reflection.Assembly] :: LoadWithPartialName ("Microsoft.SqlServer.Smo") | out-null
[System.IO.Directory] :: CreateDirectory ("C: test") | out-null
$ srv = New-Object "Microsoft.SqlServer.Management.Smo.Server" "HOME"
$ bck = new-object "Microsoft.SqlServer.Management.Smo.Backup"
$ bck.Incremental = 1
$ fil = new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
$ fil.DeviceType = 'File'
$ fil.Name = [System.IO.Path] :: Combine ("C: test", "MyDatabase" + ". diff")
$ bck.Devices.Add ($ fil)
$ bck.Database = "MyDatabase"
$ bck.SqlBackup ($ srv)
write-host "Differential Backup of MyDatabase done"

```

```

C:\WINDOWS\system32\cmd.exe - powershell
PS C:\> [System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo") | out-null
PS C:\> [System.IO.Directory]::CreateDirectory("C:\test") | out-null
PS C:\> $srv=New-Object "Microsoft.SqlServer.Management.Smo.Server" "HOME"
PS C:\> $bck=new-object "Microsoft.SqlServer.Management.Smo.Backup"
PS C:\> $bck.Incremental = 1
PS C:\> $fill=new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
PS C:\> $fill.DeviceType="File"
PS C:\> $fill.Name=[System.IO.Path]::Combine("C:\test", "MyDatabase" + ".diff")
PS C:\> $bck.Devices.Add($fill)
PS C:\> $bck.Database="MyDatabase"
PS C:\> $bck.SqlBackup($srv)
PS C:\> write-host "Diferential Backup of MyDatabase done"
Diferential Backup of MyDatabase done
PS C:\>

```

Figure 1.4

This script will perform backup Differential of MyDatabase database to c: test directory as shown below (refer to Figure 1.5)

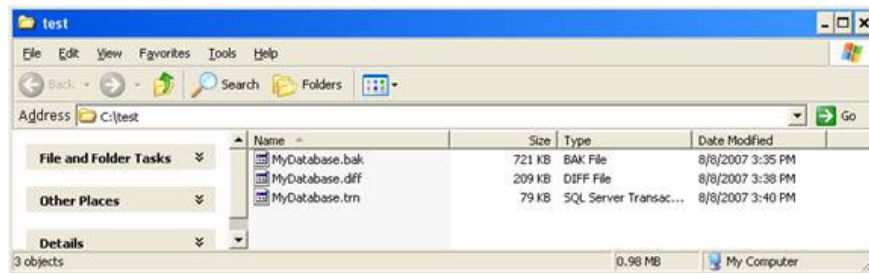


Figure 1.5

#### Method 4

Now let's connect methods 1, method 2 and method 3 to a PowerShell script that takes the Server name, Database name, Backup type and Folder folder as parameters so that it can be easily implemented. automatically. Create the Backup.ps1 script below in the C: PS folder (refer to Figure 1.6, 1.7).

```

param (
    [string] $ ServerName,
    [string] $ DatabaseName,
    [string] $ Backuptype,
    [string] $ BackupPath
)
[System.Reflection.Assembly] :: LoadWithPartialName ("Microsoft.SqlServer.Smo") | out-null
[System.IO.Directory] :: CreateDirectory ($ BackupPath) | out-null
$ srv = New-Object "Microsoft.SqlServer.Management.Smo.Server" "$ servername"
$ bck = new-object "Microsoft.SqlServer.Management.Smo.Backup"

if ($ Backuptype -eq "FULL")
{
    $ bck.Action = 'Database'
    $ extension = ". BAK"
    $ text1 = "Full Backup"
}

```

```

if ($ Backuptype -eq "TRAN")
{
$ bck.Action = 'Log'
$ extension = ". LOG"
$ text1 = "Transactional Log Backup"
}

```

```

if ($ Backuptype -eq "DIFF")
{
$ bck.Incremental = 1
$ extension = ". DIFF"
$ text1 = "Differential Backup"
}

```

```

$ fil = new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
$ fil.DeviceType = 'File'
$ fil.Name = [System.IO.Path] :: Combine ($ BackupPath, $ DatabaseName + $ extension)
$ bck.Devices.Add ($ fil)
$ bck.Database = $ DatabaseName
$ bck.SqlBackup ($ srv)
write-host $ text1 of $ Databasename done

```

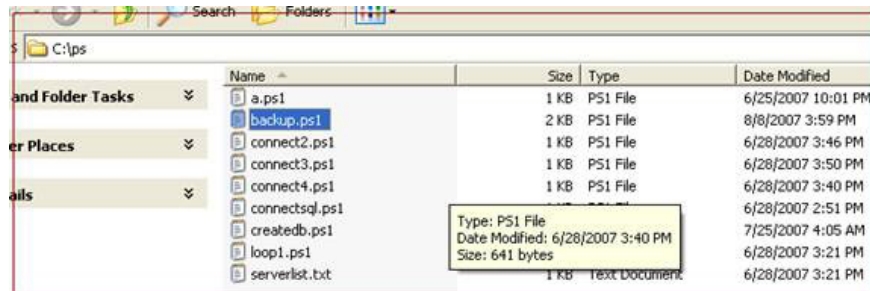


Figure 1.6

```

backup.ps1 - Notepad
File Edit Format View Help
param (
    [string] $ServerName,
    [string] $DatabaseName,
    [string] $BackupType,
    [string] $BackupPath
)
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo") |
out-null
[System.IO.Directory]::CreateDirectory($BackupPath) | out-null
$srv=new-object "Microsoft.SqlServer.Management.Smo.Server" "$serverName"
$bck=new-object "Microsoft.SqlServer.Management.Smo.Backup"

if ($BackupType -eq "FULL")
{
    $bck.Action = 'Database'
    $extension=".BAK"
    $text1="Full Backup "
}

if ($BackupType -eq "TRAN")
{
    $bck.Action = 'Log'
    $extension=".LOG"
    $text1="Transactional Log Backup "
}

if ($BackupType -eq "DIFF")
{
    $bck.Incremental = 1
    $extension=".DIFF"
    $text1="Differential Backup "
}

$fil=new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"

```

Figure 1.7

Now let's execute the above PowerShell script with the following parameters (refer to Figure 1.8).

- ./backup.ps1 HOME MyDatabase FULL C: test**
- ./backup.ps1 HOME MyDatabase DIFF C: test**
- ./backup.ps1 HOME MyDatabase TRAN C: test**

```

C:\WINDOWS\system32\cmd.exe - powershell
PS C:\> ./.backup.ps1 HOME MyDatabase DIFF C:\test
Differential Backup of MyDatabase done
PS C:\> ./.backup.ps1 HOME MyDatabase TRAN C:\test
Transactional Log Backup of MyDatabase done
PS C:\> ./.backup.ps1 HOME MyDatabase FULL C:\test
Full Backup of MyDatabase done
PS C:\>

```

Figure 1.8

Executing the script above gives the results below. (Refer to Figures 1.8 and 1.9)

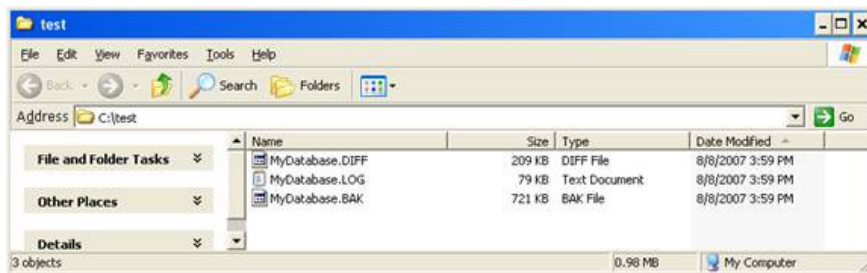


Figure 1.9

## Method 5

Let's assume that we need to add the Stamp with the file name when performing backup with the above script. To do that, update the script with the code below (refer to Figure 2.0)

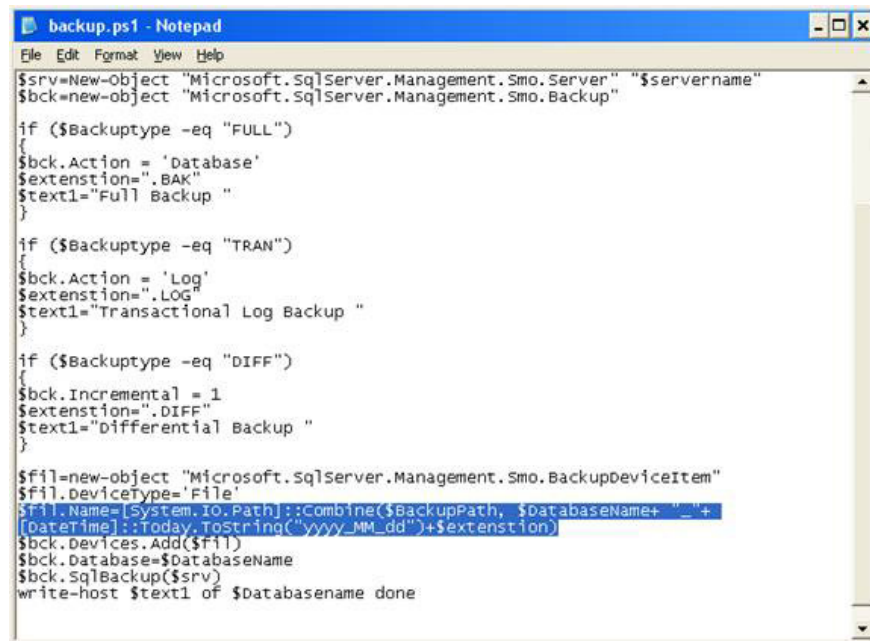
```
param (
    [string] $ ServerName,
    [string] $ DatabaseName,
    [string] $ Backuptype,
    [string] $ BackupPath
)
[System.Reflection.Assembly] :: LoadWithPartialName ("Microsoft.SqlServer.Smo") | out-null
[System.IO.Directory] :: CreateDirectory ($ BackupPath) | out-null
$ srv = New-Object "Microsoft.SqlServer.Management.Smo.Server" "$ servername"
$ bck = new-object "Microsoft.SqlServer.Management.Smo.Backup"

if ($ Backuptype -eq "FULL")
{
    $ bck.Action = 'Database'
    $ extension = ". BAK"
    $ text1 = "Full Backup"
}

if ($ Backuptype -eq "TRAN")
{
    $ bck.Action = 'Log'
    $ extension = ". LOG"
    $ text1 = "Transactional Log Backup"
}

if ($ Backuptype -eq "DIFF")
{
    $ bck.Incremental = 1
    $ extension = ". DIFF"
    $ text1 = "Differential Backup"
}

$ fil = new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
$ fil.DeviceType = 'File'
$ fil.Name = [System.IO.Path] :: Combine ($ BackupPath, $ DatabaseName + "_" + [DateTime] ::
Today.ToString ('yyyy_MM_dd')) + $ extension)
$ bck.Devices.Add ($ fil)
$ bck.Database = $ DatabaseName
$ bck.SqlBackup ($ srv)
write-host $ text1 of $ Databasename done
```



```
backup.ps1 - Notepad
File Edit Format View Help
$srv=new-object "Microsoft.SqlServer.Management.Smo.Server" "$servername"
$bck=new-object "Microsoft.SqlServer.Management.Smo.Backup"

if ($Backuptype -eq "FULL")
{
$bck.Action = 'Database'
$extension=".BAK"
$text1="Full Backup "
}

if ($Backuptype -eq "TRAN")
{
$bck.Action = 'Log'
$extension=".LOG"
$text1="Transactional Log Backup "
}

if ($Backuptype -eq "DIFF")
{
$bck.Incremental = 1
$extension=".DIFF"
$text1="Differential Backup "
}

$file=new-object "Microsoft.SqlServer.Management.Smo.BackupDeviceItem"
$file.DeviceType="File"
$file.Name=[System.IO.Path]::Combine($Backuppath, $DatabaseName+ "-" +
[DateTime]::Today.ToString("yyyy-MM-dd")+ $extension)
$bck.Devices.Add($file)
$bck.Database=$DatabaseName
$bck.SqlBackup($srv)
write-host $text1 of $DatabaseName done
```

Figure 2.0

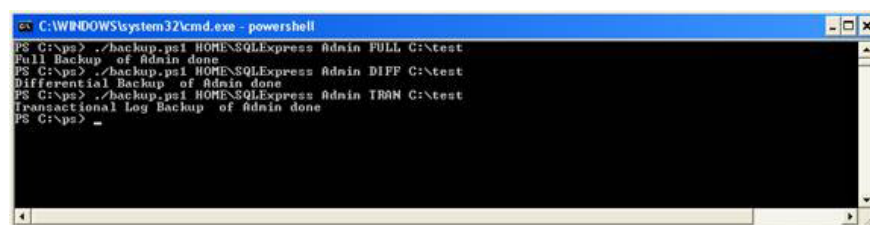
Now let's execute the above PowerShell script with the following parameters (refer to Figure 2.1).

- ./backup.ps1 HOMESQLExpress Admin FULL C: test**
- ./backup.ps1 HOMESQLExpress Admin DIFF C: test**
- ./backup.ps1 HOMESQLExpress Admin TRAN C: test**

**Note** : HOMESQLExpress is a named example of sql server on the HOME host;

Admin is the database name.

Executing the above scenario results in Figure 2.1 and 2.2.



```
C:\WINDOWS\system32\cmd.exe - powershell
PS C:\> ./backup.ps1 HOMESQLExpress Admin FULL C:\test
Full Backup of Admin done
PS C:\> ./backup.ps1 HOMESQLExpress Admin DIFF C:\test
Differential Backup of Admin done
PS C:\> ./backup.ps1 HOMESQLExpress Admin TRAN C:\test
Transactional Log Backup of Admin done
PS C:\> _
```

Figure 2.1

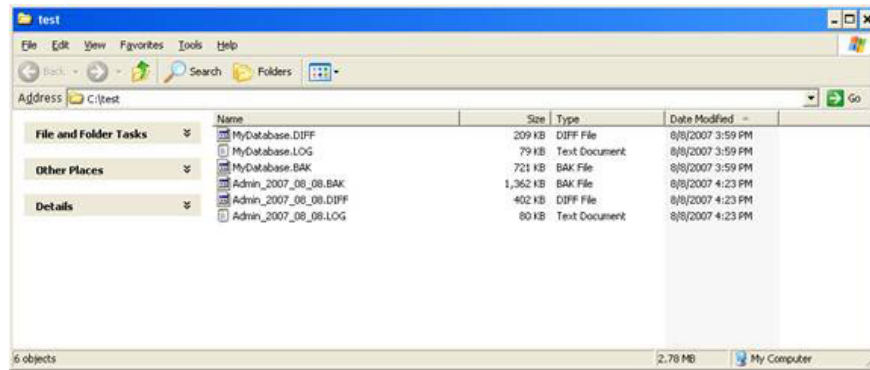
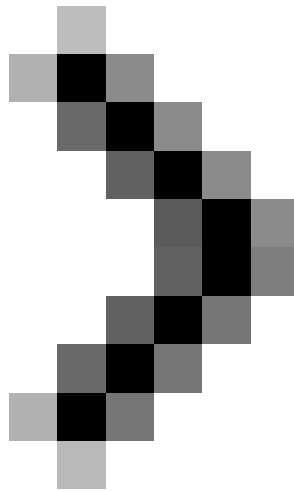
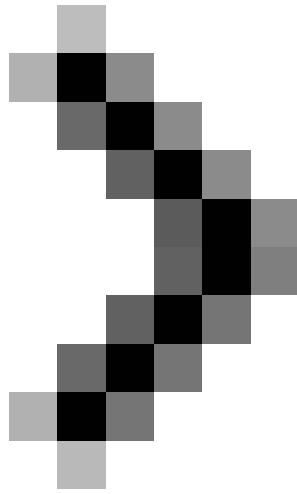


Figure 2.2

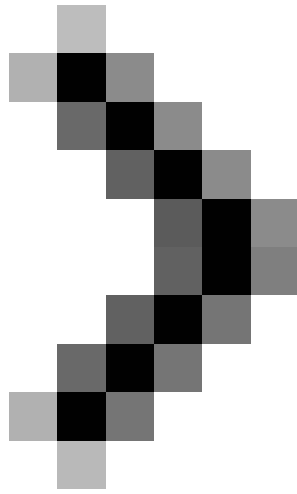
## Conclude

Part 6 of this series illustrated how to use PowerShell and PowerShell scripts to perform a complete backup - Full Backup, Transaction Log backup and Differential backup for the database with the file name already stamping outside.

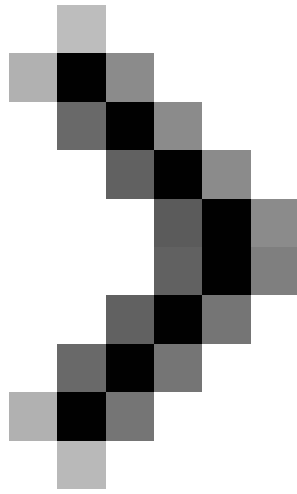




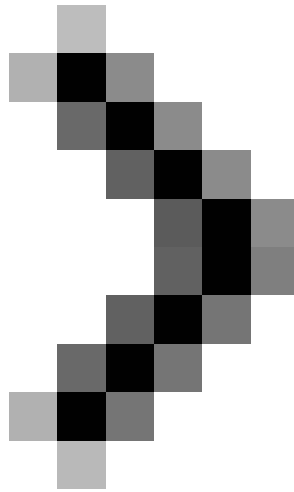
**Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 8**



**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 PowerShell and SQL Server 2005 SMO - Part 6**" 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.

---