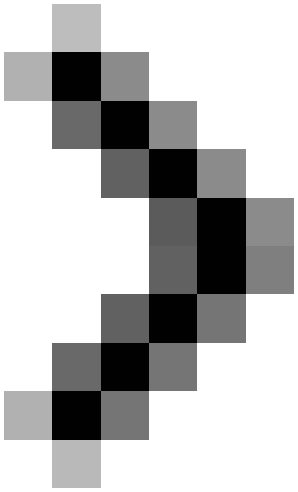
Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 2

Part I of this series shows how to set up and use simple PowerShell and SMO commands. In Part II we will learn more about PowerShell as well as its features associated with SMO. If you do & ati



Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 1

The MAK

Part I of this series shows how to set up and use simple PowerShell and SMO commands. In Part II we will learn more about PowerShell as well as its features associated with SMO. If you've ever known programming languages ??like PERL, Python or C, you can find similarities with the syntax used in PowerShell. In addition, it is compatible with operating systems like UNIX, Linux, MS-DOS .

Although PowerShell has great power, not all tasks can be handled with PowerShell. However, Power Shell's

shortcomings can be filled using .NET classes and Window managers, such as WMI. Automation techniques of the Power Shell cmdlets can be completed by creating script code.

Restore system date and time using Power Shell

A simple date and time value of the system can be restored using a simple 'date' cmdlet. [Figure 1.0]

Cmdlet:
date
Result:
Tuesday, June 12, 2007 8:52:27 AM

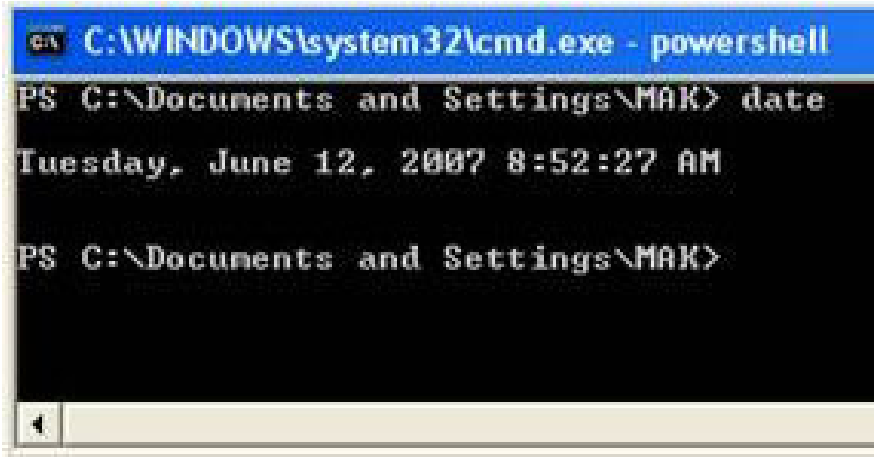


Figure 1.0

The system's Date and Time value can be recovered by using a .NET class, following the following cmdlets. [Figure 1.1]

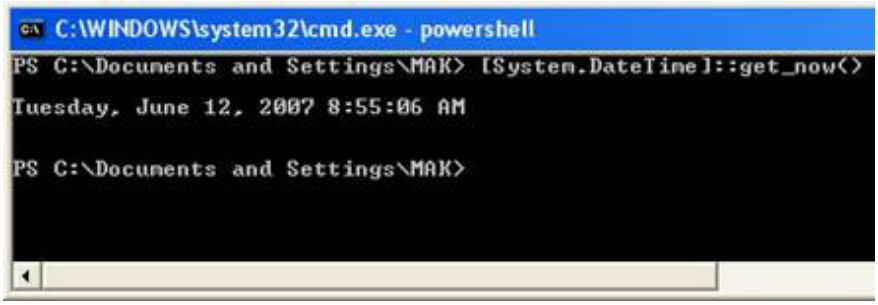


Figure 1.1

We can even give detailed information like Year, Month and Day. [Figure 1.2]

Cmdlet:
[System.DateTime] :: get_now (). Year
[System.DateTime] :: get_now (). Month
[System.DateTime] :: get_now (). Day
Result:
2007
6
twelfth

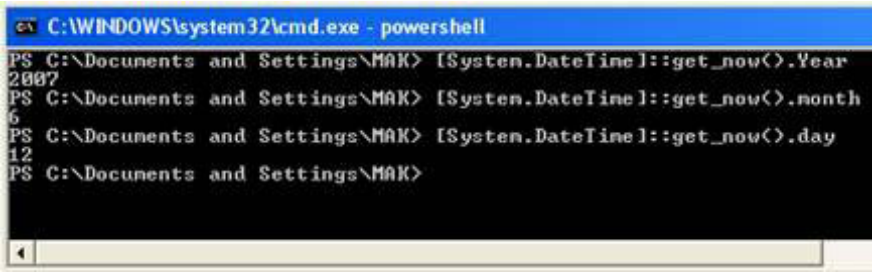


Figure 1.2

Use the WMI cmdlet to recover Date and Time information. [Figure 1.3]

Cmdlet:
get-wmiobject -Namespace rootcimv2 -Class Win32_CurrentTime
Result:
__GENUS: 2
__CLASS: Win32_LocalTime
__SUPERCLASS: Win32_CurrentTime
__DYNASTY: Win32_CurrentTime
__RELPATH: Win32_LocalTime = @
__PROPERTY_COUNT: 10
__DERIVATION: {Win32_CurrentTime}
__SERVER: HOME
__NAMESPACE: rootcimv2
__PATH: HOMErootcimv2: Win32_LocalTime = @
Day: 12
DayOfWeek: 2
Hour: 9
Milliseconds:
Minute: 3
Month: 6
Quarter: 2
Second: 0
WeekInMonth: 3
Year: 2007
__GENUS: 2

__CLASS: Win32_UTCTime
__SUPERCLASS: Win32_CurrentTime
__DYNASTY: Win32_CurrentTime
__RELPATH: Win32_UTCTime = @
__PROPERTY_COUNT: 10
__DERIVATION: {Win32_CurrentTime}
__SERVER: HOME
__NAMESPACE: rootcimv2
__PATH: HOMErootcimv2: Win32_UTCTime = @
Day: 12
DayOfWeek: 2
Hour: 13
Milliseconds:
Minute: 3
Month: 6
Quarter: 2
Second: 0
WeekInMonth: 3
Year: 2007

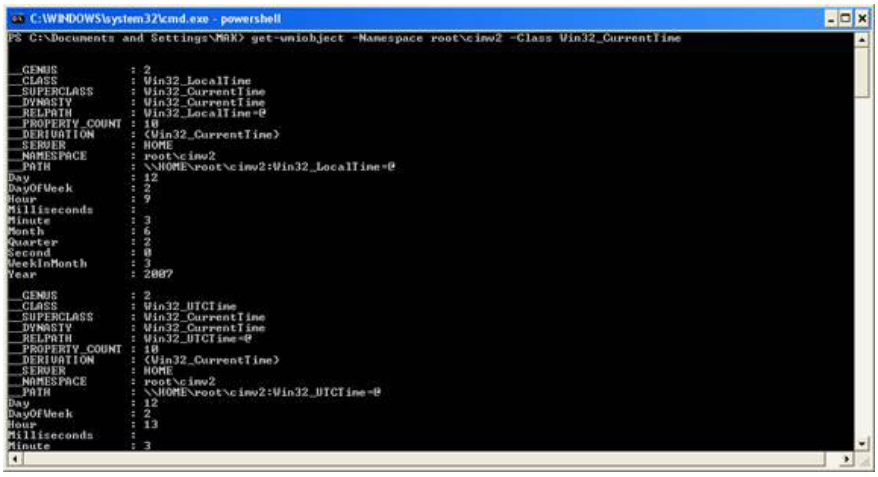


Figure 1.3

Power Shell also allows you to run or execute ad-hock SQL queries for SQL 2000 or SQL 2005 databases.

We can get the Date and Time value from SQL Server using SQLServer Management Object and ***getdate ()*** function. In this example, we make a simple connection to a SQL Server and execute the simple ***getdate () function*** to recover the Date and Time value according to the commands below. [Figure 1.4]

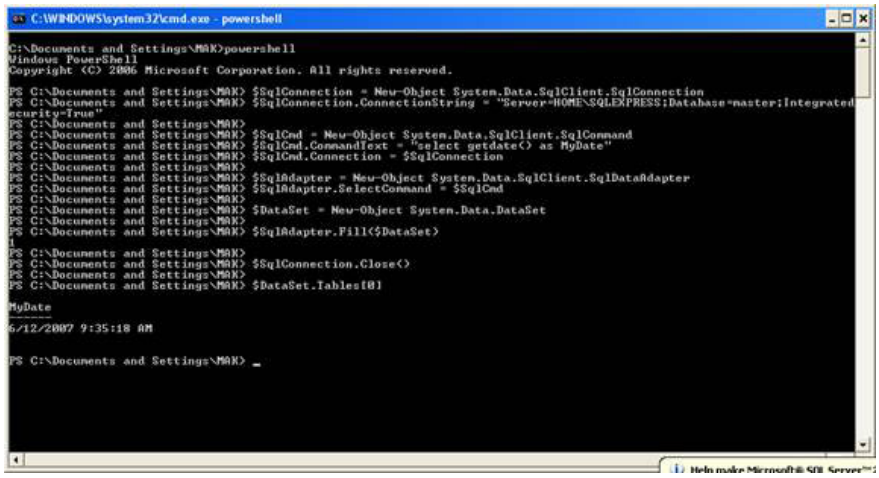


Figure 1.4

Cmdlets
$ SqlConnection = New-Object System.Data.SqlClient.SqlConnection
$ SqlConnection.ConnectionString = "Server = HOMESQLEXPRESS; Database = master; Integrated Security = True"
$ SqlCmd = New-Object System.Data.SqlClient.SqlCommand
$ SqlCmd.CommandText = "select getdate () as MyDate"
$ 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]
Result
MyDate

December 6, 2007 9:35:18 AM

This same example can be used for any adhoc query. Please execute the stored procedure 'sp_helpdb' shown below.

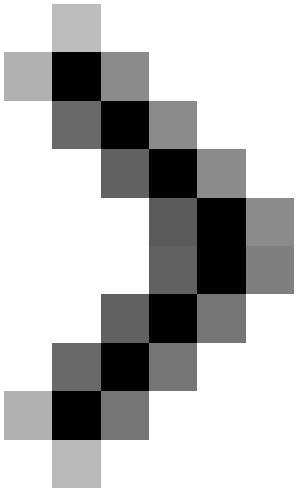
Cmdlets
$ SqlConnection = New-Object System.Data.SqlClient.SqlConnection
$ SqlConnection.ConnectionString = "Server = HOMESQLEXPRESS; Database = master; Integrated Security = True"
$ SqlCmd = New-Object System.Data.SqlClient.SqlCommand
$ SqlCmd.CommandText = "sp_helpdb"
$ SqlCmd.Connection = $ SqlConnection
$ SqlAdapter = New-Object System.Data.SqlClient.SqlDataAdapter
$ SqlAdapter.SelectCommand = $ SqlCmd
$ DataSet = New-Object System.Data.DataSet

$ SqlAdapter.Fill ($ DataSet)
$ SqlConnection.Close ()
$ DataSet.Tables [0]
Result
name: master
db_size: 4.75 MB
owner: sa
dbid: 1
created: Apr 8 2003
status: Status = ONLINE, Updateability = READ_WRITE, UserAccess = MULTI_USER, Recovery = SIMPLE, Version = 611, Col
lation = SQL_Latin1_General_CP1_CI_AS, SQLSortOrder = 52, IsAutoCreateStatistics, IsAutoUpdateStatist
ics
compatibility_level: 90
name: model
db_size: 1.69 MB
owner: sa
dbid: 3
created: Apr 8 2003
status: Status = ONLINE, Updateability = READ_WRITE, UserAccess = MULTI_USER, Recovery = SIMPLE, Version = 611, Col
lation = SQL_Latin1_General_CP1_CI_AS, SQLSortOrder = 52, IsAutoCreateStatistics, IsAutoUpdateStatist
ics
compatibility_level: 90
name: msdb
db_size: 5.44 MB
owner: sa
dbid: 4
created: Oct 14 2005
status: Status = ONLINE, Updateability = READ_WRITE, UserAccess = MULTI_USER, Recovery = SIMPLE, Version = 611, Col
lation = SQL_Latin1_General_CP1_CI_AS, SQLSortOrder = 52, IsAutoCreateStatistics, IsAutoUpdateStatist
ics, IsFullTextEnabled
compatibility_level: 90
name: tempdb
db_size: 2.50 MB
owner: sa
dbid: 2
created: Jun 12 2007
status: Status = ONLINE, Updateability = READ_WRITE, UserAccess = MULTI_USER, Recovery = SIMPLE, Version = 611, Col
lation = SQL_Latin1_General_CP1_CI_AS, SQLSortOrder = 52, IsAutoCreateStatistics, IsAutoUpdateStatist
ics
compatibility_level: 90

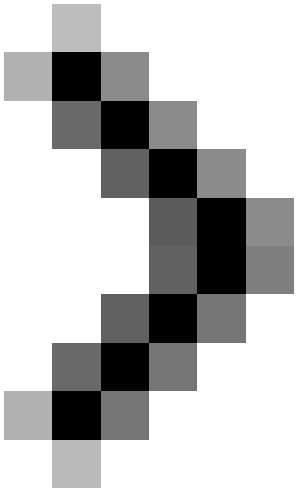
name: test
db_size: 2.68 MB
owner: HOMEMAK
dbid: 5
created: Jan 15 2007
status:
compatibility_level: 90
name: VixiaTrack
db_size: 6.94 MB
owner: HOMEMAK
dbid: 6
created: Apr 22 2007
status:
compatibility_level: 90
name: XMLTest
db_size: 2.68 MB
owner: HOMEMAK
dbid: 7
created: Apr 17 2007
status: Status = ONLINE, Updateability = READ_WRITE, UserAccess = MULTI_USER, Recovery = SIMPLE, Version = 611, Col
lation = SQL_Latin1_General_CP1_CI_AS, SQLSortOrder = 52, IsAutoClose, IsAutoCreateStatistics, IsAuto
UpdateStatistics, IsFullTextEnabled
compatibility_level: 90

Conclude

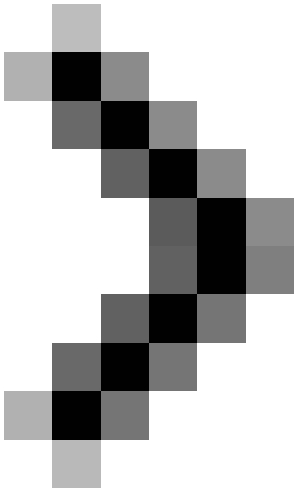
In Part II, we introduced various methods (WMI, .Net classes .) that you can use to recover information on Windows servers and SQL Server.



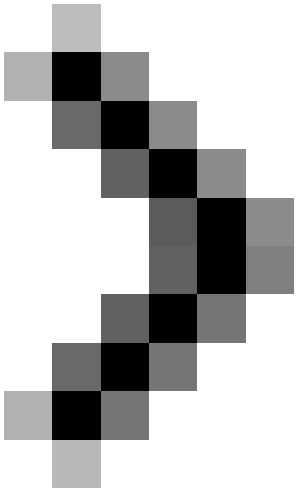
Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 3



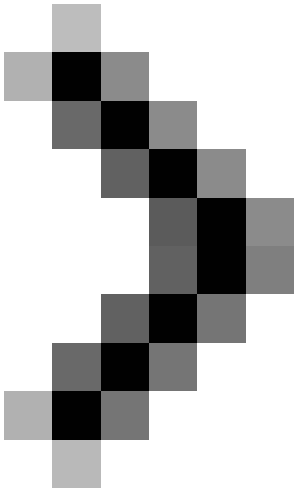
Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 4



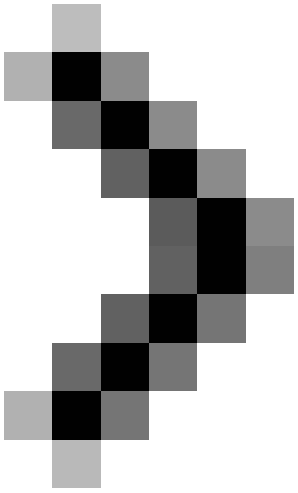
Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 5



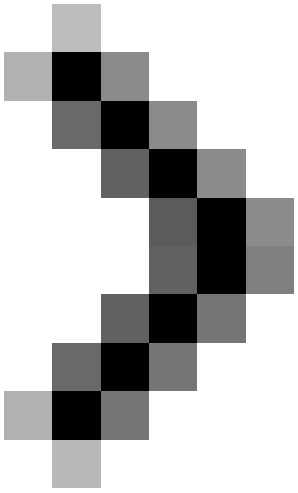
Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 6



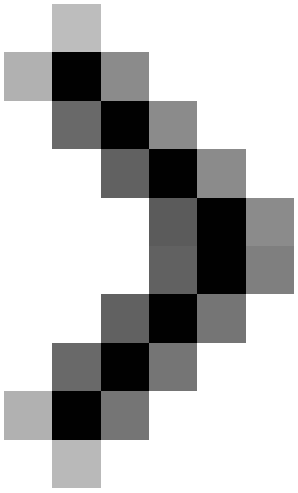
Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 7



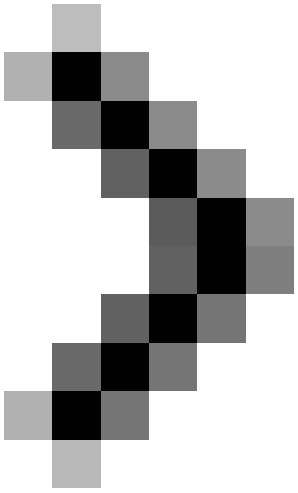
Microsoft Windows PowerShell and SQL Server 2005 SMO - Part 8



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 PowerShell and SQL Server 2005 SMO - Part 2**" 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.
