

SQL-Server mit Powershell verwalten

Die ganze Macht der Powershell-Automation auf dem
SQL-Server

Holger Voges

About_Author



Holger Voges

CCA, MCSE, MCDBA, MCT, MCITP DB
Administrator / DB Developer,
MCTIP Enterprise Administrator

Netz-Weise
Freundallee 13a
30173 Hannover
www.netz-weise.de

Agenda

- SQL-Server Powershell Cmdlets
- Powershell in Agent-Jobs
- Powershell im Management-Studio
- Powershell und SMO - hier kommst dicke!

Sql Server Managed Backup für Azure

- Automatisches Backup zu Azure
- Automatische Verschlüsselung
- Backup wird Datenvolumengesteuert durchgeführt
- Kein Hintergrundwissen zu SQL Backup notwendig

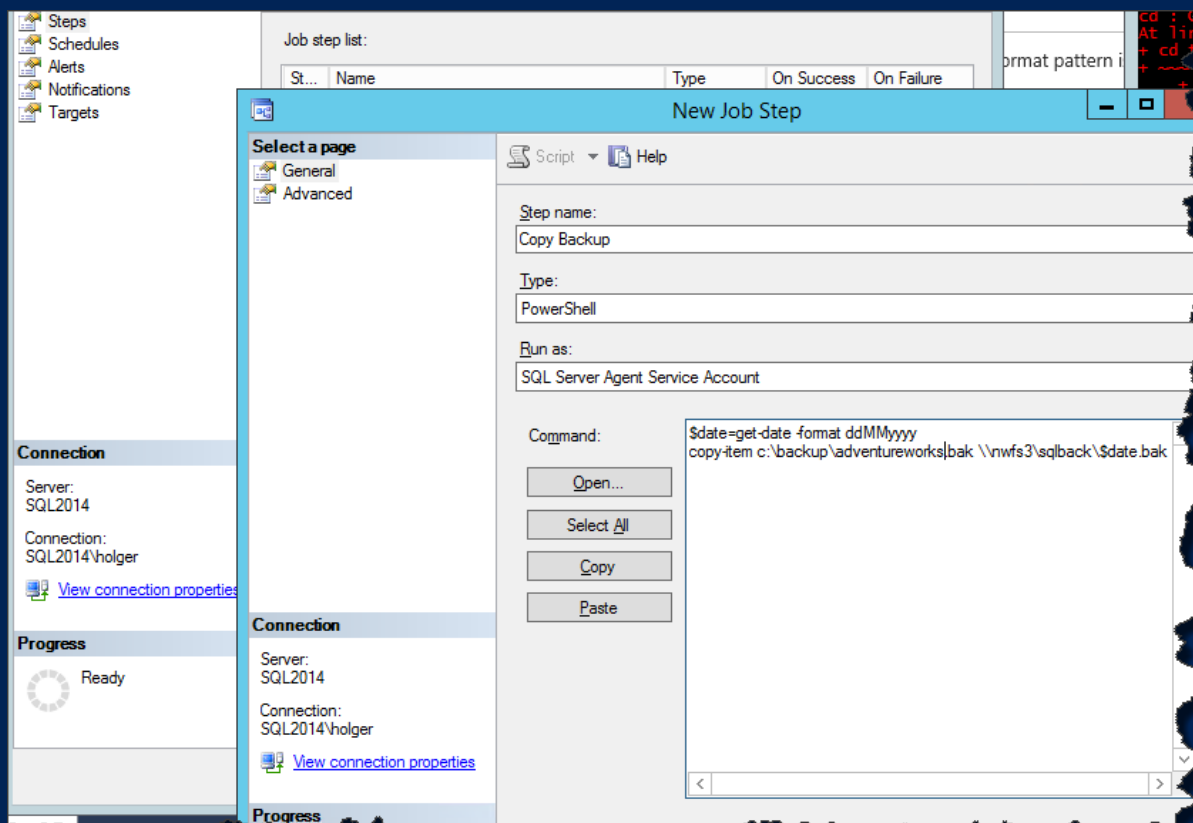


Sample Code.ps1 X

```
cd SQLSERVER:\SQL\Computer\MyInstance
$encryptionOption = New-SqlBackupEncryptionOption
                    -EncryptionAlgorithm Aes128
                    -EncryptorType ServerCertificate
                    -EncryptorName 'MyBackupCert'

Get-SqlSmartAdmin |
    Set-SqlSmartAdmin -BackupEnabled $True
                    -BackupRetentionPeriodInDays 10
                    -EncryptionOption $encryptionOption
```

Agent-Jobs per Powershell starten





Sample Code.ps1 X

```
$RSSvc = "http://localhost/ReportServer/ReportService2010.asmx?wsdl"
$RSProxy = New-WebServiceProxy -uri $RSSvc -class RS2 `
           -Namespace ReportingWebservice -Credential Get-Credential
$RSProxy.ListSubscriptions("")
$id = $RSProxy.ListSubscriptions("") |
      where Report -eq "Sales_Order_Detail_2008R2" |
      select subscriptionID -ExpandProperty `
$RSProxy.FireEvent("TimedSubscription",$id,"")
```

Fehlerbehandlung

- Jobs müssen für den Agent einen terminierenden Fehler zurück melden
- Der Powershell-Provider startet PSH-Jobs immer im SQL-Laufwerk. Soll ein Dateijob gestartet werden, immer erst in den Dateisystem-Provider wechseln -
Set-Location C:

Here-Strings verwenden

- Here-Strings sind ideal um längere SQL-Scripte in Powershell zwischen zu speichern

```
$Query = @'  
SELECT db.name, db.create_date, df.physical_name,  
df.max_size FROM master.sys.databases AS db  
INNER JOIN master.sys.database_files AS df  
ON db.name = df.name  
'@
```

Sample Code.ps1 ×

```
# SQL-Kommando per .NET ausführen
```

```
$ConnStr = "Server=SQLDB;Initial Catalog=Nwind;Integrated Security = True"
```

```
$conn= New-Object System.Data.SqlClient.SqlConnection
```

```
$conn.ConnectionString = $ConnStr
```

```
$Conn.Open()
```

```
$command = New-Object System.Data.SqlClient.SqlCommand
```

```
$command.Connection = $conn
```

```
$statement = "INSERT INTO log (runtime,errors) values  
( '$($endtime)', $($errors) )"
```

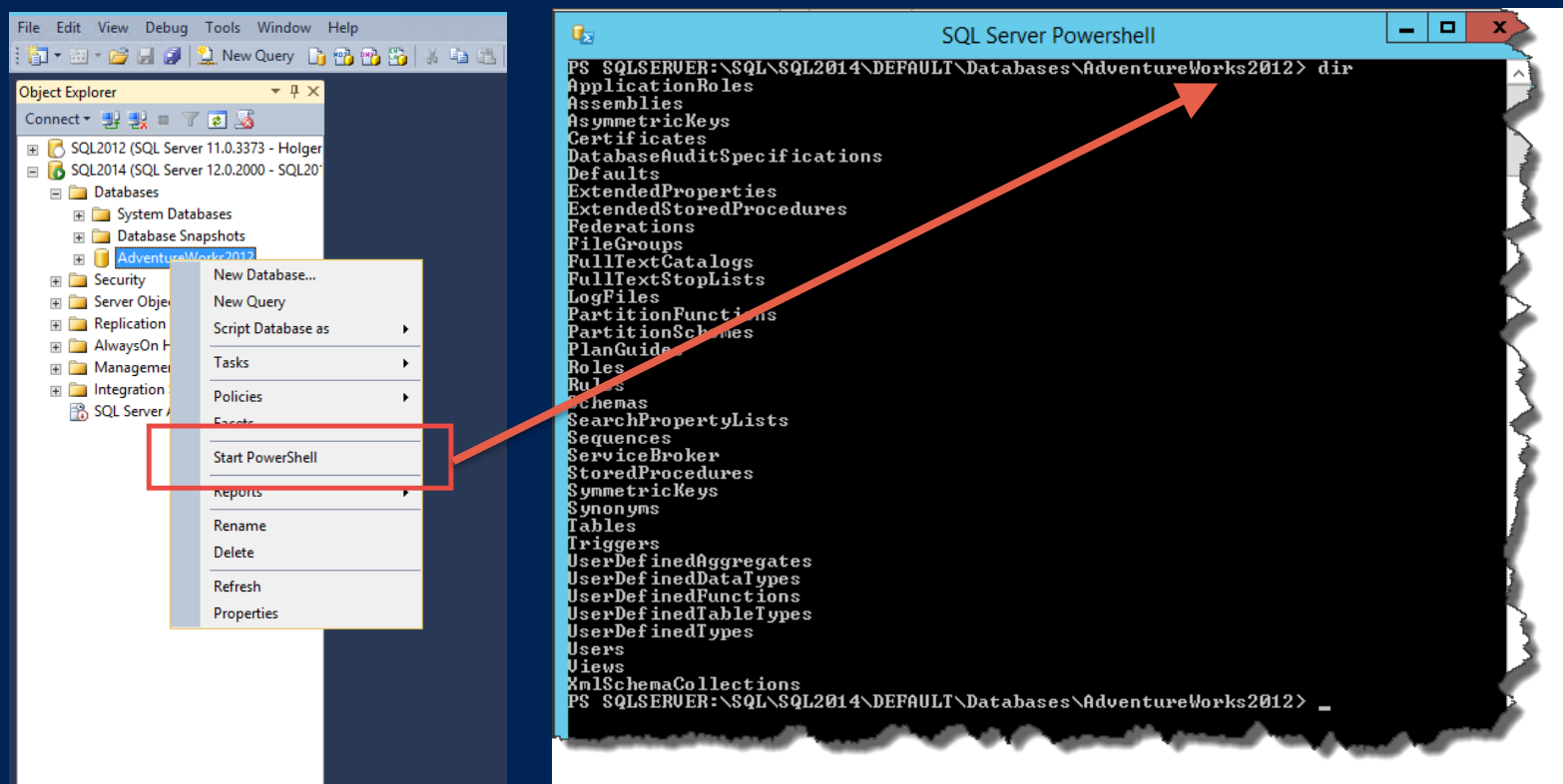
```
$command.CommandText = $Statement
```

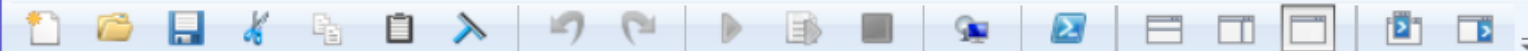
```
$command.ExecuteNonQuery()
```

Server Managed Objects (SMO)

- Grundlage für Powershell
- Wird von PSH automatisch geladen
- Stellt eine API zum SQL-Server dar
- Liefert jede Menge Diagnoseinformationen

SQLPS im Management-Studio

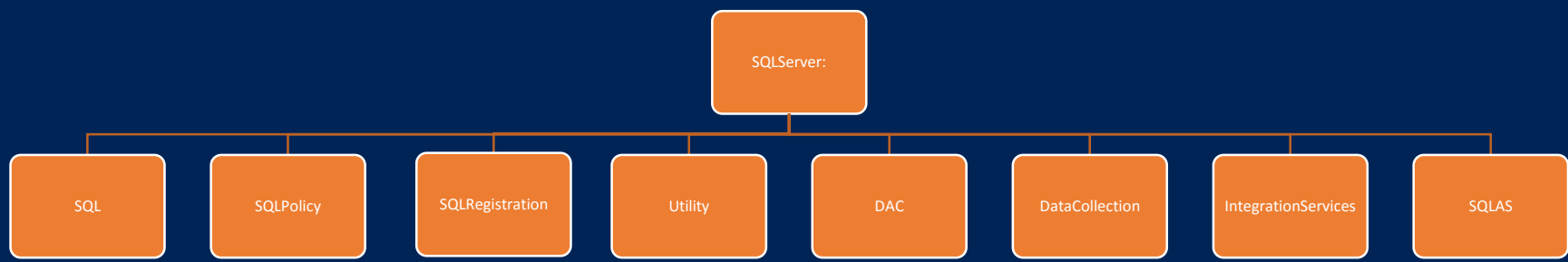




Sample Code.ps1 X

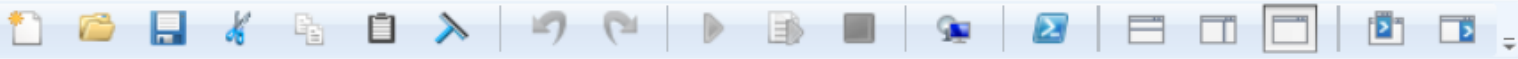
```
$instanceName = „SQL2012“  
$server = New-Object  
-TypeName Microsoft.SqlServer.Management.Smo.Server  
-ArgumentList $instanceName
```

SQL-Server Powershell Provider



Hinweise

- Benötigt Client SDK aus der SQL-Server Installationspaket (im Featurepack online verfügbar)
- SMO-Assemblies werden mit Add-Type oder der Methode LoadWithPartialName() geladen werden
- Add-Type funktioniert mit SMO leider nicht korrekt, daher besser LoadWithPartialName() verwenden
<http://www.madwithpowershell.com/2013/10/add-type-vs-reflectionassembly-in.html>



Sample Code.ps1 ×

Demo

Summary

- Out of Box Support für PSH ist im SQL-Server schlecht
- Mit SMO kann man die volle Macht der Powershell entfalten

Fragen?