# PowerShell scripting in Analysis Services

**SQL Server 2016** 

Applies To: SQL Server 2016

SQL Server 2016 includes PowerShell components to navigate, administer, and query Analysis Services server, tabular, and multidimensional objects:

- **SQLAS** provider, used for navigation of the object hierarchy, is available when you have any local instance of Analysis Services (server mode is irrelevant).
- SQLASCMDLETS module provides task-specific cmdlets, such as backup, restore, process, as well as the general
  purpose Invoke-ASCmd cmdlet that accepts any ASSL/XMLA or Tabular Model Scripting Language (TMSL) query or
  script input file.

Both components implement a subset of the Analysis Services Management Object (Microsoft.AnalysisServices Namespace) administrative interface, providing cmdlets for managing and creating Analysis Services objects. Both are extensions of the **SQLPS** root module for SQL Server. To use Analysis Services PowerShell components, you start by importing **SQLPS**. Syntax and examples for all cmdlets can be found in the Analysis Services PowerShell Reference. For an example of how to use AMO types in PowerShell to create a Tabular database, see AMO PowerShell Example.

# Step 1: Install PowerShell components

The recommended approach for getting PowerShell components is by installing SQL Server Management Studio (SSMS). This approach provides the PowerShell modules for SQL Server and the Analysis Services Management (AMO) data provider. Having SSMS also gives you a tool for easily generating XMLA and TMSL inputs for use in your PowerShell script.

Consider installing a local instance of Analysis Services. A local instance enables navigation of the object hierarchy via the **SQLAS** provider, even if you never use it to host a database.

- Go to Download SQL Server Management Studio to get the latest version of Management Studio. The latest release of Management Studio. includes an updated AMO that supports tabular metadata object definitions, for Tabular models created at compatibility level 1200.
- 2. After installing Management Studio, open a PowerShell window. It doesn't have to be an admin window.
- 3. Enter Get-Module -ListAvailable to confirm that SQLPS and SQLASCMDLETS modules appear in the list.
  - You won't see **SQLAS** unless you also install a local instance of Analysis Services (either Tabular or Multidimensional mode).
- 4. Enter Get-Command -Module sqlascmdlets to produce a list of the cmdlets used in Analysis Services administration.

**SQLASCMDLETS** are available even when the **SQLAS** provider is missing.

- Add-RoleMember cmdlet
- Backup-ASDatabase cmdlet
- Invoke-ASCmd cmdlet -inputfile

- Invoke-ProcessASDatabase
- Invoke-ProcessCube cmdlet
- Invoke-ProcessDimension cmdlet
- Invoke-ProcessPartition cmdlet
- Invoke-ProcessTable cmdlet
- Merge-Partition cmdlet
- New-RestoreFolder cmdlet
- New-RestoreLocation cmdlet
- Remove-RoleMember cmdlet
- Restore-ASDatabase cmdlet



Windows PowerShell is installed by default on newer versions of the Windows operating systems. The recommendation is 4.0 or later.

## Step 2: Load components to start an interactive session

After the components are installed, loading them starts an interactive session.

Enter Import-Module sqlps -DisableNameChecking,

Loads the SQL Server PowerShell components, including those for Analysis Services, and suppresses the warning about invalid verb names.

2. Enter sqlserver:

You should see the prompt change to **PS SQLSERVER:\>**.

- 3. If Analysis Services is installed locally, you can navigate the object hierarchy. Enter cd sqlas to open the **SQLAS** provider.
- 4. Type dir to list Analysis Services instances. The provider does not distinguish between tabular and multidimensional instances.

### **Permissions**

An interactive PowerShell session runs under the security identity of the person who starts it. Most tasks will require that the person initiating the session is also an Analysis Services server administrator.

Scheduled PowerShell tasks should be considered unattended operations. The account running the scheduler, such as SQL Server Agent service, most likely needs to be an Analysis Services administrator (depending on the task).

Local data folders, such as the default backup and data directories, are already configured with file system permissions that allow a local instance to read and write to those locations.

Remote administration, especially when conducted from client computers that don't have Analysis Services installed, requires that the remote Analysis Services server instance performing the action have file system permissions to read files during restore, or write files during backup.

# Script input files: ASSL/XMLA or TMSL

If you are using **invoke-ascmd** to execute script, the server mode, database type, and compatibility level will factor into how you construct the script.

- Multidimensional databases and Tabular databases at 1050-1103 compatibility levels respond to script written in XMLA (using the ASSL extensions specific to Analysis Services object definitions).
- Tabular databases at the SQL Server 2016 (compatibility level 1200) respond to TMSL script.

If you are working with mixed versions of Tabular models on the same SQL Server 2016 instance, remember to use the right script.

#### Note

Scripts can be generated in SQL Server management Studio and then modified as needed. Tabular databases at 1200 compatibility level are scripted in TMSL. All others are scripted in XMLA/ASSL. A SQL Server 2016 instance in Tabular mode supports both scripting languages.

### Local and remote administration

Local administration of Analysis Services via PowerShell scripts and commands is easier for two reasons:

- Enables use of the **SQLAS** provider for navigating the object hierarchy.
- File permissions that enable Analysis Services to read from default data folders, such as for backup and restore tasks, are already in place, assuming you're using those folders as the database location, and the local server instance is used for the operation.

Managing a remote instance requires extra configuration. The following steps assume that you completed the installation steps for Management Studio, but that the service instance is on a remote computer. You must have administrator rights on the Analysis Services server.

Because Analysis Services is remote, there is no **SQLAS** provider for local navigation of the object hierarchy. If you are restoring files from a local folder rather than an Analysis Services instance, you will have to create shares and grant the server read-access to the files.

- 1. Open an administrator PowerShell window.
- 2. Enter Set-ExecutionPolicy RemoteUnsigned

3. In File Explorer, make sure that any folders storing data files are shared, and that the Analysis Services instance has Context read permissions to the contents.

# **Supported Versions and Modes of Analysis Services**

The following table shows the availability of Analysis Services PowerShell in different contexts.

Context	PowerShell Feature Availability
Multidimensional instances and databases	Supported for local and remote administration.
	Merge-partition requires a local connection.
Tabular instances and databases	Supported for local and remote administration, at all compatibility levels.
	SQLAS cmdlets for Tabular models at the 1200 compatibility level using Tabular Model Scripting Language (TMSL) in JSON instead of XMLA.
Power Pivot for SharePoint instances and databases	Limited support. You can use HTTP connections and the SQLAS provider to view instance and database information.
	However, using the cmdlets is not supported. You can't use Analysis Services PowerShell to backup and restore an in-memory Power Pivot database, nor should you add or remove roles, process data, or run arbitrary XMLA script.
	For configuration purposes, Power Pivot for SharePoint has built-in PowerShell support that is provided separately. For more information, see PowerShell Reference for Power Pivot for SharePoint.
Native connections to local cubes	Not supported.
"Data Source=c:\backup\test.cub"	
HTTP connections to BI semantic model (.bism) connection files in SharePoint	Not supported.
"Data Source=http://server/shared _docs/name.bism"	
Embedded connections to Power Pivot databases	Not supported.
"Data Source=\$Embedded\$"	
Local server context in Analysis Services stored procedures	Not supported.
"Data Source=*"	

# **Examples of server administration tasks with PowerShell**

#### List the server properties

Server properties are exposed in several ways. If you are familiar with properties exposed in msmdsrv. ini or the property pages of Management Studio, you'll see from the examples below that the properties are actually returned from different objects.

This script loads an AMO Server object. If you need a fully-qualified property name, you can run this script to return the list.

```
sqlps
$as = New-Object Microsoft.AnalysisServices.Server
$as.connect("server-name\instance-name")
$as.serverproperties
```

This script returns properties and methods at the instance level. From this list, you can read values like **ServerMode**, **Version**, **ProductName**, or **ProductLevel**.

```
sqlps
$as = New-Object Microsoft.AnalysisServices.Server
$as | get-member
```

#### Get the server mode

```
sqlps
$as = New-Object Microsoft.AnalysisServices.Server
$as.ServerMode
```

#### Get the default compatibility level

```
sqlps
$as = New-Object Microsoft.AnalysisServices.Server
$as.DefaultCompatibilityLevel
```

### Get a list of databases

```
sqlps
$as = New-Object Microsoft.AnalysisServices.Server
```

\$as.databases

### Change the port number

This script creates an object for a named instance, declares the port, sets the port, updates the server instance, disconnects the object, and restarts the service. As a verification step, you can open a new connection and return the port.

You can also verify the port in the msmdsrv.ini file or in the server's General properties page in Management Studio.

```
sqlps
$as = New-Object Microsoft.AnalysisServices.Server
$as.connect("server-name\instance-name")
$port = $as.serverproperties['Port']
$port | select *
$port.Value = [int]55555
$as.Update()
$as.Disconnect()
restart-service 'MSOLAP$TABULAR'
$as.connect("server-name\instance-name")
$port | select *
```

### See Also

Grant server admin rights to an Analysis Services instance
Compatibility Level for Tabular models in Analysis Services
Tabular Model Scripting Language (TMSL) Reference
Install SQL Server PowerShell
Manage Tabular Models Using PowerShell
Configure HTTP Access to Analysis Services on Internet Information Services (IIS) 8.0

### **Community Additions**

© 2016 Microsoft