

Contents

Instance Management

Install

Post-install Configuration

- Configure the Windows Firewall

- Configure the service account

- Add administrators

- Features off by default

- Server groups in SSMS

- Determine the Server Mode

- Rename an Analysis Services Instance

Connect to Analysis Services

- Connect from client applications

- Connection String Properties

- Authentication methodologies

- Kerberos authentication

- SPN registration

- HTTP Access

- Client libraries (data providers)

- Disconnect users and sessions

Monitor Analysis Services

- SQL Server Profiler

 - Monitoring Analysis Services with SQL Server Profiler

 - Create Profiler Traces for Replay

- Extended Events

- Dynamic Management Views (DMVs)

- Performance Counters

- Clear the Analysis Services Caches

Script Administrative Tasks

- Create Scripts in Management Studio

Use Analysis Services Templates in SQL Server Management Studio

Use the Analysis Services Scripts Project in Management Studio

Schedule Administrative Tasks with SQL Server Agent

Automate Administrative Tasks with SSIS

High Availability and Scalability

Logging in Analysis Services

DBCC for Analysis Services

Analysis Services server management

5/16/2018 • 3 minutes to read • [Edit Online](#)

A server instance of Analysis Services is a copy of the **msmdsrv.exe** executable that runs as an operating system service. Each instance is fully independent of other instances on the same server, having its own configuration settings, permissions, ports, startup accounts, file storage, and server mode properties.

Each instance runs as Windows service, Msmdsrv.exe, in the security context of a defined logon account.

- The service name of default instance is MSSQLServerOLAPService.
- The service name of each named instance of is MSOLAP\$InstanceName.

NOTE

If multiple instances are installed, Setup also installs a redirector service, which is integrated with the SQL Server Browser service. The redirector service is responsible for directing clients to the appropriate named instance of Analysis Services. The SQL Server Browser service always runs in the security context of the Local Service account, a limited user account used by Windows for services that do not access resources outside the local computer.

Multi-instance means that you can scale-up by installing multiple server instances on the same hardware. For Analysis Services in particular, it also means that you can support different server modes by having multiple instances on the same server, each one configured to run in a specific mode.

Server mode is a server property that determines which storage and memory architecture is used for that instance. A server that runs in Multidimensional mode uses the resource management layer that was built for multidimensional cube databases and data mining models. In contrast, Tabular server mode uses the VertiPaq in-memory analytics engine and data compression to aggregate data as it is requested.

Differences in storage and memory architecture mean that a single instance of Analysis Services will run either tabular databases or multidimensional databases, but not both. The server mode property determines which type of database runs on the instance.

Server mode is set during installation when you specify the type of database that will run on the server. To support all available modes, you can install multiple instances of Analysis Services, each deployed in a server mode that corresponds to the projects you are building.

As a general rule, most of the administrative tasks you must perform do not vary by mode. As an Analysis Services system administrator, you can use the same procedures and scripts to manage any Analysis Services instance on your network regardless of how it was installed.

NOTE

The exception is Power Pivot for SharePoint. Server administration of a Power Pivot deployment is always within the context of a SharePoint farm. Power Pivot differs from other server modes in that it is always single-instance, and always managed through SharePoint Central Administration or the Power Pivot Configuration Tool. Although it is possible to connect to Power Pivot for SharePoint in SQL Server Management Studio or SQL Server Data Tools (SSDT), it is not desirable. A SharePoint farm includes infrastructure that synchronizes server state and oversees server availability. Using other tools can interfere with these operations. For more information about Power Pivot server administration, see [Power Pivot for SharePoint](#).

In this section

LINK	TASK DESCRIPTION
Post-install Configuration	Describes the required and optional tasks that complete or modify an installation of Analysis.
Connect to Analysis Services	Describes connection string properties, client libraries, authentication methodologies, and steps for establishing or clearing connections.
Monitor an Analysis Services Instance	Describes tools and techniques for monitoring a server instance, including how to use Performance Monitor and SQL Server Profiler.
High availability and Scalability	Describes the most commonly used techniques for making Analysis Services databases high available and scalable.
Globalization scenarios for Analysis Services	Explains language and collation support, steps for changing both properties, and tips for setting and testing language and collation behaviors.
Log operations in Analysis Services	Describes the logs and explains how to configure them.


See also

[Comparing Tabular and Multidimensional Solutions](#)

[Determine the Server Mode of an Analysis Services Instance](#)

Install SQL Server Analysis Services

6/1/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

SQL Server Analysis Services is an analytical database server that hosts Tabular models, multidimensional cubes, and data mining models that you can access from reports, spreadsheets, and dashboards.

Analysis Services is multi-instance, which means that you can install more than one copy on a single computer, or run new and old versions side-by-side. Any instance you install runs in one of three modes, as determined during setup: Multidimensional and Data Mining, Tabular, or SharePoint. If you want to use multiple modes, you'll need a separate instance for each one.

After you install the server in a particular mode, you can use it host solutions that conform to that mode. For example, a tabular mode server is required if you want tabular model data access over the network.

Get tools and designers

SQL Server Setup no longer installs the model designers or management tools used for solution design or server administration. In this release, tools have a separate installation, which you can get from the following links:

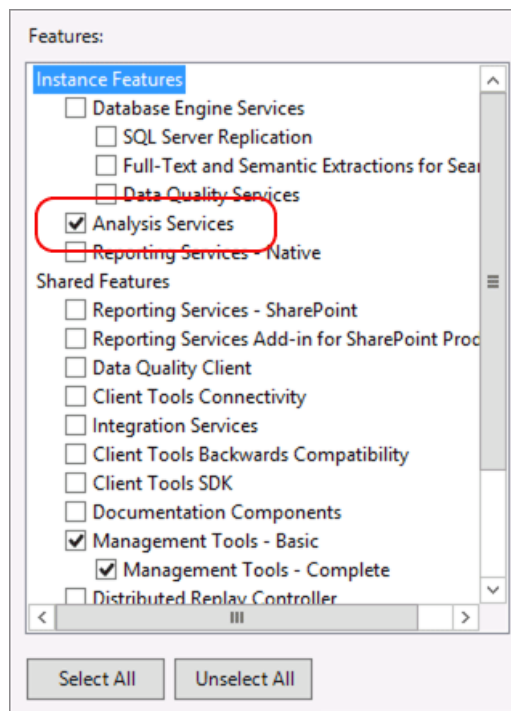
- [Download SQL Server Management Studio \(SSMS\)](#)
- [Download SQL Server Data Tools \(SSDT\)](#)

You'll need both SSMS and SSDT to work with Analysis Services instances and data. Tools can be installed anywhere, but be sure to configure ports on the server before attempting a connection. See [Configure the Windows Firewall to Allow Analysis Services Access](#) for details.

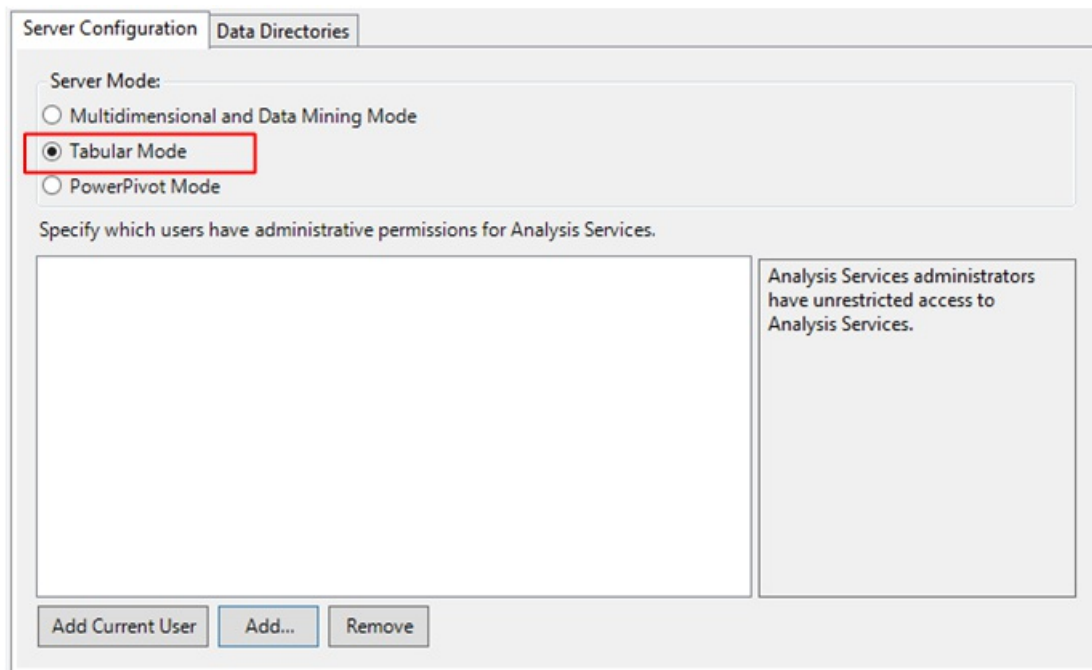
Install using a wizard

The following list shows you which pages in the SQL Server Installation wizard are used to install Analysis Services.

1. Select **Analysis Services** from the Feature Tree in Setup.



2. On the Analysis Services Configuration page, select a mode. Tabular mode is the default..



Tabular mode uses the xVelocity in-memory analytics engine (VertiPaq), which is the default storage for tabular models. After you deploy tabular models to the server, you can selectively configure tabular solutions to use DirectQuery disk storage as an alternative to memory-bound storage.

Multidimensional and Data Mining mode use MOLAP as the default storage for models deployed to Analysis Services. After deploying to the server, you can configure a solution to use ROLAP if you want to run queries directly against the relational database rather than storing query data in an Analysis Services multidimensional database .

Memory management and IO settings can be adjusted to get better performance when using non-default storage modes. See [Server Properties in Analysis Services](#) for more information.

Command Line Setup

SQL Server Setup includes a parameter (**ASSERVERMODE**) that specifies the server mode. The following

example illustrates a command line setup that installs Analysis Services in Tabular server mode.

```
Setup.exe /q /IAcceptSQLServerLicenseTerms /ACTION=install /FEATURES=AS /ASSERVERMODE=TABULAR
/INSTANCENAME=ASTabular /INDICATEPROGRESS /ASSVCACCOUNT=<DomainName\UserName> /ASSVCPASSWORD=<StrongPassword>
/ASSYSADMINACCOUNTS=<DomainName\UserName>
```

INSTANCENAME must be less than 17 characters.

All placeholder account values must be replaced with valid accounts and password.

ASSERVERMODE is case-sensitive. All values must be expressed in upper case. The following table describes the valid values for **ASSERVERMODE**.


VALUE	DESCRIPTION
TABULAR	This is the default value. If you do not set ASSERVERMODE , the server is installed in Tabular mode.
MULTIDIMENSIONAL	This value is optional.
POWERPIVOT	This value is optional. In practice, if you set the ROLE parameter, the server mode is automatically set to 1, making ASSERVERMODE optional for a Power Pivot for SharePoint installation. For more information, see Install Power Pivot from the Command Prompt .

See Also

[Determine the Server Mode of an Analysis Services Instance](#)
[Tabular Modeling](#)

Post-install Configuration (Analysis Services)

5/16/2018 • 3 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

After installing Analysis Services, further configuration is required to make the server fully operational and available for general use. This section introduces the additional tasks that complete the installation. Depending on connection requirements, you might also need to configure authentication (see [Connect to Analysis Services](#)).

Later, additional work will be required once you have databases that are ready to deploy. Namely, you will need to configure role memberships on the database to grant user access to the data, design a database backup and recovery strategy, and determine whether you need a scheduled processing workload to refresh data at regular intervals. More information about database deployment and administration can be found at these links:

[Multidimensional Model Databases](#) and [Tabular Model Databases](#).

Instance Configuration

Analysis Services is a replicable service, meaning you can install multiple instances of the service on a single server. Each additional instance is installed separately as a named instance, using SQL Server Setup, and configured independently to support its intended purpose. For example, a development server might run Flight Recorder or use default values for data storage that you might otherwise change on servers supporting production workloads. Another example that calls for adjusting system configuration is installing Analysis Services instance on hardware shared by other services. When hosting multiple data-intensive applications on the same hardware, you might want to configure server properties that lower the memory thresholds to optimize available resources across all of the applications.

Post-installation Tasks

LINK	TASK DESCRIPTION
Configure the Windows Firewall to Allow Analysis Services Access	Create an inbound rule in Windows Firewall so that requests can be routed through the TCP port used by the Analysis Services instance. This task is required. No one can access Analysis Services from a remote computer until an inbound firewall rule is defined.
Grant server admin rights to an Analysis Services instance	During installation, you had to add at least one user account to the Administrator role of the Analysis Services instance. Administrative permissions are required for many routine server operations, such as processing data from external relational databases. Use the information in this topic to add or modify the membership of the Administrator role.
Configure antivirus software on computers running SQL Server	You might need to configure scanning software, such as antivirus and antispymware applications, to exclude SQL Server folders and file types. If scanning software locks a program or data file when Analysis Services needs to use it, service disruption or data corruption can occur.

LINK	TASK DESCRIPTION
Configure Service Accounts (Analysis Services)	During installation, the Analysis Services service account was provisioned, with appropriate permissions to allow controlled access to program executables and database files. As a post-installation task, you should now consider whether to allow the use of the service account when performing additional tasks. Both processing and query workloads can be executed under the service account. These operations succeed only when the service account has appropriate permissions.
Register an Analysis Services Instance in a Server Group	SQL Server Management Studio (SSMS) lets you create server groups for organizing your SQL Server instances. Scalable deployments consisting of multiple server instances are easier to manage in server groups. Use the information in this topic to organize Analysis Services instances into groups in SSMS.
Determine the Server Mode of an Analysis Services Instance	During installation, you chose a server mode that determines the type of model (multidimensional or tabular) that runs on the server. If you are unsure of the server mode, use the information in this topic to determine which mode was installed.
Rename an Analysis Services Instance	A descriptive name can help you distinguish among multiple instances having different server modes, or among instances primarily used by departments or teams in your organization. If you want to change the instance name to one that helps you better manage your installations, use the information in this topic to learn how.

Next Steps

Learn how to connect to Analysis Services from Microsoft applications or custom applications using the client libraries. Depending on your solution requirements, you might also need to configure the service for Kerberos authentication. Connections that must cross domain boundaries will require HTTP access. See [Connect to Analysis Services](#) for instructions about the next steps.

See Also

[Installation for SQL Server 2016](#)


[Install Analysis Services in Multidimensional and Data Mining Mode](#)

[Install Analysis Services](#)

[Install Analysis Services in Power Pivot Mode](#)

Configure the Windows Firewall to Allow Analysis Services Access

5/16/2018 • 15 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

An essential first step in making Analysis Services or Power Pivot for SharePoint available on the network is to determine whether you need to unblock ports in a firewall. Most installations will require that you create at least one in-bound firewall rule that allows connections to Analysis Services.

Firewall configuration requirements vary depending on how you installed Analysis Services:

- Open TCP port 2383 when installing a default instance or creating an Analysis Services failover cluster.
- Open TCP port 2382 when installing a named instance. Named instances use dynamic port assignments. As the discovery service for Analysis Services, SQL Server Browser service listens on TCP port 2382 and redirects the connection request to the port currently used by Analysis Services.
- Open TCP port 2382 when installing Analysis Services in SharePoint mode to support Power Pivot for SharePoint 2013. In Power Pivot for SharePoint 2013, the Analysis Services instance is external to SharePoint. Inbound requests to the named 'Power Pivot' instance originate from SharePoint web applications over a network connection, requiring an open port. As with other Analysis Services named instances, create an inbound rule for SQL Server Browser service on TCP 2382 to allow access to Power Pivot for SharePoint.
- For Power Pivot for SharePoint 2010, do not open ports in Windows Firewall. As an add-in to SharePoint, the service uses ports configured for SharePoint and makes only local connections to the Analysis Services instance that loads and queries Power Pivot data models.
- For Analysis Services instances running on Windows Azure Virtual Machines, use alternate instructions for configuring server access. See [SQL Server Business Intelligence in Windows Azure Virtual Machines](#).

Although the default instance of Analysis Services listens on TCP port 2383, you can configure the server to listen on a different fixed port, connecting to the server in this format: <servername>:<portnumber>.

Only one TCP port can be used by an Analysis Services instance. On computers having multiple network cards or multiple IP addresses, Analysis Services listens on one TCP port for all IP addresses assigned or aliased to the computer. If you have specific multi-port requirements, consider configuring Analysis Services for HTTP access. You can then set up multiple HTTP endpoints on whatever ports you choose. See [Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#).

This topic contains the following sections:

- [Check port and firewall settings for Analysis Services](#)
- [Configure Windows Firewall for a default instance of Analysis Services](#)
- [Configure Windows Firewall access for a named instance of Analysis Services](#)
- [Port configuration for an Analysis Services cluster](#)
- [Port configuration for Power Pivot for SharePoint](#)
- [Use a fixed port for a default or named instance of Analysis Services](#)

For more information about the default Windows firewall settings, and a description of the TCP ports that affect the Database Engine, Analysis Services, Reporting Services, and Integration Services, see [Configure the Windows Firewall to Allow SQL Server Access](#).

Check port and firewall settings for Analysis Services

On the Microsoft Windows operating systems that are supported by SQL Server 2017, Windows Firewall is on by default and is blocking remote connections. You must manually open a port in the firewall to allow inbound requests to Analysis Services. SQL Server Setup does not perform this step for you.

Port settings are specified in the msmdsrv.ini file and in the General properties page of an Analysis Services instance in SQL Server Management Studio. If **Port** is set to a positive integer, the service is listening on a fixed port. If **Port** is set to 0, the service is listening on port 2383 if it is the default instance or on a dynamically assigned port if it is a named instance.

Dynamic port assignments are only used by named instances. The **MSOLAP\$InstanceName** service determines which port to use when it starts up. You can determine the actual port number in use by a named instance by doing the following:

- Start Task Manager and then click **Services** to get the PID of the **MSOLAP\$InstanceName**.
- Run **netstat -ao -p TCP** from the command line to view the TCP port information for that PID.
- Verify the port by using SQL Server Management Studio and connect to an Analysis Services server in this format: <IPAddress>:<portnumber>.

Although an application might be listening on a specific port, connections will not succeed if a firewall is blocking access. In order for connections to reach a named Analysis Services instance, you must unblock access to either msmdsrv.exe or the fixed port on which it is listening in the firewall. The remaining sections in this topic provide instructions for doing so.

To check whether firewall settings are already defined for Analysis Services, use Windows Firewall with Advanced Security in Control Panel. The Firewall page in the Monitoring folder shows a complete list of the rules defined for the local server.

Note that for Analysis Services, all firewall rules must be manually defined. Although Analysis Services and SQL Server Browser reserve ports 2382 and 2383, neither the SQL Server setup program nor any of the configuration tools define firewall rules that allow access to either the ports or the program executable files.

Configure Windows Firewall for a default instance of Analysis Services

The default instance of Analysis Services listens on TCP port 2383. If you installed the default instance and want to use this port, you only need to unblock inbound access to TCP port 2383 in Windows Firewall to enable remote access to the default instance of Analysis Services. If you installed the default instance but want to configure the service to listen on a fixed port, see [Use a fixed port for a default or named instance of Analysis Services](#) in this topic.

To verify whether the service is running as the default instance (MSSQLServerOLAPService), check the service name in SQL Server Configuration Manager. A default instance of Analysis Services is always listed as **SQL Server Analysis Services (MSSQLSERVER)**.

NOTE

Different Windows operating systems provide alternative tools for configuring Windows Firewall. Most of these tools let you choose between opening a specific port or program executable. Unless you have a reason for specifying the program executable, we recommend that you specify the port.

When specifying an inbound rule, be sure to adopt a naming convention that allows you to easily find the rules later (for example, **SQL Server Analysis Services (TCP-in) 2383**).

Windows Firewall with Advanced Security

1. On Windows 7 or Windows Vista, in Control Panel, click **System and Security**, select **Windows Firewall**, and then click **Advanced settings**. On Windows Server 2008 or 2008 R2, open Administrator Tools and click **Windows Firewall with Advanced Security**. On Windows Server 2012, open the Applications page and type **Windows Firewall**.
2. Right-click **Inbound Rules** and select **New Rule**.
3. In Rule Type, click **Port** and then click **Next**.
4. In Protocol and Ports, select **TCP** and then type **2383** in **Specific local ports**.
5. In Action, click **Allow the connection** and then click **Next**.
6. In Profile, clear any network locations that do not apply and then click **Next**.
7. In Name, type a descriptive name for this rule (for example, **SQL Server Analysis Services (tcp-in) 2383**), and then click **Finish**.
8. To verify that remote connections are enabled, open SQL Server Management Studio or Excel on a different computer and connect to Analysis Services by specifying the network name of the server in **Server name**.

NOTE

Other users will not have access to this server until you grant permissions. For more information, see [Authorizing access to objects and operations \(Analysis Services\)](#).

Netsh AdvFirewall Syntax

- The following command creates an inbound rule that allows incoming requests on TCP port 2383.

```
netsh advfirewall firewall add rule name="SQL Server Analysis Services inbound on TCP 2383" dir=in  
action=allow protocol=TCP localport=2383 profile=domain
```

Configure Windows Firewall access for a named instance of Analysis Services

Named instances of Analysis Services can either listen on a fixed port or on a dynamically assigned port, where SQL Server Browser service provides the connection information that is current for the service at the time of the connection.

SQL Server Browser service listens on TCP port 2382. UDP is not used. TCP is the only transmission protocol used by Analysis Services.

Choose one of the following approaches to enable remote access to a named instance of Analysis Services:

- Use dynamic port assignments and SQL Server Browser service. Unblock the port used by SQL Server Browser service in Windows Firewall. Connect to the server in this format: <servername>\<instancename>.
- Use a fixed port and SQL Server Browser service together. This approach lets you connect using this format: <servername>\<instancename>, identical to the dynamic port assignment approach, except that in this case the server listens on a fixed port. In this scenario, SQL Server Browser Service provides name resolution to the Analysis Services instance listening on the fixed port. To use this approach, configure the

server to listen on a fixed port, unblock access to that port, and unblock access to the port used by SQL Server Browser service.

SQL Server Browser service is only used with named instances, never with the default instance. The service is automatically installed and enabled whenever you install any SQL Server feature as a named instance. If you choose an approach that requires SQL Server Browser service, be sure it remains enabled and started on your server.

If you cannot use SQL Server Browser service, you must assign a fixed port in the connection string, bypassing domain name resolution. Without SQL Server Browser service, all client connections must include the port number on the connection string (for example, AW-SRV01:54321).

Option 1: Use dynamic port assignments and unblock access to SQL Server Browser service

Dynamic port assignments for named instances of Analysis Services are established by the **MSOLAP\$InstanceName** when the service starts. By default, the service claims the first available port number that it finds, using a different port number each time the service is restarted.

Instance name resolution is handled by the SQL Server browser service. Unblocking TCP port 2382 for SQL Server Browser service is always required if you are using dynamic port assignments with a named instance.

NOTE

SQL Server Browser service listens on both UDP port 1434 and TCP port 2382 for the Database Engine and Analysis Services, respectively. Even if you already unblocked UDP port 1434 for the SQL Server Browser service, you must still unblock TCP port 2382 for Analysis Services.

Windows Firewall with Advanced Security

1. On Windows 7 or Windows Vista, in Control Panel, click **System and Security**, select **Windows Firewall**, and then click **Advanced settings**. On Windows Server 2008 or 2008 R2, open Administrator Tools and click **Windows Firewall with Advanced Security**. On Windows Server 2012, open the Applications page and type **Windows Firewall**.
2. To unblock access to SQL Server Browser service, right-click **Inbound Rules** and select **New Rule**.
3. In Rule Type, click **Port** and then click **Next**.
4. In Protocol and Ports, select **TCP** and then type **2382** in **Specific local ports**.
5. In Action, click **Allow the connection** and then click **Next**.
6. In Profile, clear any network locations that do not apply and then click **Next**.
7. In Name, type a descriptive name for this rule (for example, **SQL Server Browser Service (tcp-in) 2382**), and then click **Finish**.
8. To verify that remote connections are enabled, open SQL Server Management Studio or Excel on a different computer and connect to the Analysis Services by specifying the network name of the server and the instance name in this format: <servername>\<instancename>. For example, on a server named **AW-SRV01** with a named instance of **Finance**, the server name is **AW-SRV01\Finance**.

Option 2: Use a fixed port for a named instance

Alternatively, you can assign a fixed port, and then unblock access to that port. This approach offers better auditing capability than if you allowed access to the program executable. For this reason, using a fixed port is the recommended approach for accessing any Analysis Services instance.

To assign a fixed port, follow the instructions in [Use a fixed port for a default or named instance of Analysis Services](#) in this topic, then return to this section to unblock the port.

Windows Firewall with Advanced Security

1. On Windows 7 or Windows Vista, in Control Panel, click **System and Security**, select **Windows Firewall**, and then click **Advanced settings**. On Windows Server 2008 or 2008 R2, open Administrator Tools and click **Windows Firewall with Advanced Security**. On Windows Server 2012, open the Applications page and type **Windows Firewall**.
2. To unblock access to Analysis Services, right-click **Inbound Rules** and select **New Rule**.
3. In Rule Type, click **Port** and then click **Next**.
4. In Protocol and Ports, select **TCP** and then type the fixed port in **Specific local ports**.
5. In Action, click **Allow the connection** and then click **Next**.
6. In Profile, clear any network locations that do not apply and then click **Next**.
7. In Name, type a descriptive name for this rule (for example, **SQL Server Analysis Services on port 54321**), and then click **Finish**.
8. To verify that remote connections are enabled, open SQL Server Management Studio or Excel on a different computer and connect to the Analysis Services by specifying the network name of the server and the port number in this format: <servername>:<portnumber>.

Netsh AdvFirewall Syntax

- The following commands create inbound rules that unblock TCP 2382 for SQL Server Browser service and unblock the fixed port that you specified for the Analysis Services instance. You can run either one to allow access to a named Analysis Services instance.

In this sample command, port 54321 is the fixed port. Be sure to replace it with the actual port in use on your system.

```
netsh advfirewall firewall add rule name="SQL Server Analysis Services (tcp-in) on 54321" dir=in  
action=allow protocol=TCP localport=54321 profile=domain
```

```
netsh advfirewall firewall add rule name="SQL Server Browser Services inbound on TCP 2382" dir=in  
action=allow protocol=TCP localport=2382 profile=domain
```

Use a fixed port for a default or named instance of Analysis Services

This section explains how to configure Analysis Services to listen on a fixed port. Using a fixed port is common if you installed Analysis Services as a named instance, but you can also use this approach if business or security requirements specify that you use non-default port assignments.

Note that using a fixed port will alter the connection syntax for the default instance by requiring you to append the port number to the server name. For example, connecting to a local, default Analysis Services instance listening on port 54321 in SQL Server Management Studio would require that you type localhost:54321 as the server name in the Connect to Server dialog box in Management Studio.

If you are using a named instance, you can assign a fixed port with no changes to how you specify the server name (specifically, you can use <servername\instancename> to connect to a named instance listening on a fixed port). This works only if SQL Server Browser service is running and you unblocked the port on which it is listening. SQL Server Browser service will provide redirection to the fixed port based on <servername\instancename>. As long as you open ports for both SQL Server Browser service and the named

instance of Analysis Services listening on the fixed port, SQL Server Browser service will resolve the connection to a named instance.

1. Determine an available TCP/IP port to use.

To view a list of reserved and registered ports that you should avoid using, see [Port Numbers \(IANA\)](#). To view a list of ports that are already in use on your system, open a command prompt window and type **netstat -a -p TCP** to display a list of the TCP ports that are open on the system.

2. After you determine which port to use, specify the port by either editing the **Port** configuration setting in the msmdsrv.ini file or in the General properties page of an Analysis Services instance in SQL Server Management Studio.
3. Restart the service.
4. Configure Windows Firewall to unblock the TCP port you specified. Or, if you are using a fixed port for a named instance, unblock both the TCP port you specified for that instance and TCP port 2382 for SQL Server Browser service.
5. Verify by connecting locally (in Management Studio) and then remotely from a client application on another computer. To use Management Studio, connect to an Analysis Services default instance by specifying a server name in this format: <servername>:<portnumber>. For a named instance, specify the server name as <servername>\<instancename>.

Port configuration for an Analysis Services cluster

An Analysis Services failover cluster always listens on TCP port 2383, regardless of whether you installed it as a default instance or named instance. Dynamic port assignments are not used by Analysis Services when it is installed on a Windows failover cluster. Be sure to open TCP 2383 on every node running Analysis Services in the cluster. For more information about clustering Analysis Services, see [How to Cluster SQL Server Analysis Services](#).

Port configuration for Power Pivot for SharePoint

Server architecture for Power Pivot for SharePoint is fundamentally different depending on which version of SharePoint you are using.

SharePoint 2013

In SharePoint 2013, Excel Services redirects requests for Power Pivot data models, which are subsequently loaded on an Analysis Services instance outside of the SharePoint environment. Connections follow the typical pattern, where an Analysis Services client library on a local computer sends a connection request to a remote Analysis Services instance in the same network.

Because Power Pivot for SharePoint always installs Analysis Services as a named instance, you should assume SQL Server Browser service and dynamic port assignments. As noted earlier, SQL Server Browser service listens on TCP port 2382 for connection requests sent to Analysis Services named instances, redirecting the request to the current port.

Note that Excel Services in SharePoint 2013 does not support the fixed port connection syntax, so make sure SQL Server Browser service is accessible.

SharePoint 2010

If you are using SharePoint 2010, you do not need to open ports in Windows Firewall. SharePoint opens the ports that it requires, and add-ins such as Power Pivot for SharePoint operate within the SharePoint environment. In a Power Pivot for SharePoint 2010 installation, the Power Pivot System Service has exclusive use of the local SQL Server Analysis Services (Power Pivot) service instance that is installed with it on the same computer. It uses

local connections, not network connections, to access the local Analysis Services engine service that loads, queries, and processes Power Pivot data on the SharePoint server. To request Power Pivot data from client applications, requests are routed through ports that are opened by SharePoint Setup (specifically, inbound rules are defined to allow access to SharePoint – 80, SharePoint Central Administration v4, SharePoint Web Services, and SPUserCodeV4). Because Power Pivot web services run within a SharePoint farm, the SharePoint firewall rules are sufficient for remote access to Power Pivot data in a SharePoint farm.

See Also


[SQL Server Browser Service \(Database Engine and SSAS\)](#)

[Start, Stop, Pause, Resume, Restart the Database Engine, SQL Server Agent, or SQL Server Browser Service](#)

[Configure a Windows Firewall for Database Engine Access](#)

Configure Service Accounts (Analysis Services)

5/16/2018 • 13 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Product-wide account provisioning is documented in [Configure Windows Service Accounts and Permissions](#), a topic that provides comprehensive service account information for all SQL Server services, including Analysis Services. Refer to it to learn about valid account types, Windows privileges assigned by setup, file system permissions, registry permissions, and more.

This topic provides supplemental information for Analysis Services, including additional permissions necessary for tabular and clustered installations. It also covers permissions needed to support server operations. For example, you can configure processing and query operations to execute under the service account — in which case you will need to grant additional permissions to get this to work.

- [Windows privileges assigned to Analysis Services](#)
- [File System Permissions assigned to Analysis Services](#)
- [Granting additional permissions for specific server operations](#)

An additional configuration step, not documented here, is to register a Service Principal Name (SPN) for the Analysis Services instance and service account. This step enables pass-through authentication from client applications to backend data sources in double-hop scenarios. This step only applies for services configured for Kerberos constrained delegation. See [Configure Analysis Services for Kerberos constrained delegation](#) for further instructions.

Logon account recommendations

In a failover cluster, all instances of Analysis Services should be configured to use a Windows domain user account. Assign the same account to all instances. See [How to Cluster Analysis Services](#) for details.

Standalone instances should use the default virtual account, **NT Service\MSSQLServerOLAPService** for the default instance, or **NT Service\MSOLAP\$*instance-name*** for a named instance. This recommendation applies to Analysis Services instances in all server modes, assuming Windows Server 2008 R2 and later for the operating system, and SQL Server 2012 and later for Analysis Services.

Granting permissions to Analysis Services

This section explains the permissions that Analysis Services requires for local, internal operations, such as starting the executable, reading the configuration file, and loading databases from the data directory. If instead you're looking for guidance on setting permissions for external data access and interoperability with other services and applications, see [Granting additional permissions for specific server operations](#) further on in this topic.

For internal operations, the permission holder in Analysis Services is not the logon account, but a local Windows security group created by Setup that contains the per-service SID. Assigning permissions to the security group is consistent with previous versions of Analysis Services. Also, logon accounts can change over time, but the per-service SID and local security group are constant for the lifetime of the server installation. For Analysis Services, this makes the security group, rather than the logon account, a better choice for holding permissions. Whenever you manually grant rights to the service instance, whether file system permissions or Windows privileges, be sure to grant permissions to the local security group created for the server instance.

The name of the security group follows a pattern. The prefix is always **SQLServerMSASUser\$**, followed by the

computer name, ending with the instance name. The default instance is **MSSQLSERVER**. A named instance is the name given during set up.

You can see this security group in the local security settings:

- Run `compmgmt.msc` | **Local Users and Groups** | **Groups** | **SQLServerMSASUser\$** <server-name> **\$MSSQLSERVER** (for a default instance).
- Double-click the security group to view its members.

The sole member of the group is the per-service SID. Right next to it is the logon account. The logon account name is cosmetic, there to provide context to the per-service SID. If you subsequently change the logon account and then return to this page, you'll notice that the security group and per-service SID do not change, but the logon account label is different.

Windows privileges assigned to the Analysis Services service account

Analysis Services needs permissions from the operating system for service startup and to request system resources. Requirements vary by server mode and whether the instance is clustered. If you are unfamiliar with Windows privileges, see [Privileges](#) and [Privilege Constants \(Windows\)](#) for details.

All instances of Analysis Services require the **Log on as a service** (SeServiceLogonRight) privilege. SQL Server Setup assigns the privilege for you on the service account specified during installation. For servers running in Multidimensional and Data Mining mode, this is the only Windows privilege required by the Analysis Services service account for standalone server installations, and it is the only privilege that Setup configures for Analysis Services. For clustered and tabular instances, additional Windows privileges must be added manually.

Failover cluster instances, in either Tabular or Multidimensional mode, must have **Increase scheduling priority** (SeIncreaseBasePriorityPrivilege).

Tabular instances use the following three additional privileges, which must be granted manually after the instance is installed.

Increase a process working set (SeIncreaseWorkingSetPrivilege)	This privilege is available to all users by default through the Users security group. If you lock down a server by removing privileges for this group, Analysis Services might fail to start, logging this error: "A required privilege is not held by the client." When this error occurs, restore the privilege to Analysis Services by granting it to the appropriate Analysis Services security group.
Adjust memory quotas for a process (SeIncreaseQuotaSizePrivilege)	This privilege is used to request more memory if a process has insufficient resources to complete its execution, subject to the memory thresholds established for the instance.

<p>Lock pages in memory (SeLockMemoryPrivilege)</p>	<p>This privilege is needed only when paging is turned off entirely. By default, a tabular server instance uses the Windows paging file, but you can prevent it from using Windows paging by setting VertiPagingPolicy to 0.</p> <p>VertiPagingPolicy to 1 (default), instructs the tabular server instance to use the Windows paging file. Allocations are not locked, allowing Windows to page out as needed. Because paging is being used, there is no need to lock pages in memory. Thus, for the default configuration (where VertiPagingPolicy = 1), you do not need to grant the Lock pages in memory privilege to a tabular instance.</p> <p>VertiPagingPolicy to 0. If you turn off paging for Analysis Services, allocations are locked, assuming the Lock pages in memory privilege is granted to the tabular instance. Given this setting and the Lock pages in memory privilege, Windows cannot page out memory allocations made to Analysis Services when the system is under memory pressure. Analysis Services relies on the Lock pages in memory permission as the enforcement behind VertiPagingPolicy = 0. Note that turning off Windows paging is not recommended. It will increase the rate of out-of-memory errors for operations that might otherwise succeed if paging were allowed. See Memory Properties for more information about VertiPagingPolicy.</p>
--	--

To view or add Windows privileges on the service account

1. Run GPEdit.msc | Local Computer Policy | Computer Configuration | Windows Settings | Security Settings | Local Policies | User Rights Assignments.
2. Review existing policies that include **SQLServerMSASUser\$**. This is a local security group found on computers having an Analysis Services installation. Both Windows privileges and file folder permissions are granted to this security group. Double-click **Log on as a service** policy to see how the security group is specified on your system. The full name of the security group will vary depending on whether you installed Analysis Services as a named instance. Use this security group, rather than the actual service account, when adding account privileges.
3. To add account privileges in GPEdit, right-click **Increase a process working set** and select **Properties**.
4. Click **Add User or Group**.

5. Enter the user group for the Analysis Services instance. Remember that the service account is a member of a local security group, requiring that you prepend the local computer name as the domain of the account.

The following list shows two examples for a default instance and named instance called "Tabular" on a machine called "SQL01-WIN12", where the machine name is the local domain.

- SQL01-WIN12\SQL01-WIN12\$SQLServerMSASUser\$MSSQLSERVER
- SQL01-WIN12\SQL01-WIN12\$SQLServerMSASUser\$TABULAR

6. Repeat for **Adjust memory quotas for a process**, and optionally, for **Lock pages in memory** or **Increase scheduling priority**.

NOTE

Previous versions of Setup inadvertently added the Analysis Services service account to the **Performance Log Users** group. Although this defect has been fixed, existing installations might have this unnecessary group membership. Because the Analysis Services service account does not require membership in the **Performance Log Users** group, you can remove it from the group.

File System Permissions assigned to the Analysis Services service account

NOTE

See [Configure Windows Service Accounts and Permissions](#) for a list of permissions associated with each program folder.

See [Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#) for file permission information related to IIS configuration and Analysis Services.

All file system permissions required for server operations—including permissions needed for loading and unloading databases from a designated data folder—are assigned by SQL Server Setup during installation.

The permission holder on data files, program file executables, configuration files, log files, and temporary files is a local security group created by SQL Server Setup.

There is one security group created for each instance that you install. The security group is named after the instance — either **SQLServerMSASUser\$MSSQLSERVER** for the default instance, or **SQLServerMSASUser\$<servername>\$<instancename>** for a named instance. Setup provisions this security group with the file permissions required to perform server operations. If you check the security permissions on the `\MSAS13.MSSQLSERVER\OLAP\BIN` directory, you will see that the security group (not the service account or its per-service SID) is the permission holder on that directory.

The security group contains one member only: the per-service Security Identifier (SID) of the Analysis Services instance startup account. Setup adds the per-service SID to the local security group. The use of a local security group, with its SID membership, is a small but noticeable difference in how SQL Server Setup provisions Analysis Services, as compared to the Database Engine.

If you believe that file permissions are corrupted, follow these steps to verify the service is still correctly provisioned:

1. Use the Service Control command line tool (sc.exe) to obtain the SID of a default service instance.

```
SC showsid MSSqlServerOlapService
```

For a named instance (where the instance name is Tabular), use this syntax:

```
SC showsid MSOlap$Tabular
```

2. Use **Computer Manager | Local Users and Groups | Groups** to inspect the membership of the `SQLServerMSASUser$<servername>$<instancename>` security group.

The member SID should match the per-service SID from step 1.

3. Use **Windows Explorer | Program Files | Microsoft SQL Server | MSASxx.MSSQLServer | OLAP | bin** to verify folder Security properties are granted to the security group in step 2.

NOTE

Never remove or modify a SID. To restore a per-service SID that was inadvertently deleted, see <http://support.microsoft.com/kb/2620201>.

More about per-service SIDs

Every Windows account has an associated [SID](#), but services can also have SIDs, hence referred to as per-service SIDs. A per-service SID is created when the service instance is installed, as a unique, permanent fixture of the service. The per-service SID is a local, machine-level SID generated from the service name. On a default instance, its user friendly name is NT SERVICE\MSSQLServerOLAPService.

The benefit of a per-service SID is that it allows the more widely-visible logon account to be changed arbitrarily, without affecting file permissions. For example, suppose you installed two instances of Analysis Services, a default instance and named instance, both running under the same Windows user account. While the logon account is shared, each service instance will have a unique per-service SID. This SID is distinct from the SID of the logon account. The per-service SID is used for file permissions and Windows privileges. In contrast, the logon account SID is used for authentication and authorization scenarios — different SIDs, used for different purposes.

Because the SID is immutable, file system ACLs created during service installation can be used indefinitely, regardless of how often you change the service account. As an added security measure, ACLs that specify permissions via a SID ensure that program executables and data folders are accessed only by a single instance of a service, even if other services run under the same account.

Granting additional Analysis Services permissions for specific server operations

Analysis Services executes some tasks in the security context of the service account (or logon account) that is used to start Analysis Services, and executes other tasks in the security context of the user who is requesting the task.

The following table describes additional permissions required to support tasks executing as the service account.

SERVER OPERATION	WORK ITEM	JUSTIFICATION
Remote access to external relational data sources	Create a database login for the service account	Processing refers to data retrieval from an external data source (usually a relational database), which is subsequently loaded into an Analysis Services database. One of the credential options for retrieving external data is to use the service account. This credential option works only if you create a database login for the service account and grant read permissions on the source database. See Set Impersonation Options (SSAS - Multidimensional) for more information about how the service account option is used for this task. Similarly, if ROLAP is used as the storage mode, the same impersonation options are available. In this case, the account must also have write access to the source data to process the ROLAP partitions (that is, to store aggregations).

SERVER OPERATION	WORK ITEM	JUSTIFICATION
DirectQuery	Create a database login for the service account	<p>DirectQuery is a tabular feature used to query external datasets that are either too large to fit inside the tabular model or have other characteristics that make DirectQuery a better fit than the default in-memory storage option. One of the connection options available in DirectQuery mode is to use the service account. Once again, this option works only when the service account has a database login and read permissions on the target data source. See Set Impersonation Options (SSAS - Multidimensional) for more information about how the service account option is used for this task. Alternatively, the credentials of the current user can be used to retrieve data. In most cases this option entails a double-hop connection, so be sure to configure the service account for Kerberos constrained delegation so that the service account can delegate identities to a downstream server. For more information, see Configure Analysis Services for Kerberos constrained delegation.</p>
Remote access to other SSAS instances	Add the service account to Analysis Services database roles defined on the remote server	<p>Remote partitions and referencing linked objects on other remote Analysis Services instances are both system capabilities requiring permissions on a remote computer or device. When a person creates and populates remote partitions, or sets up a linked object, that operation runs in the security context of the current user. If you subsequently automate these operations, Analysis Services will access remote instances in the security context of its service account. In order to access linked objects on a remote instance of Analysis Services, the logon account must have permission to read the appropriate objects on the remote instance, such as Read access to certain dimensions. Similarly, using remote partitions requires that the service account have administrative rights on the remote instance. Such permissions are granted on the remote Analysis Services instance, using roles that associate permitted operations with a specific object. See Grant database permissions (Analysis Services) for instructions on how to grant Full Control permissions that allow processing and query operations. See Create and Manage a Remote Partition (Analysis Services) for more information about remote partitions.</p>



SERVER OPERATION	WORK ITEM	JUSTIFICATION
Writeback	Add the service account to Analysis Services database roles defined on the remote server	When enabled in client applications, writeback is a feature of multidimensional models that allows the creation of new data values during data analysis. If writeback is enabled within any dimension or cube, the Analysis Services service account must have write permissions to the writeback table in the source SQL Server relational database. If this table does not already exist and needs to be created, the Analysis Services service account must also have create table permissions within the designated SQL Server database.
Write to a query log table in a SQL Server relational database	Create a database login for the service account and assign write permissions on the query log table	You can enable query logging to collect usage data in a database table for subsequent analysis. The Analysis Services service account must have write permissions to the query log table in the designated SQL Server database. If this table does not already exist and needs to be created, the Analysis Services logon account must also have create table permissions within the designated SQL Server database. For more information, see Improve SQL Server Analysis Services Performance with the Usage Based Optimization Wizard (Blog) and Query Logging in Analysis Services (Blog) .

See Also

[Configure Windows Service Accounts and Permissions SQL Server Service Account and Per-Service SID \(Blog\)](#)
[SQL Server uses a service SID to provide service isolation \(KB Article\)](#)
[Access Token \(MSDN\)](#)
[Security Identifiers \(MSDN\)](#)
[Access Token \(Wikipedia\)](#)
[Access Control Lists \(Wikipedia\)](#)

Grant server admin rights to an Analysis Services instance

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Members of the Server administrator role within an instance of Analysis Services have unrestricted access to all Analysis Services objects and data in that instance. A user must be a member of the Server administrator role to perform any server-wide task, such as creating or processing a database, modifying server properties, or launching a trace (other than for processing events).

Role membership is established when Analysis Services is installed. The user running the Setup program can add him or herself to the role, or add another user. You must specify at least one administrator before Setup will allow you to continue.

By default, members of the local Administrators group are also granted administrative rights in Analysis Server. Although the local group is not explicitly granted membership in the Analysis Services server administrator role, local administrators can create databases, add users and permissions, and perform any other task allowed to system administrators. The implicit granting of administrator permissions is configurable. It is determined by the **BuiltinAdminsAreServerAdmins** server property, which is set to **true** by default. You can change this property in SQL Server Management Studio. For more information, see [Security Properties](#).

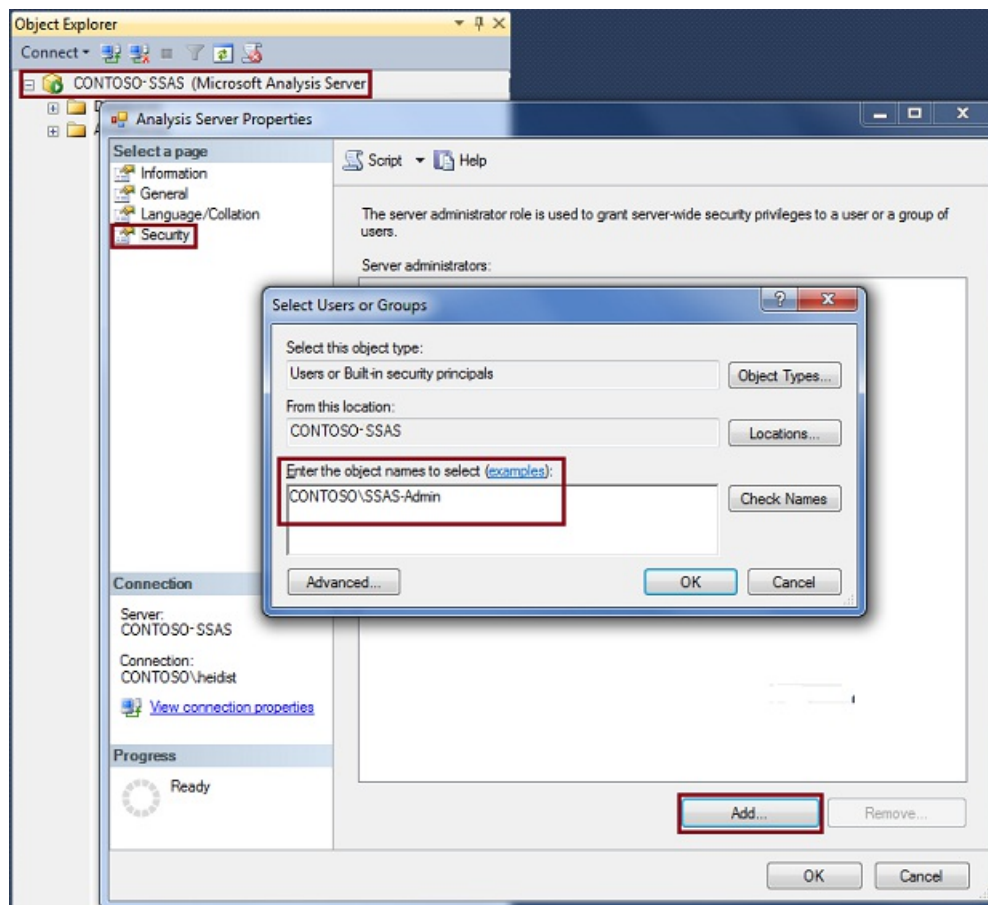
Post-installation, you can modify role membership to add any additional users who require full rights to the service. You can also manage server roles by using Analysis Management Objects (AMO). For more information, see [Developing with Analysis Management Objects \(AMO\)](#).

NOTE

Analysis Services provides a progression of increasingly granular roles for processing and querying at server, database, and object levels. See [Roles and Permissions \(Analysis Services\)](#) for instructions on how to use these roles.

Modify Server Role Membership

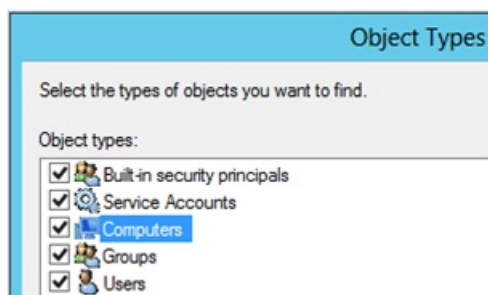
1. In SQL Server Management Studio, connect to the instance of Analysis Services, and then right-click the instance name in Object Explorer and then click **Properties**.
2. Click **Security** in the **Select a Page** pane, and then click **Add** at the bottom of the page to add one or more Windows users or groups to the server role.



Add computer accounts

You can also use SQL Server Management Studio to make a computer account a member of the Analysis Services administrators group.

1. In the **Select Users or Groups** dialog, click **Locations**.
2. Select the domain the computers that you want to add are a member of or select **Entire directory** and click **Ok**.
3. Click **Object Types**.
4. Select **Computers** and click **Ok**.



5. In the **Enter the object names to select** text box, type the name of the computer and click **Check Names** to verify the computer account is found in the current Locations. If the computer account is not found, verify the computer name and the correct domain the computer is a member of.

NT Service\SSASTelemetry account

NT Service/SSASTelemetry is a low-privileged machine account created during setup and used exclusively to run the Analysis Services implementation of the Customer Experience Improvement Program (CEIP) service. This service requires admin rights on the Analysis Services instance to run several discover commands. See [Customer Experience Improvement Program for SQL Server Data Tools](#) and [Microsoft SQL Server Privacy Statement](#) for

more information.


See Also

[Authorizing access to objects and operations \(Analysis Services\)](#)

[Security Roles \(Analysis Services - Multidimensional Data\)](#)

Features off by default (Analysis Services)

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

An instance of Analysis Services is designed to be secure by default. Therefore, features that might compromise security are disabled by default. The following features are installed in a disabled state and must specifically be enabled if you want to use them.

Feature List

To enable the following features, connect to Analysis Services using SQL Server Management Studio. Right-click the instance name and choose **Facets**. Alternatively, you can enable these features through server properties, as described in the next section.

- Ad Hoc Data Mining (OpenRowset) Queries
- Linked Objects (To)
- Linked Objects (From)
- Listen Only On Local Connections
- User Defined Functions

Server properties

Additional features that are off by default can be enabled through server properties. Connect to Analysis Services using SQL Server Management Studio. Right-click the instance name and choose **Properties**. Click **General**, and then click **Show Advanced** to display a larger property list.

- Ad Hoc Data Mining (OpenRowset) Queries
- Allow Session Mining Models (Data Mining)
- Linked Objects (To)
- Linked Objects (From)
- COM based user-defined functions
- Flight Recorder Trace Definitions (templates).
- Query logging
- Listen Only On Local Connections
- Binary XML
- Compression
- Group affinity. See [Thread Pool Properties](#) for details.

Register an Analysis Services Instance in a Server Group

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

If you have a large number of Analysis Services server instances, you can create server groups in Management Studio to make server administration easier. The purpose of a server group is to provide proximity among a group of related servers within the administrative workspace. For example, suppose you are tasked with managing ten separate instances of Analysis Services. Grouping them by server mode, up-time criteria, or by department or region would allow you to view and connect to instances that share the same characteristics more easily. You can also add descriptive information that helps you remember how the server is used.



Server groups can be created in a hierarchical structure. Local Server Group is the root node. It always contains instances of Analysis Services that run on the local computer. You can add remote servers to any group, including the local group.

After you create a server group, you must use the Registered Servers pane to view and connect to the member servers. The pane filters SQL Server instances by server type (Database Engine, Analysis Services, Reporting Services, and Integration Services). You click a server type to view the server groups created for it. To connect to a specific server within group, you double-click a server in the group.


The connection information that is defined for the server, including the server name, is persisted with server registration. You cannot modify the connection information, or use the registered name when connecting to the server using other tools.

Create a Server Group and Add Registered Servers

1. In Management Studio, click Registered Servers on the View menu to open the Registered Servers pane in the workspace. By default, a local Server Group is already created. All instances of Analysis Services that are running on the local server are members.
2. Right-click Local Server Group, select New Server Group, and give the group a name.
3. Right-click the server group and select New Server Registration. Enter the network name of a local or remote server, including the instance name if the server was installed as a named instance. Optionally, you can provide a registered server name that appears in Registered Servers. This name is used in Registered Servers only. You cannot use it to rename a server, nor can you use it in a connection string. A registered server name can be more descriptive than the actual server name or include other identifying characteristics that help you distinguish this server from other servers.

Determine the Server Mode of an Analysis Services Instance

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

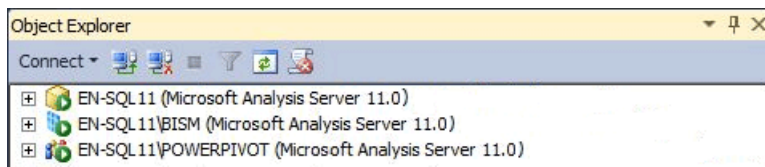
Analysis Services can be installed in one of three server modes: Multidimensional and Data Mining (default), Power Pivot for SharePoint, and Tabular. The server mode of an Analysis Services instance is determined during setup when you choose options for installing the server.

The server mode determines the type of solution that you create and deploy. If you did not install the server software and you want to know in which mode the server was installed, you can use the information in this topic to determine the mode. For more information about feature availability in a specific mode, see [Comparing Tabular and Multidimensional Solutions](#).

If you do not want to use the server mode that you installed, you must uninstall and then reinstall the software, choosing the mode that you prefer. Alternatively, you can install an additional instance of Analysis Services on the same computer so that you have multiple instances running different modes.

Server Icons in Object Explorer

The easiest way to determine server mode is to connect to the server in SQL Server Management Studio and note the icon next to the server name in Object Explorer. The following illustration shows three instances of Analysis Services deployed in Multidimensional, Tabular, and Power Pivot modes:



Viewing DeploymentMode Property in MSMDSRV.INI File

Alternatively, you can check the **DeploymentMode** property in the msmdsrv.ini file that is included in every Analysis Services instance. The value of this property identifies the server mode. Valid values are 0 (Multidimensional), 1 (SharePoint), or 2 (Tabular). You must be an Analysis Services administrator (that is, a member of the Server role) to open the msmdsrv.ini file. This file contains structured XML. You can use Notepad or another text editor to view the file.

Caution

Do not change the value of the **DeploymentMode** property. Changing the property manually after the server is installed is not supported.

About the DeploymentMode Property

DeploymentMode property determines the operational context of an Analysis Services server instance. This property is referred to as 'server mode' in dialog boxes, messages, and documentation. This property is initialized by Setup based on how you install Analysis Services. This property should be considered internal only, always using the value specified by Setup.

Valid values for this property include the following:

VALUE	DESCRIPTION
0	This is the default value. It specifies multidimensional mode, used to service multidimensional databases that use MOLAP, HOLAP, and ROLAP storage, as well as data mining models.
1	Specifies Analysis Services instances that were installed as part of a Power Pivot for SharePoint deployment. Do not change the deployment mode property of Analysis Services instance that is part of a Power Pivot for SharePoint installation. Power Pivot data will no longer run on the server if you change the mode.
2	Specifies Tabular mode used for hosting tabular model databases that use in-memory storage or DirectQuery storage.

Each mode is exclusive of the other. A server that is configured for tabular mode cannot run Analysis Services databases that contain cubes and dimensions. If the underlying computer hardware can support it, you can install multiple instances of Analysis Services on the same computer and configure each instance to use a different deployment mode. Remember that Analysis Services is a resource intensive application. Deploying multiple instances on the same system is recommended only for high-end servers.

See Also

[Install Analysis Services](#)

[Install Analysis Services in Multidimensional and Data Mining Mode](#)

[Power Pivot for SharePoint 2010 Installation](#)

[Connect to Analysis Services](#)

[Tabular Model Solutions](#)

[Multidimensional Model Solutions](#)

[Mining Models \(Analysis Services - Data Mining\)](#)

Rename an Analysis Services Instance

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

You can rename an existing instance of Microsoft Analysis Services by using the **Rename Instance** tool, installed with Management Studio (Web install).

IMPORTANT

While renaming the instance, the Analysis Services Instance Rename tool runs under elevated privileges, updating the Windows service name, security accounts, and registry entries associated with that instance. To ensure that these actions are performed, be sure to run this tool as a local system administrator.

The Analysis Services Instance Rename tool does not modify the program folder that was created for the original instance. Do not modify the program folder name to match the instance you are renaming. Changing a program folder name can prevent Setup from repairing or uninstalling the installation.

NOTE

The Analysis Services Instance Rename tool is not supported for use in a cluster environment.

To rename an instance of Analysis Services

1. Launch the **Instance Rename** tool, **asinstancereaname.exe**, from C:\Program Files (x86)\Microsoft SQL Server\130\Tools\Binn\ManagementStudio.
2. In the **Rename Instance** dialog box, in the **Instance to rename** list, select the instance that you want to rename.
3. In the **New instance name** box, enter the new name for the instance.
4. Verify that the user name and password are correct, and then click **Rename**.

The Analysis Services instance will be stopped and restarted as part of the name change.

Post-rename checklist

1. To resume access to databases that are running on the renamed instance, you will need to manually update the data connections in Excel or other client applications. Also check any predefined connections, such as Reporting Services shared data sources, Excel ODC files, or BI Semantic Model connection files that might reference the instance you just renamed. For more information, see [Connect to Analysis Services](#).
2. Update PowerShell scripts or AMO scripts that you routinely use to backup, synchronize, or process databases.
3. Update project properties for Analysis Services projects that you work with in SQL Server Data Tools. For tabular mode server instances, be sure to update the Workspace Server property on the model.bim file, as well as the Server property on the project.
4. Depending on how you specified the service account, you might need to update database logins or file permissions that grant data access rights to the service (for example, if you use the service account to process data or access linked objects on another server).

Updating a database login or file permissions will be necessary if you used a virtual account to provision the service. Virtual accounts are based on the instance name, so if you rename the instance, the virtual account is also updated at the same time. This means that any previous logins or permissions that you created for the previous instance are no longer valid.

The following example provides an illustration. Suppose you installed a tabular mode server as an instance named "Tabular" using the default virtual account, resulting in the following configuration:

- a. Instance name = <server>\TABULAR
- b. Service name = MSOLAP\$TABULAR
- c. Virtual account = NT Service\ MSOLAP\$TABULAR

Now suppose you rename the instance to "TAB2". As a result of the name change, your configuration would now look like the following:

- d. Instance name = <server>\TAB2
- e. Service name = MSOLAP\$TAB2
- f. Virtual account = NT Service\ MSOLAP\$TAB2

As you can see, database and file permissions that were previously granted to "NT Service\ MSOLAP\$TABULAR" are no longer valid. To ensure that tasks and operations performed by the service run as before, you would now need to grant new database and file permissions to "NT Service\ MSOLAP\$TAB2".

Connect to Analysis Services

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Use the information in this section to learn about connection string properties, client libraries used for connections, which authentication methods are supported by Analysis Services, and how to set up or clear connections before taking a server offline.

To learn about connecting to Azure Analysis Services, see [Connect to a server](#).

Analysis Services connections

Analysis Services uses TCP as the network protocol and XML for Analysis (XMLA) as a communication protocol. At the lowest level, all of the client libraries provided with Analysis Services implementing XMLA-over-TCP. Although it is possible to build applications based on raw XMLA, most applications and application developers use client libraries to take advantage of the object models and coding efficiencies that they provide. For client connections to Analysis Services, you can use IIS as an intermediary connection if you cannot use TCP across the stack. One advantage of using HTTP access via IIS is the ability to connect from applications that pass credentials on the connection string.

Any discussion involving connectivity typically includes authentication. In contrast with other SQL Server features, Analysis Services uses Windows credentials exclusively. You cannot use SQL Server database authentication, claims authentication, forms-based authentication, or digest on the backend connection to Analysis Services. More about authentication is provided in this section.

Connection Tasks

LINK	TASK DESCRIPTION
Connect from client applications (Analysis Services)	If you are new to Analysis Services, read this topic to get started with the tools and applications most often used with Analysis Services.
Connection String Properties (Analysis Services)	Analysis Services includes numerous server and database properties, allowing you to customize a connection for a specific application, independent of how the instance or database is configured.
Authentication methodologies supported by Analysis Services	This topic is a brief introduction to the authentication methods used by Analysis Services.
Configure Analysis Services for Kerberos constrained delegation	Many business intelligence solutions require impersonation to ensure that only authorized data is returned to each user. In this topic, learn the requirements for using impersonation. This topic also explains the steps for configuring Analysis Services for Kerberos constrained delegation.

LINK	TASK DESCRIPTION
SPN registration for an Analysis Services instance	Kerberos authentication requires a valid Service Principle Name (SPN) for services that impersonate or delegate user identities in multi-server solutions. Use the information in this topic to learn the construction and steps for SPN registration for Analysis Services.
Configure HTTP Access to Analysis Services on Internet Information Services (IIS) 8.0	Basic authentication or cross-domain boundaries are two important reasons for configuring Analysis Services for HTTP access.
Data providers used for Analysis Services connections	Analysis Services provides three client libraries for accessing server operations or Analysis Services data. This topic offers a brief introduction to ADOMD.NET, Analysis Services Management Objects (AMO), and the Analysis Services OLE DB provider (MSOLAP).
Disconnect Users and Sessions on Analysis Services Server	Clear existing connections and sessions before taking a server offline or conducting baseline performance tests.

See Also

[Post-install Configuration \(Analysis Services\)](#)
[Server Properties in Analysis Services](#)

Connect from client applications (Analysis Services)

5/16/2018 • 8 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

If you are new to Analysis Services, use the information in this topic to connect to an existing instance of Analysis Services using common tools and applications. This topic also explains how to connect under different user identities for testing purposes.

- [Connect using SQL Server Management Studio \(SSMS\)](#)
- [Connect using Excel](#)
- [Connect using SQL Server Data Tools](#)
- [Test connections](#)

Connection string reference documentation is provided separately. For more information, see [Connection String Properties \(Analysis Services\)](#).

Successful connections depend on a valid port configuration and appropriate user permissions. Click the following links to learn more about each requirement.

- [Configure the Windows Firewall to Allow Analysis Services Access](#)
- [Authorizing access to objects and operations \(Analysis Services\)](#)

Connect using SQL Server Management Studio (SSMS)

Connect to Analysis Services in SSMS to manage server instances and databases interactively. You can also run XMLA or MDX queries to perform administrative tasks or retrieve data. In contrast with other tools and applications that only load databases when a query is sent, SSMS loads all databases when you connect to the server, assuming you have permission to view the database. This means that if you have numerous tabular databases on the server, all are loaded into system memory when you connect using SSMS.

You can test permissions by running SSMS under a specific user identity and then connect to Analysis Services as that user.

Hold-down the Shift key and right-click the **SQL Server Management Studio** shortcut to access the **Run as different user** option.

1. Start SQL Server Management Studio. In the **Connect to Server** dialog box, select the Analysis Services server type.
2. In the Login tab, enter the server name by typing the name of the computer on which the server is running. You can specify the server using its network name or a fully-qualified domain name.

For a named instance, the server name must be specified in this format: servername\instance name. An example of this naming convention might be ADV-SRV062\Finance for a server that has a network name of ADV-SRV062, where Analysis Services was installed as a named instance entitled Finance.

For servers deployed in a failover cluster, connect using the network name of the SSAS cluster. This name is specified during SQL Server setup, as **SQL Server Network Name**. Note that if you installed SSAS as a named instance onto a Windows Server Failover Cluster (WSFC), you never add the instance name on the connection. This practice is unique to SSAS; in contrast, a named instance of a clustered relational database

engine does include the instance name. For example, if you installed both SSAS and the database engine as named instance (Contoso-Accounting) with a SQL Server Network Name of SQL-CLU, you would connect to SSAS using "SQL-CLU" and to the database engine as "SQL-CLU\Contoso-Accounting". See [How to Cluster SQL Server Analysis Services](#) for more information and examples.

For servers deployed in a network load balanced cluster, connect using the virtual server name of the NLB.

3. Authentication is always Windows authentication, and the user identity is always the Windows user who is connecting via Management Studio.

In order for the connection to succeed, you must have permission to access the server or a database on the server. Most tasks that you want to perform in Management Studio require administrative permissions. Ensure that the account you are connecting with is a member of the Server Administrator role. For more information, see [Grant server admin rights to an Analysis Services instance](#).

4. Click **Connection Properties** to specify a particular database, set timeout values, or encryption options. Optional connection information includes connection properties used for the current connection only.
5. Click **Additional Connection Parameters** tab to set connection properties not available in the Connect to Server dialog box. For example, you might type `Roles=Reader` in the text box.

Connecting through a role with less permission lets you test database behaviors when that role is in effect.

```
Provider=MSOLAP; Data Source=SERVERNAME; Initial Catalog=AdventureWorks2012; Roles=READER
```

Connect using Excel

Microsoft Excel is often used for analyzing business data. As part of an Excel installation, Office installs the Analysis Services OLE DB provider (MSOLAP DLL), ADOMD.NET, and other data providers so that you can more readily use the data on your network servers. If you are using a newer version of Analysis Services with an older version of Excel, you most likely need to install newer data providers on each workstation that connects to Analysis Services. See [Data providers used for Analysis Services connections](#) for more information.

When you set up a connection to an Analysis Services cube or tabular model database, Excel saves the connection information in .odc file for future use. The connection is made in security context of the current Windows user. The user account must have read permissions on the database in order for the connection to succeed.

When using Analysis Services data in an Excel workbook, connections are held for the duration of a query request. This is why you are likely to see lots of connections for each session, held for very short periods of time, when monitoring a query workload from Excel.

You can test permissions by starting Excel under a specific user identity.

Hold-down the Shift key and right-click the **Excel** shortcut to access the **Run as different user** option.

1. On the Data tab in Excel, click **From Other Sources**, and then click **From Analysis Services**. Enter the server name, and then select a cube or perspective to query.

For servers deployed in a load-balanced cluster, use the virtual server name assigned to the cluster.

2. When setting up a connection in Excel, on the last page of the Data Connection Wizard, you can specify authentication settings for Excel Services. These settings are used to set properties on the workbook should you upload it to a SharePoint server that has Excel Services. The settings are used in data refresh operations. Options include **Windows Authentication**, **Secure Store Service (SSS)**, and **None**.

Avoid using **None**. Analysis Services does not let you specify a user name and password on the connection string unless you are connecting to a server that has been configured for HTTP access. Similarly, do not use

SSS unless you already know that the SSS target application ID is mapped to a set of Windows user credentials that have user access to the Analysis Services databases. For most scenarios, using the default option of Windows authentication is the best choice for an Analysis Services connection from Excel.

For more information, see [Connect to or import data from SQL Server Analysis Services](#).

Connect using SQL Server Data Tools

SQL Server Data Tools is used for building BI solutions, including Analysis Services models, Reporting Services reports, and SSIS packages. When building reports or packages, you might need to specify a connection to Analysis Services.

The following links explain how to connect to Analysis Services from a Report Server project or an Integration Services project:

- [Analysis Services Connection Type for MDX \(SSRS\)](#)
- [Analysis Services Connection Manager](#)

NOTE

When using SQL Server Data Tools to work on an existing Analysis Services project, remember that you can connect offline using a local or version controlled project, or connect in online mode to update Analysis Services objects while the database is running. For more information, see [Connect in Online Mode to an Analysis Services Database](#). More commonly, connections from SQL Server Data Tools are in project mode, where changes are deployed to the database only when you explicitly deploy the project.

Test connections

You can use SQL Server Profiler to monitor connections to Analysis Services. The Audit Login and Audit Logout events provide evidence of a connection. The identity column indicates the security context under which the connection is made.

1. Start **SQL Server Profiler** on the Analysis Services instance and then start a new trace.
2. In Events Selection, verify that **Audit Login** and **Audit Logout** are checked in the Security Audit section.
3. Connect to Analysis Services via an application service (such as SharePoint or Reporting Services) from a remote client computer. The Audit Login event will show the identity of the user connecting to Analysis Services.

Connection errors are often traced to an incomplete or invalid server configuration. Always check server configuration first:

- Ping the server from a remote computer to ensure it allows remote connections.
- **Firewall rules on the server allow inbound connections from clients in the same domain**

With the exception of Power Pivot for SharePoint, all connections to a remote server require that you have configured the firewall to allow access to the port that Analysis Services is listening on. If you are getting connection errors, verify that the port is accessible and that user permissions are granted to the appropriate databases.

To test, use Excel or SSMS to on a remote computer, specifying the IP address and port used by the Analysis Services instance. If you can connection, the firewall rules are valid for the instance and the instance allows remote connections.

Also, when using TCP/IP for the connection protocol, remember that Analysis Services requires client

connections originate from the same domain or a trusted domain. If connections flow across security boundaries, you will most likely need to configure HTTP access. For more information, see [Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#).

- Can you connect using some tools but not others? The problem might be the wrong version of a client library. You can get client libraries from the SQL Server Feature Pack download page.

Resources that can help you resolve connection failures include the following:

[Resolving Common Connectivity Issues in SQL Server 2005 Analysis Services Connectivity Scenarios](#). This document is a few years old, but the information and methodologies still apply.

See Also

[Connect to Analysis Services](#)

[Authentication methodologies supported by Analysis Services](#)

[Impersonation](#)

[Create a Data Source \(SSAS Multidimensional\)](#)

Connection String Properties (Analysis Services)

6/8/2018 • 17 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

This topic describes connection string properties you might set in one of the designer or administration tools, or see in connection strings built by client applications that connect to and query Analysis Services data. As such, it covers just a subset of the available properties. The complete list includes numerous server and database properties, allowing you to customize a connection for a specific application, independent of how the instance or database is configured on the server.

Developers who build custom connection strings in application code should review the API documentation for ADOMD.NET client to view a more detailed list: [ConnectionString](#)

The properties described in this topic are used by the Analysis Services client libraries, ADOMD.NET, AMO, and the OLE DB provider for Analysis Services. The majority of connection string properties can be used with all three client libraries. Exceptions are called out in the description.

NOTE

When setting properties, if you inadvertently set the same property twice, the last one in the connection string is used.

For more information about how to specify an Analysis Services connection in existing Microsoft applications, see [Connect from client applications \(Analysis Services\)](#).

Connection parameters in common use

The following table describes those properties most often used when building a connection string.

PROPERTY	DESCRIPTION	EXAMPLE
Data Source or DataSource	<p>Specifies the server instance. This property is required for all connections. Valid values include the network name or IP address of the server, local or localhost for local connections, a URL if the server is configured for HTTP or HTTPS access, or the name of a local cube (.cub) file.</p> <p>Valid value for Azure Analysis Services, <code><protocol>://<region>/<servername></code> where protocol is string asazure, region is the Uri where the server was created (for example, westus.azure.windows.net) and servername is the name of your unique server within the region.</p>	<pre>Data source=asazure://westus.azure.windows.net/my</pre> <p><code>Data source=AW-SRV01</code> for the default instance and port (TCP 2383).</p> <p><code>Data source=AW-SRV01\$Finance:8081</code> for a named instance (\$Finance) and fixed port.</p> <pre>Data source=AW-SRV01.corp.Adventure-Works.com</pre> <p>for a fully qualified domain name, assuming the default instance and port.</p> <pre>Data source=172.16.254.1</pre> for an IP address of the server, bypassing DNS server lookup, useful for troubleshooting connection problems.
Initial Catalog or Catalog	<p>Specifies the name of the Analysis Services database to connect to. The database must be deployed on Analysis Services, and you must have permission to connect to it. This property is optional for AMO connections, but required for ADOMD.NET.</p>	<pre>Initial catalog=AdventureWorks2016</pre>

PROPERTY	DESCRIPTION	EXAMPLE
Provider	<p>Valid values include MSOLAP.<version>, where <version> is either 4, 5, 6 or 7.</p> <ul style="list-style-type: none"> - MSOLAP4 released in SQL Server 2008 and again SQL Server 2008 R2 (filename is msolap100.dll for SQL Server 2008 and 2008 R2) - MSOLAP5 released in SQL Server 2012 (filename is msolap110.dll) - MSOLAP6 released in SQL Server 2014 (filename is msolap1200.dll) - MSOLAP7 released in SQL Server 2016 (filename is msolap130.dll) <p>This property is optional. By default, the client libraries read the current version of the OLE DB provider from the registry. You only need to set this property if you require a specific version of the data provider, for example to connect to a SQL Server 2012 instance.</p> <p>MSOLAP4 was released in both SQL Server 2008 and SQL Server 2008 R2. The 2008 R2 version supports Power Pivot workbooks and sometimes needs to be installed manually on SharePoint servers. To distinguish between these versions, you must check the build number in the file properties of the provider: Go to Program files\Microsoft Analysis Services\AS OLEDB\10. Right-click msolap110.dll and select Properties. Click Details. View the file version information. The version should include 10.50.<buildnumber> for SQL Server 2008 R2. For more information, see Install the Analysis Services OLE DB Provider on SharePoint Servers and Data providers used for Analysis Services connections.</p>	<p><code>Provider=MSOLAP.7</code> is used for connections that require the SQL Server 2016 version of the OLE DB provider for Analysis Services.</p>
Cube	<p>Cube name or perspective name. A database can contain multiple cubes and perspectives. When multiple targets are possible, include the cube or perspective name on the connection string.</p>	<p><code>Cube=SalesPerspective</code> shows that you can use the Cube connection string property to specify either the name of a cube or the name of a perspective.</p>

Authentication and Security

This section includes connection string properties related to authentication and encryption. Analysis Services uses Windows Authentication only, but you can set properties on the connection string to pass in a specific user name and password.

Properties are listed in alphabetical order.

PROPERTY	DESCRIPTION
----------	-------------

PROPERTY	DESCRIPTION
EffectiveUserName	<p>Use when an end user identity must be impersonated on the server. Specify the account in a domain\user format. To use this property, the caller must have administrative permissions in Analysis Services. For more information about using this property in an Excel workbook from SharePoint, see Use Analysis Services EffectiveUserName in SharePoint Server 2013. For an illustration of how this property is used with Reporting Services, see Using EffectiveUserName To Impersonate in SSAS.</p> <p>EffectiveUserName is used in a Power Pivot for SharePoint installation to capture usage information. The user identity is provided to the server so that events or errors that include user identity can be recorded in the log files. In the case of Power Pivot, it is not used for authorization purposes.</p>
Encrypt Password	Specifies whether a local password is to be used to encrypt local cubes. Valid values are True or False. The default is False.
Encryption Password	The password used to decrypt an encrypted local cube. Default value is empty. This value must be explicitly set by the user.
Impersonation Level	<p>Indicates the level of impersonation that the server is allowed to use when impersonating the client. Valid values include:</p> <ul style="list-style-type: none"> - Anonymous. The client is anonymous to the server. The server process cannot obtain information about the client, nor can the client be impersonated. - Identify. The server process can get the client identity. The server can impersonate the client identity for authorization purposes but cannot access system objects as the client. - Impersonate. This is the default value. The client identity can be impersonated, but only when the connection is established, and not on every call. - Delegate. The server process can impersonate the client security context while acting on behalf of the client. The server process can also make outgoing calls to other servers while acting on behalf of the client.
Integrated Security	<p>The Windows identity of the caller is used to connect to Analysis Services. Valid values are blank, SSPI, and BASIC.</p> <p>Integrated Security=SSPI is the default value for TCP connections, allowing NTLM, Kerberos, or Anonymous authentication. Blank is the default value for HTTP connections.</p> <p>When using SSPI, ProtectionLevel must be set to one of the following: Connect, PktIntegrity, PktPrivacy.</p>
Persist Encrypted	Set this property when the client application requires the data source object to persist sensitive authentication information, such as a password, in encrypted form. By default, authentication information is not persisted.
Persist Security Info	Valid values are True and False. When set to True, security information, such as the user identity or password previously specified on the connection string, can be obtained from the connection after the connection is made. The default value is False.

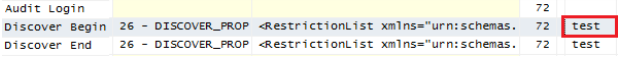
PROPERTY	DESCRIPTION
Protection Level	<p>Determines the security level used on the connection. Valid values are:</p> <ul style="list-style-type: none"> - None. Unauthenticated or anonymous connections. Performs no authentication on data sent to the server. - Connect. Authenticated connections. Authenticates only when the client establishes a relationship with a server. - Pkt Integrity. Encrypted connections. Verifies that all data is received from the client and that it has not been changed in transit. - Pkt Privacy. Signed encryption, supported only for XMLA. Verifies that all data is received from the client, that it has not been changed in transit, and protects the privacy of the data by encrypting it. <p>For more information, see Establishing Secure Connections in ADOMD.NET</p>
Roles	<p>Specify a comma-delimited list of predefined roles to connect to a server or database using permissions conveyed by that role. If this property is omitted, all roles are used, and the effective permissions are the combination of all roles. Setting the property to an empty value (for example, Roles=' ') the client connection has no role membership.</p> <p>An administrator using this property connects using the permissions conveyed by the role. Some commands might fail if the role does not provide sufficient permission.</p>
SSPI	<p>Explicitly specifies which security package to use for client authentication when Integrated Security is set to SSPI. SSPI supports multiple packages, but you can use this property to specify a particular package. Valid values are Negotiate, Kerberos, NTLM, and Anonymous User. If this property is not set, all packages will be available to the connection.</p>
Use Encryption for Data	<p>Encrypts data transmissions. Value values are True and False.</p>
User ID=...; Password=	<p>User ID and Password are used together. Analysis Services impersonates the user identity specified through these credentials. On an Analysis Services connection, putting credentials on the command line is used only when the server is configured for HTTP access, and you specified Basic authentication instead of integrated security on the IIS virtual directory. When connecting directly to the server, UserID and Password connection string params are ignored and the connection is made using the context of the logged on user.</p> <p>The user name and password must be the credentials of a Windows identity, either a local or a domain user account. Notice that User ID has an embedded space. Other aliases for this property include UserName (no space), and UID. Alias for Password is PWD.</p>

Special-purpose parameters

This section describes the remainder of the connection string parameters. These are used to ensure specific connection behaviors required by an application.

Properties are listed in alphabetical order.

PROPERTY	DESCRIPTION
----------	-------------

PROPERTY	DESCRIPTION
Application Name	<p>Sets the name of the application associated with the connection. This value can be useful when monitoring tracing events, especially when you have several applications accessing the same databases. For example, adding Application Name='test' to a connection string causes 'test' to appear in a SQL Server Profiler trace, as shown in the following screenshot:</p>  <p>Aliases for this property include sspropinitAppName, AppName. For more information, see Use Application Name parameter when connecting to SQL Server.</p>
AutoSyncPeriod	<p>Sets the frequency (in milliseconds) of client and server cache synchronization. ADOMD.NET provides client caching for frequently used objects that have minimal memory overhead. This helps reduce the number of round trips to the server. The default is 10000 milliseconds (or 10 seconds). When set to null or 0, automatic synchronization is turned off.</p>
Character Encoding	<p>Defines how characters are encoded on the request. Valid values are Default or UTF-8 (these are equivalent), and UTF-16</p>
CommitTimeout	<p>An XMLA property. Determines how long, in milliseconds, the commit phase of a currently running command waits before rolling back. When greater than 0, overrides the value of the corresponding CommitTimeout property in the server configuration.</p>
CompareCaseSensitiveStringFlags	<p>Adjusts case-sensitive string comparisons for a specified locale. For more information about setting this property, see CompareCaseSensitiveStringFlags Property.</p>
Compression Level	<p>If TransportCompression is XPRESS, you can set the compression level to control how much compression is used. Valid values are 0 through 9, with 0 having least compression, and 9 having the most compression. Increased compression slows performance. The default value is 0.</p>
Connect Timeout	<p>Determines the maximum amount of time (in seconds) the client attempts a connection before timing out. If a connection does not succeed within this period, the client quits trying to connect and generates an error.</p>
DbpropMsmdRequestMemoryLimit	<p>This property overrides the Memory\QueryMemoryLimit server property value for a connection. Specified in kilobytes.</p>
MDX Compatibility	<p>The purpose of this property is to ensure a consistent set of MDX behaviors for applications that issue MDX queries. Excel, which uses MDX queries to populate and calculate a PivotTable connected to Analysis Services, sets this property to 1, to ensure that placeholder members in ragged hierarchies are visible in a PivotTable. Valid values include 0, 1, 2.</p> <p>0 and 1 expose placeholder members; 2 does not. If this is empty, 0 is assumed.</p>
MDX Missing Member Mode=Error	<p>Indicates whether missing members are ignored in MDX statements. Valid values are Default, Error, and Ignore. Default uses a server-defined value. Error generates an error when a member does not exist. Ignore specifies that missing values should be ignored.</p>

PROPERTY	DESCRIPTION
Optimize Response	<p>A bitmask indicating which of the following query response optimizations are enabled.</p> <ul style="list-style-type: none"> - 0x01 Use the NormalTupleSet (this is the default) - 0x02 Use when slicers are empty
Packet Size	A network packet size (in bytes) between 512 and 32,767. The default network packet size is 4096.
Protocol Format	<p>Sets the format of the XML sent to the server. Valid values are Default, XML, or Binary. The protocol is XMLA. You can specify that the XML be sent in compressed form (this is the default), as raw XML, or in a binary format. Binary format encodes XML elements and attributes, making them smaller. Compression is a proprietary format that further reduces the size of requests and responses. Compression and binary formats are used to speed up data transfer requests and responses.</p> <p>You must use a client library on the connection if using binary or compressed format. OLE DB provider can format requests and responses in binary or compressed format. AMO and ADOMD.NET format the requests as Text, but accept responses in binary or compressed format.</p> <p>This connection string property is equivalent to the EnableBinaryXML and EnableCompression server configuration settings.</p>
Real Time Olap	Set this property to bypass caching, causing all partitions to actively listen for query notifications. By default, this property is not set.
Safety Options	Sets the safety level for user-defined functions and actions. Valid values are 0, 1, 2. In an Excel connection this property is Safety Options=2. Details about this option can be found in ConnectionString .
SQLQueryMode	Specifies whether SQL queries include calculations. Valid values are Data, Calculated, IncludeEmpty. Data means that no calculations are allowed. Calculated allows calculations. IncludeEmpty allows calculations and empty rows to be returned in the query result.
Timeout	Specifies how long (in seconds) the client library waits for a command to complete before generating an error.
Transport Compression	Defines how client and server communications are compressed, when compression is specified via the Protocol Format property. Valid values are Default, None, Compressed and gzip . Default is no compression for TCP, or gzip for HTTP. None indicates that no compression is used. Compressed uses XPRESS compression (SQL Server 2008 and later). gzip is only valid for HTTP connections, where the HTTP request includes Accept-Encoding=gzip.
UseExistingFile	Used when connecting to a local cube. This property specifies whether the local cube is overwritten. Valid values are True or False. If set to True, the cube file must exist. The existing file will be the target of the connection. If set to False, the cube file is overwritten.

PROPERTY	DESCRIPTION
VisualMode	<p>Set this property to control how members are aggregated when dimension security is applied.</p> <p>For cube data that everyone is allowed to see, aggregating all of the members makes sense because all of the values that contribute to the total are visible. However, if you filter or restrict dimensions based on user identity, showing a total based on all the members (combining both restricted and allowed values into a single total) might be confusing or show more information than should be revealed.</p> <p>To specify how members are aggregated when dimension security is applied, you can set this property to True to use only allowed values in the aggregation, or False to exclude restricted values from the total.</p> <p>When set on the connection string, this value applies to the cube or perspective level. Within a model, you can control visual totals at a more granular level.</p> <p>Valid values are 0, 1, and 2.</p> <ul style="list-style-type: none"> - 0 is the default. Currently, the default behavior is equivalent to 2, where aggregations include values that are hidden from the user. - 1 excludes hidden values from the total. This is the default for Excel. - 2 includes hidden values in the total. This is the default value on the server. <p>Aliases for this property include Visual Total or Default MDX Visual Mode.</p>

Reserved for future use

The following properties are allowed on a connection string, but are not operational in current releases of Analysis Services.

- Authenticated User
- Cache Authentication
- Cache Mode (Use of this property was investigated in earlier releases. Although you might find blog posts recommending its usage, you should avoid setting this property unless instructed by Microsoft Support).
- Cache Policy
- Cache Ratio
- Cache Ratio2
- Dynamic Debug Limit
- Debug Mode
- Mode
- SQLCompatibility
- Use Formula Cache

Example connection strings

This section shows the connection string that you'll most likely use when setting up an Analysis Services connection in commonly used applications.

Generic connection string

You might use a connection string like this one if you are configuring a connection from Reporting Services.

```
Data source=<servername>; initial catalog=<databasename>
```

Connection string in Excel

The default ADOMD.NET connection string in Excel specifies the data provider, server, database name, Windows integrated security. The MDX Compatibility level is always set to 1. Although you can change the value for the current session, Excel will reset MDX Compatibility to 1 when the file is next opened.

```
Provider=MSOLAP.5;Integrated Security=SSPI;Persist Security Info=True;Initial Catalog=Adventure Works DW 2008R2;Data Source=AW-SRV01;MDX Compatibility=1;Safety Options=2;MDX Missing Member Mode=Error
```

For more information, see [Data Connections, Data Sources, and Connection Strings \(Report Builder and SSRS\)](#) and [Data Authentication for Excel Services in SharePoint Server 2013](#).

Connection string formats used in Analysis Services

This section lists all of the connection string formats supported by Analysis Services. With the exception of connections to Power Pivot databases, you can specify these connections strings in applications that connect to Analysis Services.

Native (or direct) connections to the server

`Data Source=server[:port][\instance]` where “port” and “\instance” are optional. For example, specifying “Data Source=server1” opens a connection to the default instance (and default port 2383) on a server named “server1”.

“Data Source=server1:port1” will open a connection to an Analysis Services instance running on port “port1” on “server1”.

“Data Source=server1\instance1” will open a connection to SQL Browser (on its default port 2382), resolve the port for the named instance “instance1”, then open the connection to that Analysis Services port.

“Data Source=server1:port1\instance1” will open a connection to SQL Browser on “port1”, resolve the port for the “instance1” named instance, then open the connection to that Analysis Services port.

Local cube connections (.cub files)

`Data Source=<path>`, for example “Data Source=c:\temp\a.cub”

Http(s) connections to msmdpump.dll

`Data Source=<URL>`, where the URL is the HTTP or HTTPS address to the virtual IIS folder that contains the msmdpump.dll. For more information, see [Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#).

Http(s) connections to Power Pivot workbooks (.xlsx, .xlsb or .xlsm files)

`Data Source=<URL>`, where the URL is the SharePoint path to a Power Pivot workbook that has been published to a SharePoint library. For example, `Data Source=http://localhost/Shared Documents/Sales.xlsx`.

Http(s) connections to BI Semantic Model Connection files

`Data Source=<URL>` where the URL is the SharePoint path to the .bism file. For example, `Data Source=http://localhost/Shared Documents/Sales.bism`.

Embedded Power Pivot connections

`Data Source=$Embedded$` where \$embedded\$ is a moniker that refers to an embedded Power Pivot data model inside the workbook. This connection string is created and managed internally. Do not modify it. Embedded connection strings are resolved by the Power Pivot for Excel add-in on client workstations, or by Power Pivot for SharePoint instances in a SharePoint farm.

Local server context in Analysis Services stored procedures

`Data Source=*`, where * resolves to the local instance.

Encrypting Connection Strings

Analysis Services uses its own encryption keys to encrypt connection strings. It does not generate a self-signed certificate.

Analysis Services encrypts and stores the connection strings it uses to connect to each of its data sources. If the connection to a data source requires a user name and password, you can have Analysis Services store the name and password with the

connection string, or prompt you for the name and password each time a connection to the data source is required. Having Analysis Services prompt you for user information means that this information does not have to be stored and encrypted. However, if you store this information in the connection string, this information does need to be encrypted and secured.

To encrypt and secure the connection string information, Analysis Services uses the Data Protection API.

Analysis Services uses a separate encryption key to encrypt connection string information for each Analysis Services database. Analysis Services creates this key when you create a database, and encrypts connection string information based on the Analysis Services startup account. When Analysis Services starts, the encrypted key for each database is read, decrypted, and stored. Analysis Services then uses the appropriate decrypted key to decrypt the data source connection string information when Analysis Services needs to connect to a data source.

See Also

[Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#)



[Configure Analysis Services for Kerberos constrained delegation](#)

[Data providers used for Analysis Services connections](#)

[Connect to Analysis Services](#)

Authentication methodologies supported by Analysis Services

5/16/2018 • 5 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Connections from a client application to an Analysis Services instance require Windows authentication (integrated). You can provide a Windows user identity using any of the following methods:

- NTLM
- Kerberos (see [Configure Analysis Services for Kerberos constrained delegation](#))
- EffectiveUserName on the connection string
- Basic or Anonymous (requires configuration for HTTP access)
- Stored credentials

Notice that Claims authentication is not supported. You cannot use a Windows Claims token to access Analysis Services. The Analysis Services client libraries only work with Windows security principles. If your BI solution includes claims identities, you will need Windows identity shadow accounts for each user, or use stored credentials to access Analysis Services data.

For more information about BI and Analysis Services authentication flows, see [Microsoft BI Authentication and Identity Delegation](#).

Understanding your authentication alternatives

Connecting to an Analysis Services database requires a Windows user or group identity and associated permissions. The identity might be a general purpose login used by anyone who needs to view a report, but a more likely scenario includes the identity of individual users.

Often, a tabular or multidimensional model will have different levels of data access, by object or within the data itself, based on who is making the request. To meet this requirement, you can use NTLM, Kerberos, EffectiveUserName, or Basic authentication. All of these techniques offer an approach for passing in different user identities with each connection. However, most of these choices are subject to a single hop limitation. Only Kerberos with delegation allows the original user identity to flow across multiple computer connections to a backend data store on a remote server.

NTLM

For connections that specify `SSPI=Negotiate`, NTLM is the backup authentication subsystem used when a Kerberos domain controller is not available. Under NTLM, any user or client application can access a server resource as long as the request is a direct connection from a client to the server, the person requesting the connection has permission to the resource, and the client and server computers are in the same domain.

In multi-tier solutions, the single hop restriction of NTLM can be a major constraint. The user identity making the request can be impersonated on exactly one remote server, but travels no further. If the current operation requires services running on multiple computers, you will need to configure Kerberos constrained delegation to reuse the security token on backend servers. Alternatively, you can use stored credentials or Basic authentication to pass in new identity information over a single hop connection.

Kerberos Authentication and Kerberos Constrained Delegation

Kerberos authentication is the basis of Windows integrated security in Active Directory domains. As with NTLM, impersonation under Kerberos is limited to a single hop unless you enable delegation.

To support multi-hop connections, Kerberos provides both constrained and unconstrained delegation, but for most scenarios, constrained delegation is considered a security best practice. Constrained delegation allows a service to pass the security token of the user identity to a designated down-level service on a remote computer. For multi-tier applications, delegating a user identity from a middle tier application server to a backend database such as Analysis Services is a common requirement. For example, a tabular or multidimensional model that returns different data based on user identity would require identity delegation from a middle tier service to avoid having the user re-enter credentials, or getting security credentials some other way.

Constrained delegation requires additional configuration in Active Directory, where services on both the sending and receiving end of the request are explicitly authorized for delegation. Although there are configuration costs up front, once the service is configured, password updates are managed independently in Active Directory. You do not need to update stored account information in applications, as you would if using the stored credentials option described further on.

For more information about configuring Analysis Services for constrained delegation, see [Configure Analysis Services for Kerberos constrained delegation](#).

NOTE

Windows Server 2012 supports constrained delegation across domains. In contrast, configuring Kerberos constrained delegation in domains at lower functional levels, such as Windows Server 2008 or 2008 R2, require both client and server computers to be members of the same domain.

EffectiveUserName

EffectiveUserName is a connection string property used for passing identity information to Analysis Services. Power Pivot for SharePoint uses it to record user activity in the usage logs. Excel Services and PerformancePoint Services can use it to retrieve data used by workbooks or dashboards in SharePoint. It can also be used in custom applications or scripts that perform operations on an Analysis Services instance.

For more information about using EffectiveUserName in SharePoint, see [Use Analysis Services EffectiveUserName in SharePoint Server 2010](#).

Basic Authentication and Anonymous User

Basic authentication provides yet a fourth alternative for connecting to a backend server as a specific user. Using Basic authentication, the Windows user name and password are passed on the connection string, introducing additional wire encryption requirements to ensure sensitive information is protected while in transit. An important advantage to using Basic authentication is that the authentication request can cross domain boundaries.

For Anonymous authentication, you can set the anonymous user identity to a specific Windows user account (IUSR_GUEST by default) or an application pool identity. The anonymous user account will be used on the Analysis Services connection, and must have data access permissions on the Analysis Services instance. When you use this approach, only the user identity associated with the Anonymous account is used on the connection. If your application requires additional identity management, you will need to choose one of the other approaches, or supplement with an identity management solution that you provide.

Basic and Anonymous are available only when you configure Analysis Services for HTTP access, using IIS and the msmdpump.dll to establish the connection. For more information, see [Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#).

Stored Credentials

Most middle tier application services include functionality for storing a user name and password subsequently used to retrieve data from a down-level data store, such as Analysis Services or the SQL Server relational engine. As such, stored credentials provide a fifth alternative for retrieving data. Limitations with this approach include maintenance overhead associated with keeping user names and passwords up to date, and the use of a single identity on the connection. If your solution requires the identity of the original caller, then stored credentials would not be a viable alternative.

For more information about stored credentials, see [Create, Modify, and Delete Shared Data Sources \(SSRS\)](#) and [Use Excel Services with Secure Store Service in SharePoint Server 2013](#).

See Also

[Using Impersonation with Transport Security](#)

[Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#)

[Configure Analysis Services for Kerberos constrained delegation](#)

[SPN registration for an Analysis Services instance](#)

[Connect to Analysis Services](#)

Configure Analysis Services for Kerberos constrained delegation

5/16/2018 • 8 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

When configuring Analysis Services for Kerberos authentication, you are most likely interested in achieving one or both of the following outcomes: having Analysis Services impersonate a user identity when querying data; or having Analysis Services delegate a user identity to a down-level service. Each scenario calls for slightly different configuration requirements. Both scenarios require verification to ensure configuration was done properly.

TIP

Microsoft Kerberos Configuration Manager for SQL Server is a diagnostic tool that helps troubleshoot Kerberos related connectivity issues with SQL Server. For more information, see [Microsoft Kerberos Configuration Manager for SQL Server](#).

This topic contains the following sections:

- [Allow Analysis Services to impersonate a user identity](#)
- [Configure Analysis Services for trusted delegation](#)
- [Test for impersonated or delegated identity](#)

NOTE

Delegation is not required if the connection to Analysis Services is a single hop, or your solution uses stored credentials provided by SharePoint Secure Store Service or Reporting Services. If all connections are direct connections from Excel to an Analysis Services database, or based on stored credentials, you can use Kerberos (or NTLM) without having to configure constrained delegation.

Kerberos constrained delegation is required if the user identity has to flow over multiple computer connections (known as "double-hop"). When Analysis Services data access is contingent upon user identity, and the connection request is from a delegating service, use the checklist in the next section to ensure that Analysis Services is able to impersonate the original caller. For more information about Analysis Services authentication flows, see [Microsoft BI Authentication and Identity Delegation](#).

As a security best practice, Microsoft always recommends constrained delegation over unconstrained delegation. Unconstrained delegation is a major security risk because it allows the service identity to impersonate another user on *any* downstream computer, service, or application (as opposed to just those services explicitly defined via constrained delegation).

Allow Analysis Services to impersonate a user identity

To allow up-level services such as Reporting Services, IIS, or SharePoint to impersonate a user identity on Analysis Services, you must configure Kerberos constrained delegation for those services. In this scenario, Analysis Services impersonates the current user using the identity provided by the delegating service, returning results based on role membership of that user identity.

TASK	DESCRIPTION
<p>Step 1: Verify that accounts are suitable for delegation</p>	<p>Ensure that the accounts used to run the services have the correct properties in Active Directory. Service accounts in Active Directory must not be marked as sensitive accounts, or specifically excluded from delegation scenarios. For more information, see Understanding User Accounts.</p> <p>Note: Generally, all accounts and servers must belong to the same Active Directory domain or to trusted domains in the same forest. However, because Windows Server 2012 supports delegation across domain boundaries, you can configure Kerberos constrained delegation across a domain boundary if the domain functional level is Windows Server 2012. Another alternative is to configure Analysis Services for HTTP access and use IIS authentication methods on the client connection. For more information, see Configure HTTP Access to Analysis Services on Internet Information Services (IIS) 8.0.</p>
<p>Step 2: Register the SPN</p>	<p>Before setting up constrained delegation, you must register a Service Principle Name (SPN) for the Analysis Services instance. You will need the Analysis Services SPN when configuring Kerberos constrained delegation for middle tier services. See SPN registration for an Analysis Services instance for instructions.</p> <p>A Service Principle Name (SPN) specifies the unique identity of a service in a domain configured for Kerberos authentication. Client connections using integrated security often request an SPN as part of SSPI authentication. The request is forwarded to an Active Directory Domain Controller (DC), with the KDC granting a ticket if the SPN presented by the client has a matching SPN registration in Active Directory.</p>

TASK	DESCRIPTION
Step 3: Configure constrained delegation	<p>After validating the accounts you want to use and registering SPNs for those accounts, your next step is to configure up-level services, such as IIS, Reporting Services, or SharePoint web services for constrained delegation, specifying the Analysis Services SPN as the specific service for which delegation is allowed.</p> <p>Services that run in SharePoint, such as Excel Services or Reporting Services in SharePoint mode, often host workbooks and reports that consume Analysis Services multidimensional or tabular data. Configuring constrained delegation for these services is a common configuration task, and necessary for supporting data refresh from Excel Services. The following links provide instructions for SharePoint services, as well as other services likely to present a downstream data connection request for Analysis Services data:</p> <p>Identity delegation for Excel Services (SharePoint Server 2010) or How to configure Excel Services in SharePoint Server 2010 for Kerberos authentication</p> <p>Identity delegation for PerformancePoint Services (SharePoint Server 2010)</p> <p>Identity delegation for SQL Server Reporting Services (SharePoint Server 2010)</p> <p>For IIS 7.0 see Configure Windows Authentication (IIS 7.0) or How to configure SQL Server 2008 Analysis Services and SQL Server 2005 Analysis Services to use Kerberos authentication.</p>
Step 4: Test connections	<p>When testing, connect from remote computers, under different identities, and query Analysis Services using the same applications as business users. You can use SQL Server Profiler to monitor the connection. You should see the user identity on the request. For more information, see Test for impersonated or delegated identity in this section.</p>

Configure Analysis Services for trusted delegation

Configuring Analysis Services for Kerberos constrained delegation allows the service to impersonate a client identity on a down-level service, such as the relational database engine, so that data can be queried as if the client was connected directly.

Delegation scenarios for Analysis Services are limited to tabular models configured for **DirectQuery** mode. This is the only scenario in which Analysis Services can pass delegated credentials to another service. In all other scenarios, such as SharePoint scenarios mentioned in the previous section, Analysis Services is on the receiving end of the delegation chain. For more information about DirectQuery, see [DirectQuery Mode](#).

NOTE

A common misconception is that ROLAP storage, processing operations, or access to remote partitions somehow introduce requirements for constrained delegation. This is not the case. All of these operations are executed directly by the service account (also referred to as the processing account), on its own behalf. Delegation is not required for these operations in Analysis Services, given that permissions for such operations are granted directly to the service account (for example, granting db_datareader permissions on the relational database so that the service can process data). For more information about server operations and permissions, see [Configure Service Accounts \(Analysis Services\)](#).

This section explains how to set up Analysis Services for trusted delegation. After you complete this task, Analysis Services will be able to pass delegated credentials to SQL Server, in support of DirectQuery mode used in Tabular solutions.

Before you start:

- Verify that Analysis Services is started.
- Verify the SPN registered for Analysis Services is valid. For instructions, see [SPN registration for an Analysis Services instance](#)

When both prerequisites are satisfied, continue with the following steps. Note that you must be a domain administrator to set up constrained delegation.

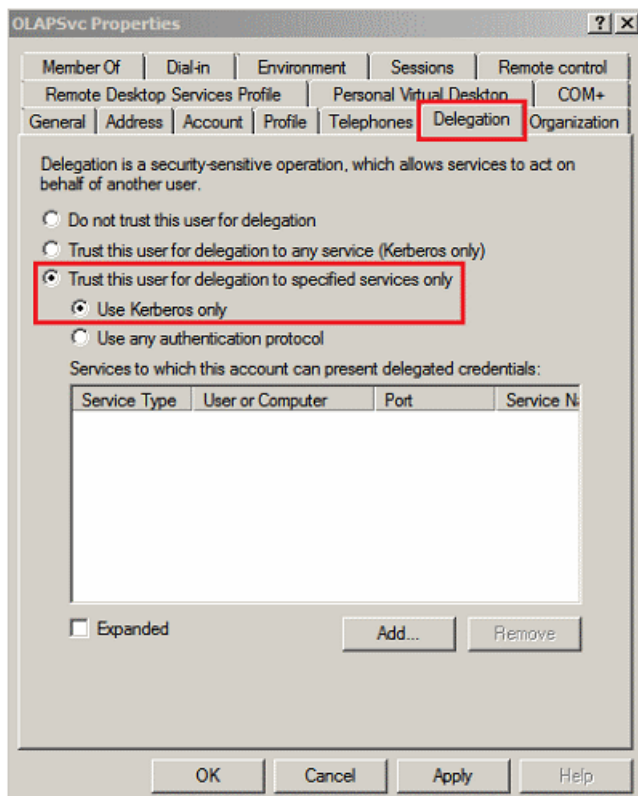
1. In Active Directory Users and Computers, find the service account under which Analysis Services runs. Right-click the service account and choose **Properties**.

For illustrative purposes, the following screenshots use OlapSvc and SQLSvc to represent Analysis Services and SQL Server, respectively.

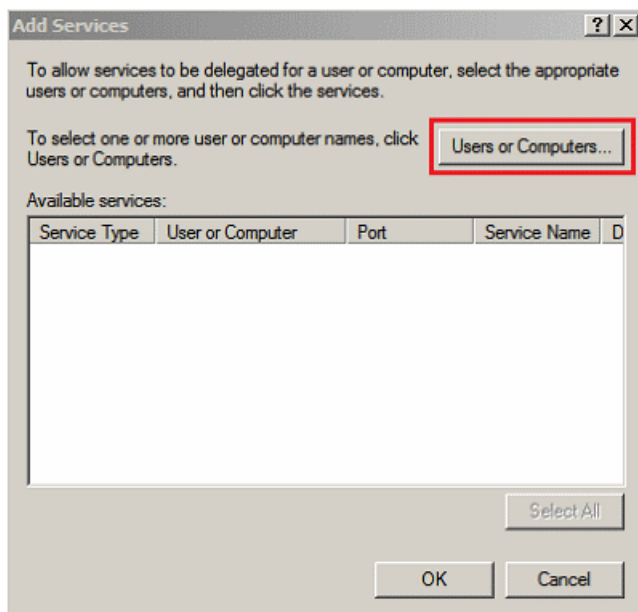
OlapSvc is the account that will be configured for constrained delegation to SQLSvc. When you complete this task, OlapSvc will have permission to pass delegated credentials on a service ticket to SQLSvc, impersonating the original caller when requesting data.

2. On the Delegation tab, select **Trust this user for delegation to specified services only**, followed by **Use Kerberos only**. Click **Add** to specify which service Analysis Services is permitted to delegate credentials.

Delegation tab appears only when the user account (OlapSvc) is assigned to a service (Analysis Services), and the service has an SPN registered for it. SPN registration requires that the service is running.

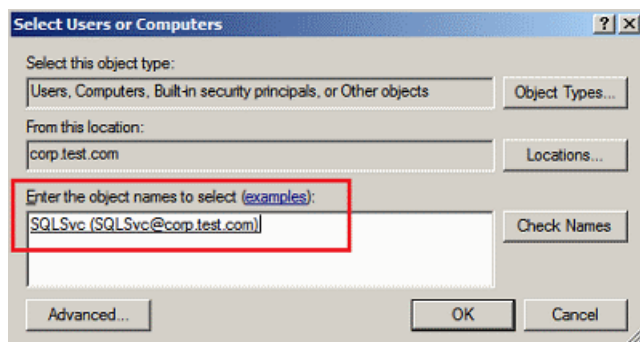


- On the Add Services page, click **Users or Computers**.

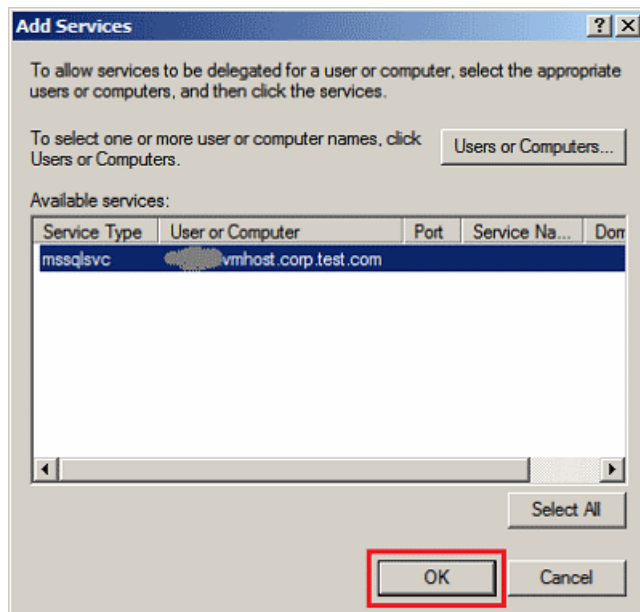


- On the Select Users or Computer page, enter the account used to run the SQL Server instance providing data to Analysis Services tabular model databases. Click **OK** to accept the service account.

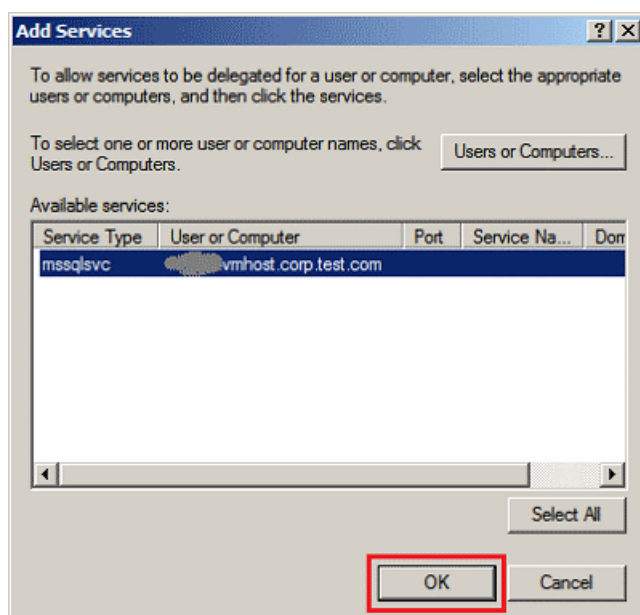
If you cannot select the account you want, verify that SQL Server is running and has an SPN registered for that account. For more information about SPNs for the database engine, see [Register a Service Principal Name for Kerberos Connections](#).



- The SQL Server instance should now appear in Add Services. Any service also using that account will also appear in the list. Choose the SQL Server instance you want to use. Click **OK** to accept the instance.



- The properties page of the Analysis Services service account should now look similar to the following screenshot. Click **OK** to save your changes.



- Test for successful delegation by connecting from a remote client computer, under a different identity, and query the tabular model. You should see the user identity on the request in SQL Server Profiler.

Test for impersonated or delegated identity

Use SQL Server Profiler to monitor the identity of the user who is querying data.

1. Start **SQL Server Profiler** on the Analysis Services instance and then start a new trace.
2. In Events Selection, verify that **Audit Login** and **Audit Logout** are checked in the Security Audit section.
3. Connect to Analysis Services via an application service (such as SharePoint or Reporting Services) from a remote client computer. The Audit Login event will show the identity of the user connecting to Analysis Services.

Thorough testing will require the use of network monitoring tools that can capture Kerberos requests and responses on the network. The Network Monitor utility (netmon.exe), filtered for Kerberos, can be used for this task. For more information about using Netmon 3.4 and other tools for testing Kerberos authentication, see [Configuring Kerberos authentication: Core configuration \(SharePoint Server 2010\)](#).

Additionally, see [The Most Confusing Dialog Box in Active Directory](#) for a thorough description of each option in the Delegation tab of the Active Directory object's property dialog box. This article also explains how to use LDP to test and interpret test results.

See Also

[Microsoft BI Authentication and Identity Delegation](#)

[Mutual Authentication Using Kerberos](#)

[Connect to Analysis Services](#)

[SPN registration for an Analysis Services instance](#)

[Connection String Properties \(Analysis Services\)](#)

SPN registration for an Analysis Services instance

5/16/2018 • 10 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

A Service Principle Name (SPN) uniquely identifies a service instance in an Active Directory domain when Kerberos is used to mutually authenticate client and service identities. An SPN is associated with the logon account under which the service instance runs.

For client applications connecting to Analysis Services via Kerberos authentication, the Analysis Services client libraries construct an SPN using the host name from the connection string and other well-known variables, such as the service class, that are fixed in any given release of Analysis Services.

For mutual authentication to occur, the SPNs constructed by the client must match a corresponding SPN object on an Active Directory Domain Controller (DC). This means that you might need to register multiple SPNs for a single Analysis Services instance to cover all of the ways in which a user might specify the host name on a connection string. For example, you probably need two SPNs to handle both the fully-qualified domain name (FQDN) of a server, as well as the short computer name. Correctly registering the Analysis Services SPN is essential for a successful connection. If the SPN is non-existent, malformed, or duplicated, the connection will fail.

SPN registration is a manual task performed by the Analysis Services administrator. Unlike the SQL Server database engine, Analysis Services never auto-registers its SPN at service startup. Manual registration is required when Analysis Services runs under the default virtual account, a domain user account, or a built-in account, including a per-service SID.

SPN registration is not required if the service runs under a predefined managed service account created by a domain administrator. Note that depending on the functional level of your domain, registering an SPN can require domain administrator permissions.

TIP

Microsoft Kerberos Configuration Manager for SQL Server is a diagnostic tool that helps troubleshoot Kerberos related connectivity issues with SQL Server. For more information, see [Microsoft Kerberos Configuration Manager for SQL Server](#).

This topic contains the following sections:

[When SPN registration is required](#)

[SPN format for Analysis Services](#)

[SPN registration for a virtual account](#)

[SPN registration for a domain account](#)

[SPN registration for a built-in account](#)

[SPN registration for a named instance](#)

[SPN registration for an SSAS cluster](#)

[SPN registration for SSAS instances configured for HTTP access](#)

[SPN registration for SSAS instances listening on fixed ports](#)

When SPN registration is required

Any client connection that specifies "SSPI=Kerberos" on the connection string will introduce SPN registration requirements for an Analysis Services instance.

SPN registration is required under the following circumstances. For more detailed information, see [Configure Analysis Services for Kerberos constrained delegation](#).

- Identity delegation is necessary to flow the user identity from the client application or middle-tier service to Analysis Services. Identity delegation is typically used when per-user permissions or filters are defined on specific objects.

A common scenario involving identity delegation is configuring middle-tier services, such as Excel Services or Reporting Services, for constrained delegation for the purpose of impersonating a user identity when retrieving data in Analysis Services. To support this behavior, you must provide an Analysis Services SPN as the destination service when configuring Excel Services or Reporting Services for constrained delegation.

- Analysis Services delegates a user identity when retrieving data from a SQL Server relational database for tabular databases using DirectQuery mode. This is the only scenario in which Analysis Services will delegate the user identity to another service.

SPN format for Analysis Services

Use **setspn** to register an SPN. On newer operating systems, **setspn** is installed as a system utility. For more information, see [SetSPN](#).

The following table describes each part of an Analysis Services SPN.

ELEMENT	DESCRIPTION
Service class	MSOLAPSvc.3 identifies the service as an Analysis Services instance. The .3 is a reference to the version of the XMLA-over-TCP/IP protocol used in Analysis Services transmissions. It is unrelated to product release. As such, MSOLAPSvc.3 is the correct service class for SQL Server 2005, 2008, 2008 R2, 2012 and any future release of Analysis Services until the protocol itself is revised.
Host-name	<p>Identifies the computer on which the service is running. This can be a fully-qualified domain name or a NetBIOS name. You should register an SPN for both.</p> <p>When registering an SPN for the NetBIOS name of a server, be sure to use <code>SetupSPN -S</code> to check for duplicate registration. NetBIOS names are not guaranteed to be unique in a forest, and having a duplicate SPN registration will cause the connection to fail.</p> <p>For Analysis Services load balanced clusters, the host name should be the virtual name assigned to the cluster.</p> <p>Never create an SPN using the IP address. Kerberos uses the DNS resolution capabilities of the domain. Specifying an IP address bypasses that capability.</p>

ELEMENT	DESCRIPTION
Port-number	Although the port number is part of SPN syntax, you never specify a port number when registering an Analysis Services SPN. The colon (:) character, typically used to provide a port number in standard SPN syntax, is used by Analysis Services to specify the instance name. For an Analysis Services instance, the port is assumed to be the default port (TCP 2383) or a port assigned by the SQL Server Browser service (TCP 2382).
Instance-name	<p>Analysis Services is a replicable service that can be installed multiple times on the same computer. Each instance is identified through its instance name.</p> <p>The instance name is prefixed with a colon (:) character. For example, given a host computer named SRV01 and a named instance of SSAS-Tabular, the SPN should be SRV01:SSAS-Tabular.</p> <p>Note that the syntax for specifying a named Analysis Services instance differs from that used by other SQL Server instances. Other services uses a backslash (\) to append the instance name in an SPN.</p>
Service-account	This is the startup account of the MSSQLServerOLAPService Windows service. It can be a Windows domain user account, virtual account, managed service account (MSA) or a built-in account such as a per-service SID, NetworkService, or LocalSystem. A Windows domain user account can be formatted as domain\user or user@domain.

SPN registration for a virtual account

Virtual accounts are the default account type for SQL Server services. The virtual account is **NT Service\MSOLAPService** for a default instance and **NT Service\MSOLAP\$<instance-name>** for a named instance.

As the name implies, these accounts do not exist in Active Directory. A virtual account exists only on the local computer. When connecting to external services, applications, or devices, the connection is made using the local machine account. For this reason, an SPN registration for Analysis Services running under a virtual account is actually an SPN registration for the machine account.

Example syntax for a default instance running as NT Service\MSOLAPService

This example shows **setspn** syntax for Analysis Services default instance running under the default virtual account. In this example, the computer host name is **AW-SRV01**. As noted, SPN registration must specify the *machine account* instead of the virtual account, **NT Service\MSOLAPService**.

```
Setspn -s MSOLAPSvc.3/AW-SRV01.AdventureWorks.com AW-SRV01
```

NOTE

Remember to create two SPN registrations, one for the NetBIOS host name and a second for a fully-qualified domain name of the host. Different client applications use different host name conventions when connecting to Analysis Services. Having two SPN registrations ensures that both versions of the host name are accounted for.

Example syntax for a named instance running as NT Service\MSOLAP\$<instance-name>

This example shows **setspn** syntax for a named instance running under the default virtual account. In this example, the computer host name is **AW-SRV02**, and the instance name is **AW-FINANCE**. Again, it is the machine account that is specified for the SPN, rather than the virtual account **NT Service\MSOLAP\$<instance-name>**.

```
Setspn -s MSOLAPSvc.3/AW-SRV02.AdventureWorks.com:AW-FINANCE AW-SRV02
```

SPN registration for a domain account

Using a domain account to run an Analysis Services instance is a common practice.

For Analysis Services instances that run in a network or hardware load balanced cluster, a domain account is required, with each instance in the cluster running under the same domain account.

Example syntax for a default instance running as a domain user

This example shows **setspn** syntax for Analysis Services default instance running under a domain user account, **SSAS-Service**, in the AdventureWorks domain.

```
Setspn -s msolapsvc.3\AW-SRV01.Adventureworks.com AdventureWorks\SSAS-Service
```

TIP

Verify whether the SPN was created for the Analysis Services server by running `Setspn -L <domain account>` or `Setspn -L <machinename>`, depending on how the SPN was registered. You should see `MSOLAPVC.3/<hostname>` in the list.

SPN registration for a built-in account

Although this practice is not recommended, older Analysis Services installations are sometimes configured to run under built-in accounts like Network Service, Local Service, or Local System.

Example syntax for a default instance running under a built-in account

SPN registration for a service running under a built-in account or per-service SID is equivalent to the SPN syntax used for the virtual account. Instead of the account name, use the machine account:

```
Setspn -s MSOLAPSvc.3/AW-SRV01.AdventureWorks.com AW-SRV01
```

SPN registration for a named instance

Named instances of Analysis Services use dynamic port assignments that are detected by the SQL Server Browser service. When using a named instance, register an SPN for both the SQL Server Browser Service and the Analysis Services named instance. For more information, see [An SPN for the SQL Server Browser service is required when you establish a connection to a named instance of SQL Server Analysis Services or of SQL Server](#).

Example of SPN syntax for the SQL Browser Service running as LocalService

The service class is **MSOLAPDisco.3**. By default, this service runs as NT AUTHORITY\LocalService, which means SPN registration is set for the machine account. In this example, the machine account is **AW-SRV01**, corresponding to the computer name.

```
Setspn -S MSOLAPDisco.3/AW-SRV01.AdventureWorks.com AW-SRV01
```

SPN registration for an SSAS cluster

For Analysis Services failover clusters, the host name should be the virtual name assigned to the cluster. This is the SQL Server network name, specified during SQL Server Setup when you installed Analysis Services on top of an existing WSFC. You can find this name in Active Directory. You can also find it in **Failover Cluster Manager** | **Role** | **Resources** tab. The server name on the Resources tab is what should be used as the 'virtual name' in the SPN command.

SPN syntax for an Analysis Services cluster

```
Setspn -s msolapsvc.3/<virtualname.FQDN > <domain user account>
```

Recall that nodes in an Analysis Services cluster are required to use the default port (TCP 2383) and run under the same domain user account so that each node has the same SID. See [How to Cluster SQL Server Analysis Services](#) for more information.

SPN registration for SSAS instances configured for HTTP access

Depending on solution requirements, you might have configured Analysis Services for HTTP access. If your solution includes IIS as a middle tier component, and Kerberos authentication is a solution requirement, you might need to manually register an SPN for IIS. For more information, see "Configure the settings on the computer running IIS" in [How to configure SQL Server 2008 Analysis Services and SQL Server 2005 Analysis Services to use Kerberos authentication](#).

In terms of SPN registration for the Analysis Services instance, there is no difference between an instance configured for TCP or HTTP. The connection to Analysis Services from IIS, using the MSMDBPUMP ISAPI extension, is always TCP.

This means that you can use the instructions from previous sections for default or named instance to register the SPN. When specifying the host name, be sure to use the host name you specified in the msmdpump.ini file when you configured the service for HTTP access.

For more information about HTTP access, see [Configure HTTP Access to Analysis Services on Internet Information Services \(IIS\) 8.0](#).

SPN registration for SSAS instances listening on fixed ports

You cannot specify a port number on an Analysis Services SPN registration. If you installed Analysis Services as the default instance and configured it to listen on a fixed port, you must now configure it to listen on the default port (TCP 2383). For named instances, you need to use SQL Server Browser service and dynamic port assignments.

An Analysis Services instance can only listen on a single port. Using multiple ports is not supported. For more information about port configuration, see [Configure the Windows Firewall to Allow Analysis Services Access](#).

See Also

[Microsoft BI Authentication and Identity Delegation Mutual Authentication Using Kerberos](#)

[How to configure SQL Server 2008 Analysis Services and SQL Server 2005 Analysis Services to use Kerberos authentication](#)

[Service Principal Names \(SPNs\) SetSPN Syntax \(Setspn.exe\)](#)

[What SPN do I use and how does it get there?](#)

[SetSPN](#)

[Service Accounts Step-by-Step Guide](#)

[Configure Windows Service Accounts and Permissions](#)

[How to use SPNs when you configure Web applications that are hosted on Internet Information Services](#)

[what's new in service accounts](#)

[Configure Kerberos authentication for SharePoint 2010 Products \(white paper\)](#)

Configure HTTP Access to Analysis Services on IIS

8.0

5/16/2018 • 17 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

This article explains how to set up an HTTP endpoint for accessing an Analysis Services instance. You can enable HTTP access by configuring MSMDPUMP.dll, an ISAPI extension that runs in Internet Information Services (IIS) and pumps data to and from client applications and an Analysis Services server. This approach provides an alternative means for connecting to Analysis Services when your BI solution calls for the following capabilities:

- Client access is over Internet or extranet connections, with restrictions on which ports can be enabled.
- Client connections are from non-trusted domains in the same network.
- Client application runs in a network environment that allows HTTP but not TCP/IP connections.
- Client applications cannot use the Analysis Services client libraries (for example, a Java application running on a UNIX server). If you cannot use the Analysis Services client libraries for data access, you can use SOAP and XML/A over a direct HTTP connection to an Analysis Services instance.
- Authentication methods other than Windows integrated security are required. Specifically, you can use Anonymous connections and Basic authentication when configuring Analysis Services for HTTP access. Digest, Forms, and ASP.NET authentication are not supported. A requirement of Basic authentication is one of the primary reasons for enabling HTTP access. To learn more, see [Microsoft BI Authentication and Identity Delegation](#).

You can configure HTTP access for any supported version or edition of Analysis Services, running either tabular mode or multidimensional mode. Local cubes are an exception. You cannot connect to a local cube via an HTTP endpoint.

Setting up HTTP access is a post-installation task. Analysis Services must be installed before you can configure it for HTTP access. As the Analysis Services administrator, you will need to grant permissions to Windows accounts before HTTP access is possible. Additionally, it is a best practice to validate your installation first, ensuring that it is fully operational before configuring the server any further. After HTTP access is configured, you can use both the HTTP endpoint and the regular network name of the server over TCP/IP. Setting up HTTP access does not invalidate other approaches for data access.

As you move forward with MSMDPUMP configuration, remember there are two connections to consider: client-to-IIS, IIS-to-SSAS. The instructions in this article are about IIS-to-SSAS. Your client application might require additional configuration before it can connect to IIS. Decisions such as whether to use SSL, or how to configure bindings, are out of scope for this article. See [Web Server \(IIS\)](#) for more information about IIS.

Overview

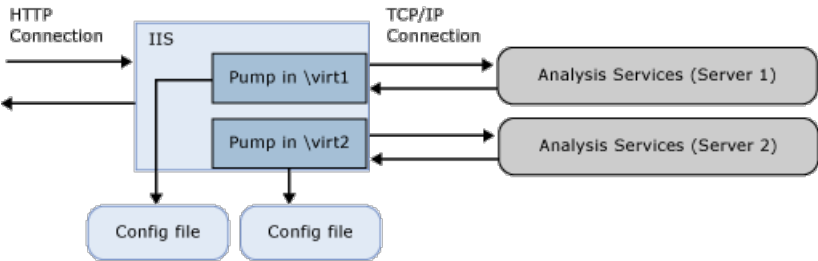
MSMDPUMP is an ISAPI extension that loads into IIS and provides redirection to a local or remote Analysis Services instance. By configuring this ISAPI extension, you create an HTTP endpoint to an Analysis Services instance.

You must create and configure one virtual directory for each HTTP endpoint. Each endpoint will need its own set of MSMDPUMP files, for each Analysis Services instance you want to connect to. A configuration file in this file

set specifies the name of the Analysis Services instance used for each HTTP endpoint.

On IIS, MSMDPUMP connects to Analysis Services using the Analysis Services OLE DB provider over TCP/IP. Although client requests can originate outside of domain trust, both Analysis Services and IIS must be in the same domain or in trusted domains in order for the native connection to succeed.

When MSMDPUMP connects to Analysis Services, it does so under a Windows user identity. This account will either be the Anonymous account if you configured the virtual directory for anonymous connections, or a Windows user account. The account must have the appropriate data access rights on the Analysis Services server and database.



The following table lists additional considerations when you enable HTTP access for different scenarios.

SCENARIO	CONFIGURATION
IIS and Analysis Services on the same computer	<p>This is the simplest configuration because it allows you to use the default configuration (where the server name is localhost), the local Analysis Services OLE DB provider, and Windows integrated security with NTLM. Assuming that the client is also in the same domain, authentication is transparent to the user, with no additional work on your part.</p>
IIS and Analysis Services on different computers	<p>For this topology, you must install the Analysis Services OLE DB provider on the web server. You must also edit the msmdpump.ini file to specify the location of Analysis Services instance on the remote computer.</p> <p>This topology adds a double-hop authentication step, where credentials must flow from the client to the web server, and on to the backend Analysis Services server. If you are using Windows credentials and NTLM, you will get an error because NTLM does not allow delegation of client credentials to a second server. The most common solution is to use Basic authentication with Secure Sockets Layer (SSL), but this will require users to provide a user name and password when accessing the MSMDPUMP virtual directory. A more straightforward approach might be to enable Kerberos and configure Analysis Services constrained delegation so that users can access Analysis Services in a transparent manner. See Configure Analysis Services for Kerberos constrained delegation for details.</p> <p>Consider which ports to unblock in Windows Firewall. You will need to unblock ports on both servers to allow access to the web application on IIS, and to Analysis Services on a remote server.</p>

SCENARIO	CONFIGURATION
Client connections are from a non-trusted domain or an extranet connection	<p>Client connections from a non-trusted domain introduce further restrictions on authentication. By default, Analysis Services uses Windows integrated authentication, which requires users to be on the same domain as the server. If you have Extranet users who connect to IIS from outside the domain, those users will get a connection error if the server is configured to use the default settings.</p> <p>Workarounds include having Extranet users connect through a VPN using domain credentials. However, a better approach might be to enable Basic authentication and SSL on your IIS web site.</p>

Prerequisites

The instructions in this article assume IIS is already configured and that Analysis Services is already installed. Windows Server 2012 ships with IIS 8.x as a server role that you can enable on the system.

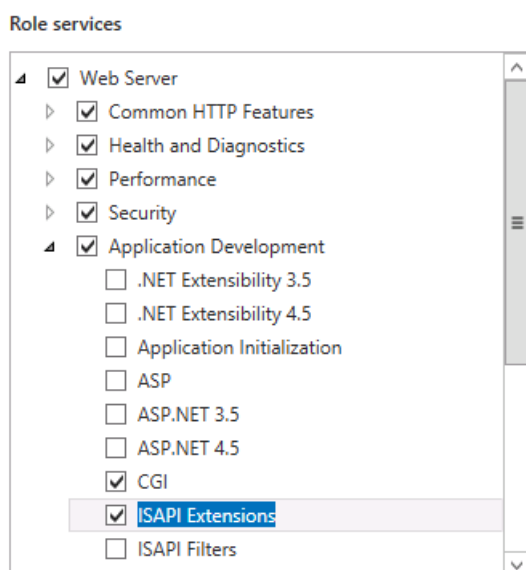
Extra configuration in IIS 8.0

The default configuration of IIS 8.0 is missing components that are necessary for HTTP access to Analysis Services. These components, found in the **Security** and **Application Development** feature areas of the **Web Server (IIS)** role, include the following:

- **Security | Windows Authentication**, or **Basic Authentication**, and any other security features required for your data access scenario.
- **Application Development | CGI**
- **Application Development | ISAPI Extensions**

To verify or add these components, use **Server Manager | Manage | Add Roles and Features**. Step through the wizard until you get to **Server Roles**. Scroll down to find **Web Server (IIS)**.

1. Open **Web Server | Security** and choose the authentication methods.
2. Open **Web Server | Application Development** and choose **CGI** and **ISAPI Extensions**.



When IIS is on a remote server

A remote connection between IIS and Analysis Services requires that you install the Analysis Services OLE DB provider (MSOLAP) on the Windows server running IIS.

3. Go to the download page for [SQL Server 2014 Feature Pack](#)
4. Click the red Download button.
5. Scroll down to find ENU\x64\SQL_AS_OLEDB.msi
6. Follow the instructions in the wizard to complete the installation.

NOTE

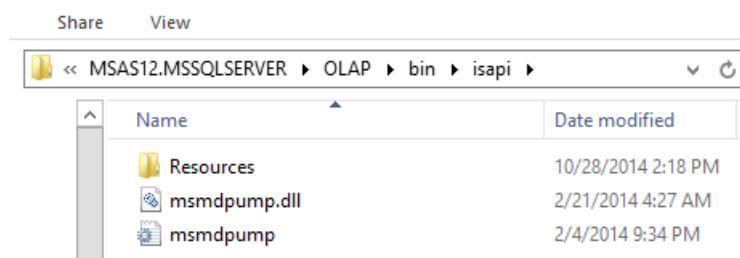
Remember to unblock the ports in Windows Firewall to allow client connections to a remote Analysis Services server. For more information, see [Configure the Windows Firewall to Allow Analysis Services Access](#).

Step 1: Copy the MSMDPUMP files to a folder on the Web server

Each HTTP endpoint that you create must have its own set of MSMDPUMP files. In this step, you copy the MSMDPUMP executable, configuration file, and resource folder from the Analysis Services program folders to a new virtual directory folder that you will create on the file system of the computer running IIS.

The drive must be formatted for the NTFS file system. The path to the folder that you create must not contain any spaces.

1. Copy the following files, found at <drive>:\Program Files\Microsoft SQL Server\<instance>\OLAP\bin\isapi: MSMDPUMP.DLL, MSMDPUMP.INI, and a Resources folder.



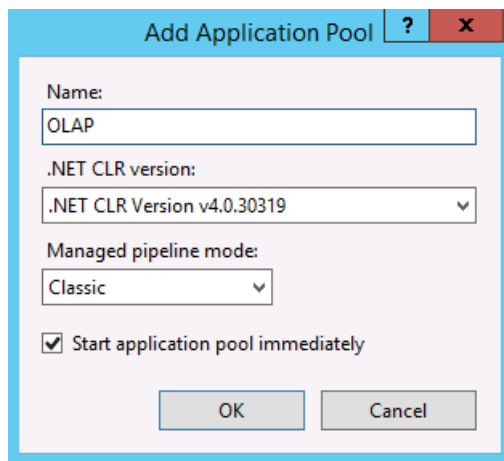
2. On the web server, create a new folder: <drive>:\inetpub\wwwroot**OLAP**
3. Paste the files that you previously copied into this new folder.
4. Verify that the \inetpub\wwwroot\OLAP folder on your web server contains the following:
MSMDPUMP.DLL, MSMDPUMP.INI, and a Resources folder. Your folder structure should look like this:
 - <drive>:\inetpub\wwwroot\OLAP\MSMDPUMP.dll
 - <drive>:\inetpub\wwwroot\OLAP\MSMDPUMP.ini
 - <drive>:\inetpub\wwwroot\OLAP\Resources

Step 2: Create an application pool and virtual directory in IIS

Next, create an application pool and an endpoint to the pump.

Create an application pool

1. Start IIS Manager.
2. Open the server folder, right-click **Application Pools** and then click **Add Application Pool**. Create an application pool named **OLAP**, using the .NET Framework, with Managed pipeline mode set to **Classic**.



Add Application Pool [?] [X]

Name:

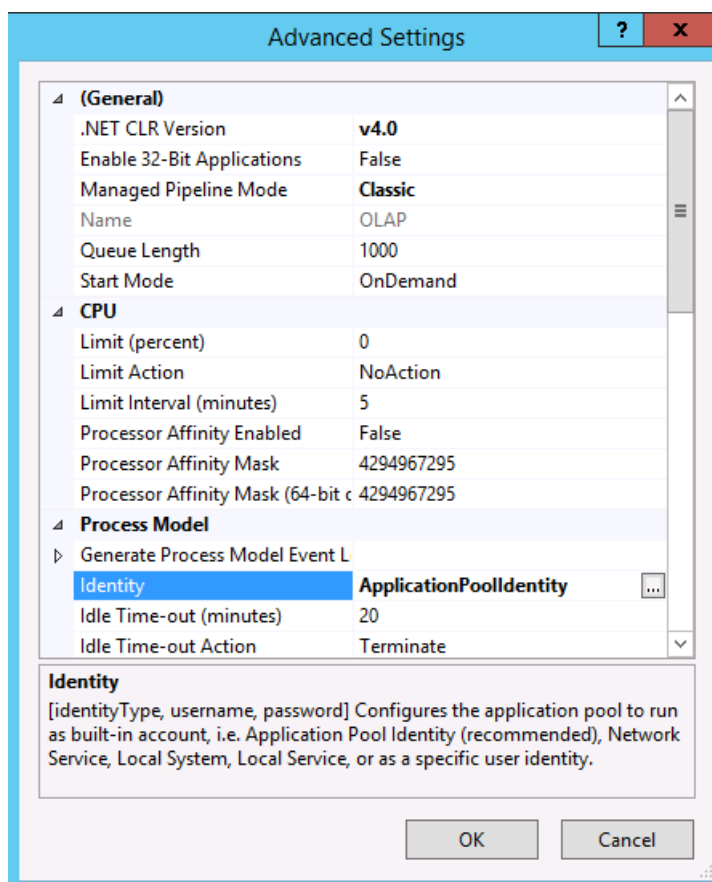
.NET CLR version:

Managed pipeline mode:

☒ Start application pool immediately

OK Cancel

- By default, IIS creates application pools using **ApplicationPoolIdentity** as the security identity, which is a valid choice for HTTP access to Analysis Services. If you have specific reasons for changing the identity, right-click **OLAP**, and then select **Advanced Settings**. Select **ApplicationPoolIdentity**. Click the **Change** button for this property to replace the built-in account with the custom account you want to use.



Advanced Settings [?] [X]

(General)

.NET CLR Version	v4.0
Enable 32-Bit Applications	False
Managed Pipeline Mode	Classic
Name	OLAP
Queue Length	1000
Start Mode	OnDemand

CPU

Limit (percent)	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit c	4294967295

Process Model

Generate Process Model Event L

Identity	ApplicationPoolIdentity [...]
Idle Time-out (minutes)	20
Idle Time-out Action	Terminate

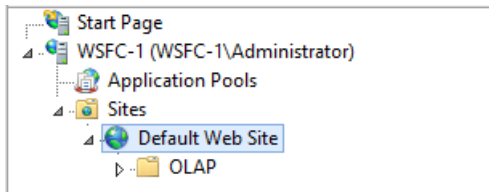
Identity
 [identityType, username, password] Configures the application pool to run as built-in account, i.e. Application Pool Identity (recommended), Network Service, Local System, Local Service, or as a specific user identity.

OK Cancel

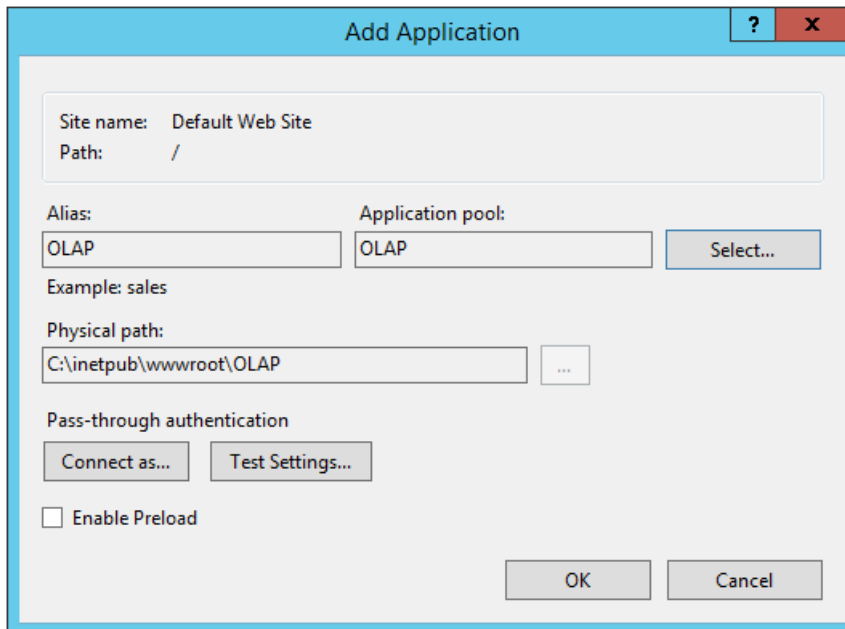
- By default, on a 64-bit operating system, IIS sets the **Enable 32-bit Applications** property to **false**. If you copied msmdpump.dll from a 64-bit installation of Analysis Services, this is the correct setting for the MSMDPUMP extension on a 64-bit IIS server. If you copied the MSMDPUMP binaries from a 32-bit installation, set it to **true**. Check this property now in **Advanced Settings** to ensure it is set correctly.

Create an application

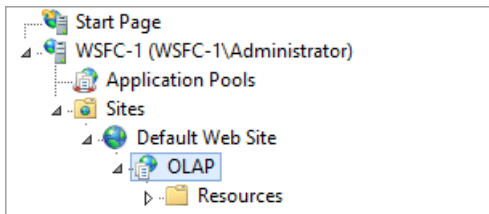
- In IIS Manager, open **Sites**, open **Default Web Site**. You should see a folder named **Olap**. This is a reference to the OLAP folder you created under \inetpub\wwwroot.



2. Right-click the folder and choose **Convert to Application**.
3. In Add Application, enter **OLAP** for the alias. Click **Select** to choose the OLAP application pool. Physical Path should be set to C:\inetpub\wwwroot\OLAP



4. Click **OK**. Refresh the web site and notice that the OLAP folder is now an application under the default web site. The virtual path to the MSMDPUMP file is now established.



NOTE

Previous versions of these instructions included steps for creating a virtual directory. That step is no longer necessary.

Step 3: Configure IIS authentication and add the extension

In this step, you further configure the SSAS virtual directory you just created. You will specify an authentication method and then add a script map. Supported authentication methods for Analysis Services over HTTP include:

- Windows authentication (Kerberos or NTLM)
- Basic authentication
- Anonymous authentication

Windows authentication is considered the most secure, and leverages existing infrastructure for networks that use Active Directory. To use Windows authentication effectively, all browsers, client applications, and server applications must support it. This is the most secure and recommended mode,

but it requires that IIS be able to access a Windows domain controller that can authenticate the identity of the user requesting a connection.

For topologies that put Analysis Services and IIS on different computers, you will need to address double-hop issues that arise when a user identity needs to be delegated to a second service on a remote machine, typically by enabling Analysis Services for Kerberos constrained delegation. For more information, see [Configure Analysis Services for Kerberos constrained delegation](#).

Basic authentication is used when you have Windows identities, but user connections are from non-trusted domains, prohibiting the use of delegated or impersonated connections. Basic authentication lets you specify a user identity and password on a connection string. Instead of using the security context of the current user, credentials on the connection string are used to connect to Analysis Services. Because Analysis Services supports only Windows authentication, any credentials passed to it must be a Windows user or group that is a member of the domain in which Analysis Services is hosted.

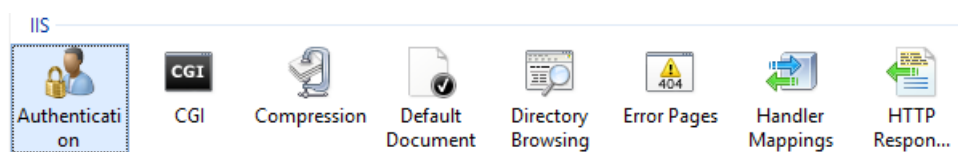
Anonymous authentication is often used during initial testing because its ease of configuration helps you to quickly validate HTTP connectivity to Analysis Services. In just a few steps, you can assign a unique user account as the identity, grant that account permissions in Analysis Services, use the account to verify data access in a client application, and then disable Anonymous authentication when testing is complete.

You can also use Anonymous authentication in a production environment if your users do not have Windows user accounts, but follow best practices by locking down permissions on the host system, as called out in this article: [Enable Anonymous Authentication \(IIS 7\)](#). Be sure that authentication is set on the virtual directory, and not on the parent web site, to further reduce the level of account access.

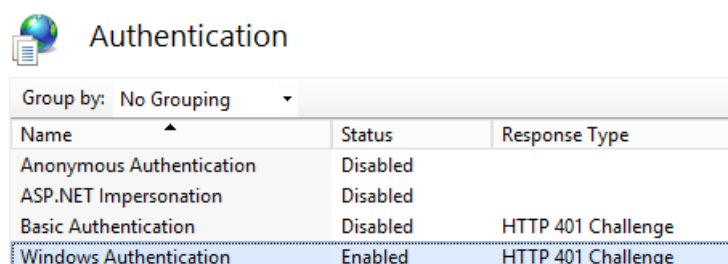
When Anonymous is enabled, any user connection to the HTTP endpoint is allowed to connect as the anonymous user. You won't be able to audit individual user connections, nor use the user identity to select data from a model. As you can see, using Anonymous impacts everything from model design, to data refresh and access. However, if users do not have a Windows user login to start with, using the Anonymous account might be your only option.

Set the authentication type and add a script map

1. In IIS Manager, open **Sites**, open **Default Web Site**, and then select the **OLAP** virtual directory.
2. Double-click **Authentication** in the IIS section of the main page.



3. Enable **Windows Authentication** if you are using Windows integrated security.



4. Alternatively, enable **Basic Authentication** if your client and server applications are in different domains. This mode requires the user to enter a user name and password. The user name and password are transmitted over the HTTP connection to IIS. IIS will try to impersonate the user using the provided credentials when connecting to MSMDPUMP, but the credentials will not be delegated to Analysis Services. Instead, you will need to pass a valid user name and password on a connection, as described in

Step 6 in this document.

IMPORTANT

Please note that it is imperative for anyone building a system where the password is transmitted to have ways of securing the communication channel. IIS provides a set of tools that help you secure the channel. For more information, see [How to Set Up SSL on IIS 7](#).

5. Disable **Anonymous Authentication** if you are using Windows or Basic authentication. When Anonymous authentication is enabled, IIS will always use it first, even if other authentication methods are enabled.

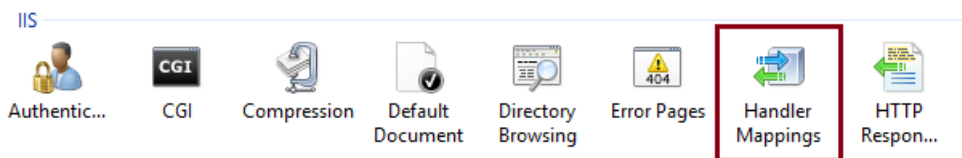
Under Anonymous authentication, the pump (msmdpump.dll) runs as the user account you established for anonymous user. There is no distinction between the user connecting to IIS and the user connecting to Analysis Services. By default, IIS uses the IUSR account, but you can change it to a domain user account that has network permissions. You'll need this capability if IIS and Analysis Services are on different computers.

For instructions on how to configure credentials for Anonymous authentication, see [Anonymous Authentication](#).

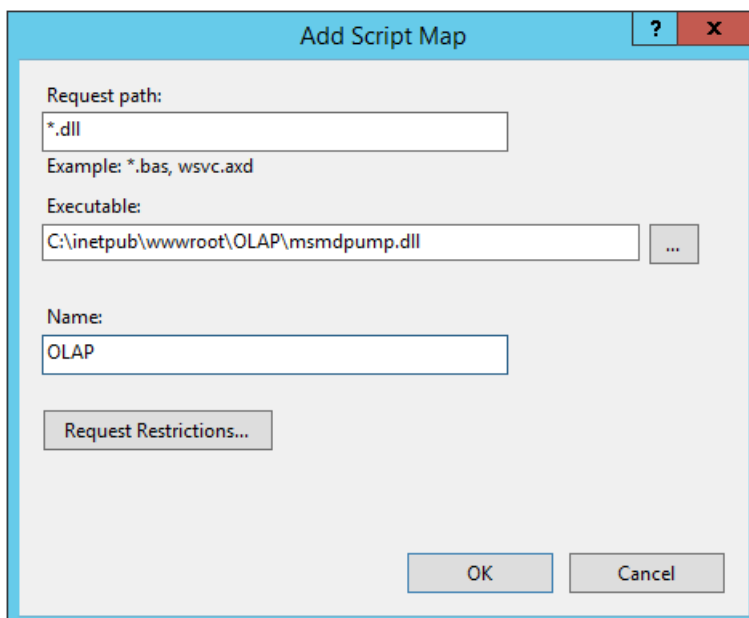
IMPORTANT

Anonymous authentication is most likely found in an extremely controlled environment, where users are given or denied access by way of access control lists in the file system. For best practices, see [Enable Anonymous Authentication \(IIS 7\)](#).

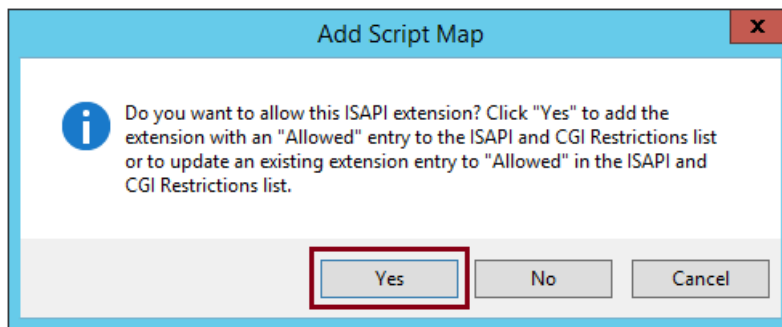
6. Click the **OLAP** virtual directory to open the main page. Double-click **Handler Mappings**.



7. Right-click anywhere on the page and then select **Add Script Map**. In the Add Script Map dialog box, specify ***.dll** as the request path, specify **c:\inetpub\wwwroot\OLAP\msmdpump.dll** as the executable, and type **OLAP** as the name. Keep all of the default restrictions associated with this script map.



8. When prompted to allow the ISAPI extension, click **Yes**.



Step 4: Edit the MSMDPUMP.INI file to set the target server

The MSMDPUMP.INI file specifies the Analysis Services instance that MSMDPUMP.DLL connects to. This instance can be local or remote, installed as the default or as a named instance.

Open the msmdpump.ini file located in folder C:\inetpub\wwwroot\OLAP and take a look at the contents of this file. It should look like the following:

```
<ConfigurationSettings>
<ServerName>localhost</ServerName>
<SessionTimeout>3600</SessionTimeout>
<ConnectionPoolSize>100</ConnectionPoolSize>
</ConfigurationSettings>
```

If the Analysis Services instance for which you are configuring HTTP access is located on the local computer and installed as a default instance, there is no reason to change this setting. Otherwise, you must specify the server name (for example, <ServerName>ADWRKS-SRV01</ServerName>). For a server that is installed as a named instance, be sure to append the instance name (for example, <ServerName>ADWRKS-SRV01\Tabular</ServerName>).

By default, Analysis Services listens on TCP/IP port 2383. If you installed Analysis Services as the default instance, you do not need to specify any port in <ServerName> because Analysis Services knows how to listen on port 2383 automatically. However, you do need to allow inbound connections to that port in Windows Firewall. For more information, see [Configure the Windows Firewall to Allow Analysis Services Access](#).

If you configured a named or default instance of Analysis Services to listen on a fixed port, you must add the port number to the server name (for example, <ServerName>AW-SRV01:55555</ServerName>) and you must allow inbound connections in Windows Firewall to that port.

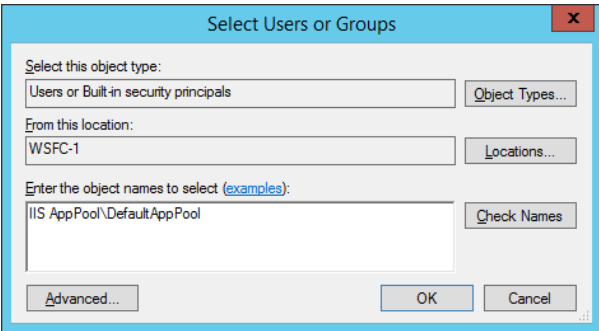
Step 5: Grant data access permissions

As previously noted, you will need to grant permissions on the Analysis Services instance. Each database object will have roles that provide a given level of permissions (read or read/write), and each role will have members consisting of Windows user identities.

To set permissions, you can use SQL Server Management Studio. Under the **Database | Roles** folder, you can create roles, specify database permissions, assign membership to Windows user or group accounts, and then grant read or write permissions on specific objects. Typically, **Read** permissions on a cube are sufficient for client connections that use, but do not update, model data.

Role assignment varies depending on how you configured authentication.

--	--

Anonymous	Add to the Membership list the account specified in Edit Anonymous Authentication Credentials in IIS. For more information, see Anonymous Authentication ,
Windows authentication	<p>Add to the Membership list the Windows user or group accounts requesting Analysis Services data via impersonation or delegation.</p> <p>Assuming Kerberos constrained delegation is used, the only accounts that need permissions are the Windows user and group accounts requesting access. No permissions are necessary for the application pool identity.</p>
Basic authentication	<p>Add to the Membership list the Windows user or group accounts that will be passed on the connection string.</p> <p>In addition, if you are passing credentials via EffectiveUserName on the connection string, then the application pool identity must have administrator rights on the Analysis Services instance. In SSMS, right-click the instance Properties Security Add. Enter the application pool identity. If you used the built-in default identity, the account is specified as IIS AppPool\DefaultAppPool.</p> 

For more information about setting permissions, see [Authorizing access to objects and operations \(Analysis Services\)](#).

Step 6: Test your configuration

The connection string syntax for MSMDPUMP is the URL to the MSMDPUMP.dll file.

If the web application is listening on a fixed port, append the port number to the server name or IP address (for example, `http://my-web-srv01:8080/OLAP/msmdpump.dll` or `http://123.456.789.012:8080/OLAP/msmdpump.dll`).

To quickly test the connection, you can open a connection using Internet Explorer, Microsoft Excel, or SQL Server Management Studio.

Troubleshoot connections using Internet Explorer

A connection request that terminates with this error might not give you much to go on: "Either a connection cannot be made to '<server name>', or Analysis Service is not running on the server".

To get a more informative error, do the following:

1. In **Internet Explorer** > **Internet Options** > **Advanced**, clear the checkbox for **Show Friendly HTTP messages**.
2. Retry the connection (for example, `http://my-web-srv01:8080/OLAP/msmdpump.dll`)

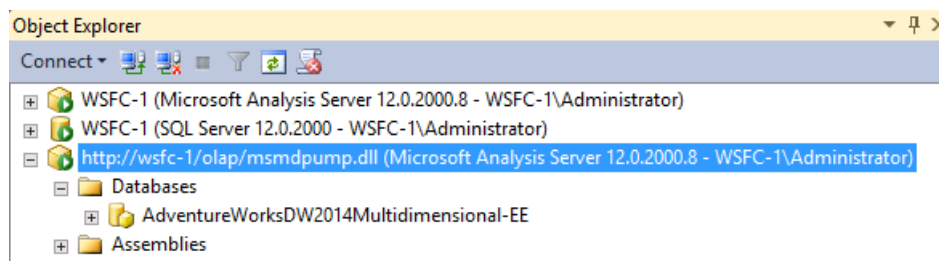
If you see an ERROR XML displayed in the browser window, you can eliminate MSMDPUMP as the potential cause and shift your focus to the certificate.

Test connections using SQL Server Management Studio

3. In Management Studio, in the Connect to Server dialog box, select **Analysis Services** as the server type. In Server name, enter the HTTP address of the msmdpump extension:

`http://my-web-srv01/OLAP/msmdpump.dll` .

Object Explorer displays the HTTP connection:



4. Authentication must be Windows authentication, and the person using Management Studio must be an Analysis Services administrator. An administrator can grant further permissions to enable access by other users.

Test connections using Excel

5. On the Data tab in Excel, in Get External Data, click **From Other Sources**, and then choose **From Analysis Services** to start the Data Connection wizard.
6. In Server name, enter the HTTP address of the msmdpump extension:

`http://my-web-srv01/OLAP/msmdpump.dll` .

7. For Log on credentials, choose **Use Windows Authentication** if you are using Windows integrated security or NTLM, or Anonymous user.

For Basic authentication, choose **Use the following User Name and Password**, and then specify the credentials used to sign on. The credentials you provide will be passed on the connection string to Analysis Services.

Test connections using AMO

You can test HTTP access programmatically using AMO, substituting the URL of the endpoint for the server name. For details, see [Forum Post \(How to sync SSAS 2008 R2 databases via HTTPS across domain/forest and firewall boundaries\)](#).

An example connection string illustrating the syntax for HTTP(S) access using Basic authentication:

```
Data Source=https://<servername>/olap/msmdpump.dll; Initial Catalog=AdventureWorksDW2012; Integrated Security=Basic; User ID=XXXX; Password=XXXXX;
```

For more information about setting up the connection programmatically, see [Establishing Secure Connections in ADOMD.NET](#).



As a final step, be sure to follow-up with more rigorous testing by using a client computer that runs in the network environment from which the connections will originate.

See Also

[Forum post \(http access using msmdpump and basic authentication\)](#)
[Configure the Windows Firewall to Allow Analysis Services Access](#)
[Authorizing access to objects and operations \(Analysis Services\)](#)

Client libraries (data providers) used for Analysis Services connections

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Analysis Services provides three client libraries, also known as **data providers**, for server and data access from tools and client applications. Tools like SSMS and SSDT, and applications like Power BI Desktop and Excel connect to Analysis Services by using these libraries. Two of the client libraries, ADOMD.NET and Analysis Services Management Objects (AMO), are managed client libraries. The Analysis Services OLE DB provider (MSOLAP DLL) is a native client library. Client libraries are the same for both SQL Server Analysis Services and Azure Analysis Services.

Where to get newer versions

The version installed on a client computer should match the major version of the server providing the data. If the server installation is newer than the data providers installed on the workstations in your network, you might need to install newer libraries.

Client libraries included in earlier SQL Server Feature Packs correspond to that SQL version; however, they may not be the latest. Connecting to Azure Analysis Services may require later versions. All versions are backward compatible.

To get the latest, see [Client libraries for connecting to Azure Analysis Services](#).

See also

[Connect to Analysis Services](#)

Disconnect Users and Sessions on Analysis Services Server

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

An administrator of Analysis Services may want to end user activity as part of workload management. You do this by canceling sessions and connections. Sessions can be formed automatically when a query is run (implicit), or named at the time of creation by the administrator (explicit). Connections are open conduits over which queries can be run. Both sessions and connections can be ended while they are active. For example, an administrator may want to end processing for a session if the processing is taking too long or if some doubt has arisen as to whether the command being executed was written correctly.

Ending Sessions and Connections

To manage sessions and connections, you can use Dynamic Management Views (DMVs) and XMLA:

1. In SQL Server Management Studio, connect to an Analysis Services instance.
2. Paste any one of the following DMV queries in an MDX query window to get a list of all sessions, connections, and commands that are currently executing:

```
Select * from $System.Discover_Sessions
```

```
Select * from $System.Discover_Connections
```

```
Select * from $System.Discover_Commands
```

3. Press F5 to execute the query.

The DMV query returns session and connection information in a tabular result set that is easier read and copy from.

Keep the query window open. In the next step, you will want to return to this page to copy the SPIDs of the session you want to disconnect.

To end a session, open a second XMLA query window.

4. Paste the following syntax into an MDX query window, replacing the ConnectionID, SessionID, or SPID placeholder with a valid value copied from the previous step.

```
<Cancel xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">  
  
  <ConnectionID>111</ConnectionID>  
  <SessionID>222</SessionID>  
  <SPID>333</SPID>  
  
  <CancelAssociated>1</CancelAssociated>  
</Cancel>
```

5. Press F5 to execute the cancel command.

Ending a connection cancels all sessions and SPIDs, closing the host session.

Ending a session stops all commands (SPIDs) that are running as part of that session.

Ending a SPID cancels a particular commend.

In rare cases, Analysis Services will not close a connection if it cannot track all the sessions and SPIDs associated with the connection (for example, when multiple sessions are open in an HTTP scenario).

For more information about the XMLA referenced in this topic, see [Execute Method \(XMLA\)](#) and [Cancel Element \(XMLA\)](#).

See Also

[Managing Connections and Sessions \(XMLA\)](#)

[BeginSession Element \(XMLA\)](#)

[EndSession Element \(XMLA\)](#)

[Session Element \(XMLA\)](#)

Use SQL Server Profiler to Monitor Analysis Services

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

SQL Server Profiler tracks engine process events, such as the start of a batch or a transaction, and it captures data about those events, thus enabling you to monitor server and database activity (for example, user queries or login activity). You can capture SQL Server Profiler data to a SQL Server table or a file for later analysis, and you can also replay the events captured on the same or another Analysis Services instance to see exactly what happened. You can replay events in real time or on a step-by-step basis. It is also very useful to run the trace events along with the Performance counters on the same machine. The profiler can correlate these two based on time and display them together along a single timeline. Trace events will give you more details while Performance counters give you an aggregate view. For information about how to create and run traces, see [Create Profiler Traces for Replay \(Analysis Services\)](#).

The following table describes the topics in this section.

In This Section


TOPIC	DESCRIPTION
Introduction to Monitoring Analysis Services with SQL Server Profiler	Describes how to use SQL Server Profiler to monitor an Analysis Services instance.
Create Profiler Traces for Replay (Analysis Services)	Describes the requirements for creating a trace for replay by using SQL Server Profiler.
Analysis Services Trace Events	Describes Analysis Services event classes. These event classes map to actions generated by Analysis Services and are used for trace replays.

See Also

[Monitor an Analysis Services Instance](#)

Introduction to Monitoring Analysis Services with SQL Server Profiler

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

You can use SQL Server Profiler to monitor events generated by an instance of Microsoft SQL Server Analysis Services. By using SQL Server Profiler, you can do the following:

- Monitor the performance of an instance of Analysis Services.
- Debug Multidimensional Expressions (MDX) statements.
- Identify MDX statements that run slowly.
- Test MDX statements in the development phase of a project by stepping through statements to confirm that the code works as expected.
- Troubleshoot problems in Analysis Services by capturing events on a production system and replaying them on a test system. This approach is useful for testing or debugging purposes and lets users continue to use the production system without interference.
- Audit and review activity that occurred on an instance of Analysis Services. A security administrator can review any one of the audited events. This includes the success or failure of a login try and the success or failure of permissions in accessing statements and objects.
- Display data about the captured events to the screen, or capture and save data about each event to a file or SQL Server table for future analysis or playback. When you replay data, you can rerun the saved events as they originally occurred, either in real time or step by step.

Using SQL Server Profiler

To use SQL Server Profiler to create or replay traces, you must be a member of the Analysis Services server role. If you are a member of the Analysis Services server role, you can start SQL Server Profiler from the Microsoft SQL Server program group on the **Start** menu.

When you use SQL Server Profiler, note the following:

- Trace definitions are stored with the Analysis Services database by using the CREATE statement.
- Multiple traces can be running at the same time.
- Multiple connections can receive events from the same trace.
- A trace can continue when Analysis Services stops and restarts.

NOTE

Passwords are not revealed in trace events, but are replaced by ***** in the event.

For optimal performance, use SQL Server Profiler to monitor only those events in which you are most interested. Monitoring too many events adds overhead and can cause the trace file or table to grow very large, especially when you monitor over a long period of time. In addition, use filtering to limit the amount

of data that is collected and to prevent traces from becoming too large.

See Also

[Analysis Services Trace Events](#)

[Create Profiler Traces for Replay \(Analysis Services\)](#)

Create Profiler Traces for Replay (Analysis Services)

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

To replay queries, discovers, and commands that are submitted by users to Microsoft SQL Server Analysis Services, SQL Server Profiler must gather the required events. In order to initiate collection of these events, appropriate event classes must be selected in the **Event Selection** tab of the **Trace Properties** dialog box. For example if the Query Begin event class is selected, events that contain queries are collected and used for replay. Also, the trace file contains sufficient information to support replaying server transactions in a distributed environment in the original sequence of transactions.

Replay for Queries

To replay queries, SQL Server Profiler must capture the following events:

- Audit Login event class with all its data columns. This event class provides information about which user logged in and about the session settings. The server process ID (SPID) provides the reference to the user session. For more information, see [Security Audit Data Columns](#).
- Query Begin event class with all its data columns. This event class provides information about the query that was submitted to Analysis Services. The Event Subclass column provides information about the type of query. The TextData column provides the actual text of the query. The RequestParameters column provides the parameters for parameterized queries, and the RequestProperties column provides the properties of an XML for Analysis (XMLA) request. For more information, see [Queries Events Data Columns](#).
- Query End event class with all its data columns. This event class verifies the status of the query execution. For more information, see [Queries Events Data Columns](#).

Replay for Discovers

To replay discovers, SQL Server Profiler must capture the following events:

- Audit Login event class with all its data columns. This event class provides information about which user logged in and about the session settings. The SPID provides the reference to the user session. For more information, see [Security Audit Data Columns](#).
- Discover Begin event class with all its data columns. The TextData column provides the <RequestType> portion of the discover request, and the RequestProperties column provides the <Properties> portion of the discover request. The EventSubclass column provides the discover type. For more information, see [Discover Events Data Columns](#).
- Discover End event class with all its data columns. This event class verifies the status of the discover request. For more information, see [Discover Events Data Columns](#).

Replay for Commands

To replay commands, SQL Server Profiler must capture the following events:

- Command Begin event class with all its data columns. The TextData column provides the command details, such as the process type, database ID, and cube ID. The RequestParameters column provides the parameters for parameterized command, and the RequestProperties column provides the properties of an

XMLA request. For more information, see [Command Events Data Columns](#).

- Command End event class with all its data columns. This event class verifies the status of the command. For more information, see [Command Events Data Columns](#).



See Also

[Analysis Services Trace Events](#)

[Introduction to Monitoring Analysis Services with SQL Server Profiler](#)

Monitor Analysis Services with SQL Server Extended Events

5/16/2018 • 4 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Extended Events (*xEvents*) is a light-weight tracing and performance monitoring system that uses very few system resources, making it an ideal tool for diagnosing problems on both production and test servers. It's also highly scalable, configurable, and in SQL Server 2016, easier to use through new built-in tool support. In SQL Server Management Studio, on connections to Analysis Services instances, you can configure, run, and monitor a live trace, similar to using SQL Server Profiler. The addition of better tooling should make xEvents a more reasonable replacement for SQL Server Profiler and creates more symmetry in how you diagnose issues in your database engine and Analysis Services workloads.

Besides SQL Server Management Studio, you can also configure Analysis Services Extended Event sessions the old way, through XMLA scripting, as was supported in previous releases.

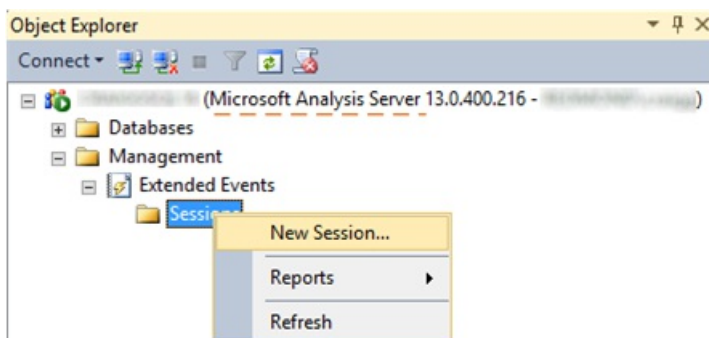
All Analysis Services events can be captured and targeted to specific consumers, as defined in [Extended Events](#).

NOTE

Watch this [quick video introduction](#) or read the [supporting blog post](#) to learn more about xEvents for Analysis Services in SQL Server 2016.

Use Management Studio to Configure Analysis Services

For both tabular and multidimensional instances, Management Studio provides a new Management folder that contains user-initiated xEvent sessions. You can run multiple sessions at once. However, in the current implementation, the Analysis Services Extended Events user interface does not support updating or replaying an existing session.



Choose Events

If you already know which events you want to capture, searching for them is the easiest way to add them to the trace. Otherwise, the following events are commonly used for monitoring operations:

- **CommandBegin** and **CommandEnd**.
- **QueryBegin**, **QueryEnd**, and **QuerySubcubeVerbose** (shows the entire MDX or DAX query sent to the server), plus **ResourceUsage** for stats on resources consumed by the query and how many rows are returned.

- **ProgressReportBegin** and **ProgressReportEnd** (for processing operations).
- **AuditLogin** and **AuditLogout** (captures the user identity under which a client application connects to Analysis Services).

Choose Data Storage

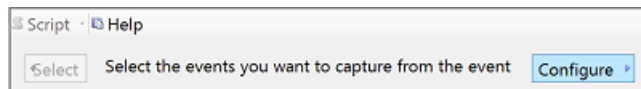
A session can be streamed live to a window in Management Studio or persisted to a file for subsequent analysis using Power Query or Excel.

- **event_file** stores session data in an .xel file.
- **event_stream** enables the **Watch Live Data** option in Management Studio.
- **ring_buffer** stores session data in memory for as long as the server is running. On a server restart, the session data is thrown out

Add Event Fields

Be sure to configure the session to include event fields so that you can easily see information of interest.

Configure is an option on the far side of the dialog box.



In configuration, on the Event Fields tab, select **TextData** so that this field appears adjacent to the event, showing return values, including queries that are executing on the server.

After you configure a session for the desired events and data storage, you can click the script button to send your configuration to one of supported destinations including a file, a new query in SQL Server Management Studio, and the clipboard.

Refresh Sessions

Once you create the session, be sure to refresh the Sessions folder in Management Studio to see the session you just created. If you configured an event_stream, you can right-click the session name and choose **Watch Live Data** to monitor server activity in real time.

XMLA Script to Start Extended Events in Analysis Services

Extended Event tracing is enabled using a similar XMLA create object script command as shown below:

```

<Execute ...>
  <Command>
    <Batch ...>
      <Create ...>
        <ObjectDefinition>
          <Trace>
            <ID>trace_id</ID>
            <Name>trace_name</Name>
            <ddl1300_300:XEvent>
              <event_session ...>
                <event package="AS" name="AS_event">
                  <action package="PACKAGE0" .../>
                </event>
                <target package="PACKAGE0" name="asynchronous_file_target">
                  <parameter name="filename" value="data_filename.xel"/>
                  <parameter name="metadatafile" value="metadata_filename.xem"/>
                </target>
              </event_session>
            </ddl1300_300:XEvent>
          </Trace>
        </ObjectDefinition>
      </Create>
    </Batch>
  </Command>
</Properties></Properties>
</Execute>

```

Where the following elements are to be defined by the user, depending on the tracing needs:

trace_id

Defines the unique identifier for this trace.

trace_name

The name given to this trace; usually a human readable definition of the trace. It is a common practice to use the *trace_id* value as the name.

AS_event

The Analysis Services event to be exposed. See [Analysis Services Trace Events](#) for names of the events.

data_filename

The name of the file that contains the events data. This name is suffixed with a time stamp to avoid data overwriting if the trace is sent over and over.

metadata_filename

The name of the file that contains the events metadata. This name is suffixed with a time stamp to avoid data overwriting if the trace is sent over and over.

XMLA Script to Stop Extended Events in Analysis Services

To stop the Extended Events tracing object you need to delete that object using a similar XMLA delete object script command as shown below:

```
<Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
  <Command>
    <Batch ...>
      <Delete ...>
        <Object>
          <TraceID>trace_id</TraceID>
        </Object>
      </Delete>
    </Batch>
  </Command>
  <Properties></Properties>
</Execute>
```

Where the following elements are to be defined by the user, depending on the tracing needs:

trace_id



Defines the unique identifier for the trace to be deleted.

See Also

[Extended Events](#)

Use Dynamic Management Views (DMVs) to Monitor Analysis Services

5/16/2018 • 7 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Analysis Services Dynamic Management Views (DMV) are query structures that expose information about local server operations and server health. The query structure is an interface to schema rowsets that return metadata and monitoring information about an Analysis Services instance.

For most DMV queries, you use a **SELECT** statement and the **\$System** schema with an XML/A schema rowset.

```
SELECT * FROM $System.<schemaRowset>
```

DMV queries return information about server state that is current at the time the query was run. To monitor operations in real time, use tracing instead. For more information, see [Use SQL Server Profiler to Monitor Analysis Services](#).

Benefits of Using DMV queries

DMV queries return information about operations and resource consumption that are not available through other means.

DMV queries are an alternative to running XML/A Discover commands. For most administrators, writing a DMV query is simpler because the query syntax is based on SQL. In addition, the result set is returned in a tabular format that is easier to read and copy from.

Examples and scenarios

A DMV query can help you answer questions about active sessions and connections, and which objects are consuming the most CPU or memory at a specific point in time. This section provides examples for scenarios where DMV queries are most commonly used. You can also review the [SQL Server 2008 R2 Analysis Services Operations Guide](#) for additional insights into using DMV queries to monitor a server instance.

```
Select * from $System.discover_object_activity
```

 /** This query reports on object activity since the service last started. For example queries based on this DMV, see [New System.Discover_Object_Activity](#).

```
Select * from $System.discover_object_memory_usage
```

 /** This query reports on memory consumption by object.

```
Select * from $System.discover_sessions
```

 /** This query reports on active sessions, including session user and duration.

```
Select * from $System.discover_locks
```

 /** This query returns a snapshot of the locks used at a specific point in time.

Query syntax

The query engine for DMVs is the Data Mining parser. The DMV query syntax is based on the [SELECT \(DMX\)](#) statement.

Although DMV query syntax is based on a SQL SELECT statement, it does not support the full syntax of a SELECT

statement. Notably, JOIN, GROUP BY, LIKE, CAST, and CONVERT are not supported.

```
SELECT [DISTINCT] [TOP <n>] <select list>
FROM $System.<schemaRowset>
[WHERE <condition expression>]
[ORDER BY <expression>[DESC|ASC]]
```

The following example for DISCOVER_CALC_DEPENDENCY illustrates the use of the WHERE clause for supplying a parameter to the query:

```
SELECT * FROM $System.DISCOVER_CALC_DEPENDENCY
WHERE OBJECT_TYPE = 'ACTIVE_RELATIONSHIP'
```

Alternatively, for schema rowsets that have restrictions, the query must include the SYSTEMRESTRICTSCHEMA function. The following example returns CSDL metadata about tabular models running on a tabular mode server. Note that CATALOG_NAME is case-sensitive:

```
Select * from SYSTEMRESTRICTSCHEMA ($System.Discover_cSDL_metadata, [CATALOG_NAME] = 'Adventure Works DW')
```

Tools and permissions

You must have system administrator permissions on the Analysis Services instance to query a DMV.

You can use any client application that supports MDX or DMX queries, including SQL Server Management Studio, a Reporting Services report, or a PerformancePoint Dashboard.

To run a DMV query from Management Studio, connect to the instance you want to query, click **New Query**. You can run a query from an MDX or a DMX query window.

DMV reference

Not all schema rowsets have a DMV interface. To return a list of all the schema rowsets that can be queried using DMV, run the following query.

```
SELECT * FROM $System.DBSchema_Tables
WHERE TABLE_TYPE = 'SCHEMA'
ORDER BY TABLE_NAME ASC
```

NOTE

If a DMV is not available for a given rowset, the server returns the following error: "The <schemarowset> request type was not recognized by the server". All other errors point to problems with the syntax.

ROWSET	DESCRIPTION
DBSCHEMA_CATALOGS Rowset	Returns a list of the Analysis Services databases on the current connection.
DBSCHEMA_COLUMNS Rowset	Returns a list of all the columns in the current database. You can use this list to construct a DMV query.

ROWSET	DESCRIPTION
DBSCHEMA_PROVIDER_TYPES Rowset	Returns properties about the base data types supported by the OLE DB data provider.
DBSCHEMA_TABLES Rowset	Returns a list of all the tables in the current database. You can use this list to construct a DMV query.
DISCOVER_CALC_DEPENDENCY Rowset	Returns a list of the columns and tables used in a model that have dependencies on other columns and tables.
DISCOVER_COMMAND_OBJECTS Rowset	Provides resource usage and activity information about objects in use by the referenced command.
DISCOVER_COMMANDS Rowset	Provides resource usage and activity information about currently executing command.
DISCOVER_CONNECTIONS Rowset	Provides resource usage and activity information about open connections to Analysis Services.
DISCOVER_CSDL_METADATA Rowset	Returns information about a tabular model. Requires the addition of SYSTEMRESTRICTSCHEMA and additional parameters.
DISCOVER_DB_CONNECTIONS Rowset	Provides resource usage and activity information about open connections from Analysis Services to external data sources, for example during processing or importing.
DISCOVER_DIMENSION_STAT Rowset	Returns the attributes in a dimension or columns in a table, depending on the model type.
DISCOVER_ENUMERATORS Rowset	Returns metadata about the enumerators supported for a specific data source.
DISCOVER_INSTANCES Rowset	Returns information about the specified instance. Requires the addition of SYSTEMRESTRICTSCHEMA and additional parameters.
DISCOVER_JOBS Rowset	Returns information about current jobs.
DISCOVER_KEYWORDS Rowset (XMLA)	Returns the list of reserved keywords.
DISCOVER_LITERALS Rowset	Returns the list of literals, including data types and values, supported by XMLA.
DISCOVER_LOCKS Rowset	Returns a snapshot of the locks used at a specific point in time.
DISCOVER_MEMORYGRANT Rowset	Returns information about memory allocated by Analysis Services at start up.
DISCOVER_MEMORYUSAGE Rowset	Shows memory usage by specific objects.
DISCOVER_OBJECT_ACTIVITY Rowset	Reports on object activity since the service last started.

ROWSET	DESCRIPTION
DISCOVER_OBJECT_MEMORY_USAGE Rowset	Reports on memory consumption by object.
DISCOVER_PARTITION_DIMENSION_STAT Rowset	Provides information about the attributes in a dimension. Requires the addition of SYSTEMRESTRICTSCHEMA and additional parameters.
DISCOVER_PARTITION_STAT Rowset	Provides information about the partitions in a dimension, table, or measure group. Requires the addition of SYSTEMRESTRICTSCHEMA and additional parameters.
DISCOVER_PERFORMANCE_COUNTERS Rowset	Lists the columns used in a performance counter. Requires the addition of SYSTEMRESTRICTSCHEMA and additional parameters.
DISCOVER_PROPERTIES Rowset	Returns information about properties supported by XMLA for the specified data source.
DISCOVER_SCHEMA_ROWSETS Rowset	Returns names, restrictions, description and other information for all enumeration values supported by XMLA.
DISCOVER_SESSIONS Rowset	Reports on active sessions, including session user and duration.
DISCOVER_STORAGE_TABLE_COLUMN_SEGMENTS Rowset	Provides information at the column and segment level about storage tables used by an Analysis Services database running in Tabular or SharePoint mode.
DISCOVER_STORAGE_TABLE_COLUMNS Rowset	Allows the client to determine the assignment of columns to storage tables used by an Analysis Services database running in Tabular or SharePoint mode.
DISCOVER_STORAGE_TABLES Rowset	Returns information about the tables used for storage of models in a Tabular model database.
DISCOVER_TRACE_COLUMNS Rowset	Returns an XML description of the columns available in a trace.
DISCOVER_TRACE_DEFINITION_PROVIDERINFO Rowset	Returns name and version information of the provider.
DISCOVER_TRACE_EVENT_CATEGORIES Rowset	Returns a list of available categories.
DISCOVER_TRACES Rowset	Returns a list of traces actively running on the current connection.
DISCOVER_TRANSACTIONS Rowset	Returns a list of transactions actively running on the current connection.
DISCOVER_XEVENT_TRACE_DEFINITION Rowset	Returns a list of xevent traces actively running on the current connection.

ROWSET	DESCRIPTION
DMSHEMA_MINING_COLUMNS Rowset	Lists the individual columns of all mining models available on the current connection.
DMSHEMA_MINING_FUNCTIONS Rowset	Returns a list of functions supported by the data mining algorithms on the server.
DMSHEMA_MINING_MODEL_CONTENT Rowset	Returns a rowset consisting of columns that describe the current model.
DMSHEMA_MINING_MODEL_CONTENT_PMML Rowset	Returns a rowset consisting of columns that describe the current model in PMML format.
DMSHEMA_MINING_MODEL_XML Rowset	Returns a rowset consisting of columns that describe the current model in PMML format.
DMSHEMA_MINING_MODELS Rowset	Returns a list of the mining models in the current database.
DMSHEMA_MINING_SERVICE_PARAMETERS Rowset	Returns a list of the parameters for the algorithms on the server.
DMSHEMA_MINING_SERVICES Rowset	Provides a list of the data mining algorithms available on the server.
DMSHEMA_MINING_STRUCTURE_COLUMNS Rowset	Returns a list of all of the columns from all of the mining models available in the current connection.
DMSHEMA_MINING_STRUCTURES Rowset	Lists the mining structures available in the current connection.
MDSHEMA_CUBES Rowset	Returns information about the cubes that are defined in the current database.
MDSHEMA_DIMENSIONS Rowset	Returns information about the dimensions that are defined in the current database.
MDSHEMA_FUNCTIONS Rowset	Returns a list of functions available to client applications connected to the database.
MDSHEMA_HIERARCHIES Rowset	Returns information about the hierarchies that are defined in the current database.
MDSHEMA_INPUT_DATASOURCES Rowset	Returns information about the data source objects that are defined in the current database.
MDSHEMA_KPIS Rowset	Returns information about the KPIs that are defined in the current database.
MDSHEMA_LEVELS Rowset	Returns information about the levels within the hierarchies that are defined in the current database.
MDSHEMA_MEASUREGROUP_DIMENSIONS Rowset	Lists the dimension of measure groups.
MDSHEMA_MEASUREGROUPS Rowset	Returns a list of measure groups in the current connection.

ROWSET	DESCRIPTION
MDSHEMA_MEASURES Rowset	Returns a list of measures in the current connection.
MDSHEMA_MEMBERS Rowset	Returns a list of all members in the current connection, listed by database, cube, and dimension.
MDSHEMA_PROPERTIES Rowset	Returns a fully qualified name of each property, along with property type, data type, and other metadata.
MDSHEMA_SETS Rowset	Returns a list of set that are defined in the current connection.

See also

[New System.Discover_Object_Activity](#)

[New SYSTEMRESTRICTEDSCHEMA Function for Restricted Rowsets and DMVs](#)

Performance Counters (SSAS)

5/16/2018 • 17 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Using Performance Monitor, you can monitor the performance of a Microsoft SQL Server Analysis Services (SSAS) instance by using performance counters.

Performance Monitor is a Microsoft Management Control (MMC) snap-in that tracks resource usage. You can start this MMC snap-in by typing in **PerfMon** at the command prompt or from Control Panel by clicking **Administrative Tools**, then **Performance Monitor**. Performance Monitor lets you track server and process performance and activity by using predefined objects and counters, and monitor events by using user-defined counters. Performance Monitor collects counts instead of data about the events, for example, memory usage, number of active transactions, or CPU activity. You can also set thresholds on specific counters to generate alerts that notify operators.

Performance Monitor can monitor remote and local instances of Analysis Services or SQL Server. For more information, see [Using Performance Monitor](#).

To see the description of any counter that can be used with SQL Server Analysis Services, in Performance Monitor, open the **Add Counters** dialog box, select a performance object, and then click **Show Description**. The most important counters are CPU usage, memory usage, disk IO rate. It is recommended you start with these important counters, and move to more detailed counters when you have a better idea of what else could be improved through monitoring. For more information about which counters to include, see the [SQL Server 2008 R2 Operations Guide](#).

Counters are grouped so you can more easily find related counters.

Counters by Groups

GROUP	DESCRIPTION
Cache	Statistics related to the Analysis Services aggregation cache.
Connection	Statistics related to Microsoft Analysis Services connections.
Data Mining Prediction	Statistics related to processing data mining models processing.
Data Mining Model Processing	Statistics related to creating predictions from data mining models.
Locks	Statistics related to Microsoft Analysis Services internal server locks.
MDX	Statistics related to Microsoft Analysis Services MDX calculations.
Memory	Statistics related to Microsoft Analysis Services internal server memory.

GROUP	DESCRIPTION
Proactive Caching	Statistics related to Microsoft Analysis Services Proactive Caching.
Processing Aggregations	Statistics related to processing of aggregations in MOLAP data files.
Processing Indexes	Statistics related to processing of indexes for MOLAP data files.
Processing	Statistics related to processing of data.
Storage Engine Query	Statistics related to Microsoft Analysis Services storage engine queries.
Threads	Statistics related to Microsoft Analysis Services threads.

Cache

Statistics related to the Microsoft Analysis Services aggregation cache.

COUNTER	DESCRIPTION
Current KB	Current memory used by the aggregation cache, in KB.
KB added/sec	Rate of memory added to the cache, KB/sec.
Current entries	Current number of cache entries.
Inserts/sec	Rate of insertions into the cache. The rate is tracked per partition per cube per database.
Evictions/sec	Rate of evictions from the cache. This is per partition per cube per database. Evictions are typically due to background cleaner.
Total inserts	Insertions into the cache. The rate is tracked per partition per cube per database.
Total evictions	Evictions from the cache. Evictions are tracked per partition per cube per database. Evictions are typically due to background cleaner.
Direct hits/sec	Rate of cache direct hits. A cache hit indicates that queries were answered from an existing cache entry.
Misses/sec	Rate of cache misses.
Lookups/sec	Rate of cache lookups.
Total direct hits	Total count of direct cache hits. A direct cache hit indicates that queries were answered from existing cache entries.
Total misses	Total count of cache misses.

COUNTER	DESCRIPTION
Total lookups	Total number of lookups into the cache.
Direct hit ratio	Ratio of cache direct hits to cache lookups, for the period between counter values.
Total filtered iterator cache hits	Total number of cache hits that returned an indexed iterator over the filtered results.
Total filtered iterator cache misses	Total number of cache hits that were unable to build an indexed iterator over the filtered results and had to build a new cache with the filtered results.

Connection

Statistics related to Microsoft Analysis Services connections.

COUNTER	DESCRIPTION
Current connections	Current number of client connections established.
Requests/sec	Rate of connection requests. These are arrivals.
Total requests	Total connection requests. These are arrivals.
Successes/sec	Rate of successful connection completions.
Total successes	Total successful connections.
Failures/sec	Rate of connection failures.
Total failures	Total failed connection attempts.
Current user sessions	Current number of user sessions established.

Data Mining Model Processing

Statistics related to Microsoft Analysis Services Data Mining model processing.

COUNTER	DESCRIPTION
Cases/sec	Rate at which cases are processed.
Current models processing	Current number of models being processed.

Data Mining Prediction

Statistics related to Microsoft Analysis Services Data Mining prediction.

COUNTER	DESCRIPTION
Concurrent DM queries	Current number of data mining queries being actively worked on.

COUNTER	DESCRIPTION
Predictions/sec	Number of predictions generated in data mining queries
Rows/sec	Number of rows handled during a data mining prediction query.
Queries/sec	Number of data mining queries that were handled.
Total Queries	Total data mining queries received by the server.
Total Rows	Total rows returned by data mining queries.
Total Predictions	Total data mining prediction queries received by the server.

Locks

Statistics related to Microsoft Analysis Services internal server locks.

COUNTER	DESCRIPTION
Current latch waits	Current number of threads waiting for a latch. These are latch requests that could not be given immediate grants and are in a wait state.
Latch waits/sec	Rate of latch requests that could not be granted immediately and had to wait before being granted.
Current locks	Current number of locked objects.
Current lock waits	Current number of clients waiting for a lock.
Lock requests/sec	Number of lock requests per second.
Lock grants/sec	Number of lock grants per second.
Lock waits/sec	Number of lock waits per second. These are lock requests that could not be given immediate lock grants and were put in a wait state.
Lock denials/sec	Rate of lock denials.
Unlock requests/sec	Number of unlock requests per second.
Total deadlocks detected	Total number of deadlocks detected.

MDX

Statistics related to Microsoft Analysis Services MDX Calculations.

COUNTER	DESCRIPTION
Number of calculation covers	Total number of evaluation nodes built by MDX execution plans, including active and cached.

COUNTER	DESCRIPTION
Current number of evaluation nodes	Current (approximate) number of evaluation nodes built by MDX execution plans, including active and cached.
Number of Storage Engine evaluation nodes	Total number of Storage Engine evaluation nodes built by MDX execution plans.
Number of cell-by-cell evaluation nodes	Total number of cell-by-cell evaluation nodes built by MDX execution plans.
Number of bulk-mode evaluation nodes	Total number of bulk-mode evaluation nodes built by MDX execution plans.
Number of evaluation nodes that covered a single cell	Total number of evaluation nodes built by MDX execution plans that covered only one cell.
Number of evaluation nodes with calculations at the same granularity	Total number of evaluation nodes built by MDX execution plans for which the calculations were at the same granularity as the evaluation node.
Current number of cached evaluation nodes	Current (approximate) number of cached evaluation nodes built by MDX execution plans.
Number of cached Storage Engine evaluation nodes	Total number of cached Storage Engine evaluation nodes built by MDX execution plans
Number of cached bulk-mode evaluation nodes	Total number of cached bulk-mode evaluation nodes built by MDX execution plans.
Number of cached 'other' evaluation nodes	Total number of cached evaluation nodes built by MDX execution plans that are neither Storage Engine nor Bulk-mode.
Number of evictions of evaluation nodes	Total number of cache evictions of evaluation nodes due to collisions.
Number of hash index hits in the cache of evaluation nodes	Total number of hits in the cache of evaluation nodes that were satisfied by the hash index.
Number of cell-by-cell hits in the cache of evaluation nodes	Total number of cell-by-cell hits in the cache of evaluation nodes.
Number of cell-by-cell misses in the cache of evaluation nodes	Total number of cell-by-cell misses in the cache of evaluation nodes.
Number of subcube hits in the cache of evaluation nodes	Total number of subcube hits in the cache of evaluation nodes.
Number of subcube misses in the cache of evaluation nodes	Total number of subcube misses in the cache of evaluation nodes.
Total Sonar subcubes	Total number of subcubes that the query optimizer generated.
Total cells calculated	Total number of cell properties calculated.

COUNTER	DESCRIPTION
Total recomputes	Total number of cells recomputed due to error.
Total flat cache inserts	Total number of cell values inserted into flat calculation cache.
Total calculation cache registered	Total number of calculation caches registered.
Total NON EMPTY	Total number of times a NON EMPTY algorithm was used.
Total NON EMPTY unoptimized	Total number of times an unoptimized NON EMPTY algorithm was used.
Total NON EMPTY for calculated members	Total number of times a NON EMPTY algorithm looped over calculated members.
Total Autoexist	Total number of times Autoexist was performed.
Total EXISTING	Total number of times an EXISTING set operation was performed.

Memory

Statistics related to Microsoft Analysis Services internal server memory.

COUNTER	DESCRIPTION
Page Pool 64 Alloc KB	Memory borrowed from system, in KB. This memory is given away to other parts of the server.
Page Pool 64 Lookaside KB	Current memory in 64KB lookaside list, in KB. (Memory pages ready to be used.)
Page Pool 8 Alloc KB	Memory borrowed from 64KB page pool, in KB. This memory is given away to other parts of the server.
Page Pool 8 Lookaside KB	Current memory in 8KB lookaside list, in KB. (Memory pages ready to be used.)
Page Pool 1 Alloc KB	Memory borrowed from 64KB page pool, in KB. This memory is given away to other parts of the server.
Page Pool 1 Lookaside KB	Current memory in 8KB lookaside list, in KB. (Memory pages ready to be used.)
Cleaner Current Price	Current price of memory, \$/byte/time, normalized to 1000.
Cleaner Balance/sec	Rate of balance+shrink operations.
Cleaner Memory shrunk KB/sec	Rate of shrinking, in KB/sec.
Cleaner Memory shrinkable KB	Amount of memory, in KB, subject to purging by the background cleaner.

COUNTER	DESCRIPTION
Cleaner Memory nonshrinkable KB	Amount of memory, in KB, not subject to purging by the background cleaner.
Cleaner Memory KB	Amount of memory, in KB, known to the background cleaner. (Cleaner memory shrinkable + Cleaner memory nonshrinkable.)
Memory Usage KB	Memory usage of the server process as used in calculating cleaner memory price. Equal to counter Process\PrivateBytes plus the size of memory-mapped data, ignoring any memory which was mapped or allocated by the xVelocity in-memory analytics engine (VertiPaq) in excess of the xVelocity engine Memory Limit.
Memory Limit Low KB	Low memory limit, from configuration file.
Memory Limit High KB	High memory limit, from configuration file.
AggCacheKB	Current memory allocated to aggregation cache, in KB.
Quota KB	Current memory quota, in KB. Memory quota is also known as a memory grant or memory reservation.
Quota Blocked	Current number of quota requests that are blocked until other memory quotas are freed.
Filestore KB	Current memory allocated to filestore (file cache), in KB.
Filestore Page Faults/sec	Filestore page fault rate.
Filestore Reads/sec	Filestore pages read/sec.
Filestore KB Reads/sec	Filestore KB read/sec.
Filestore Writes/sec	Filestore pages written/sec. The writes are asynchronous.
Filestore KB Write/sec	Filestore KB written/sec. The writes are asynchronous.
Filestore IO Errors/sec	Filestore IO Error rate.
Filestore IO Errors	Filestore IO Errors total.
Filestore Clock Pages Examined/sec	Rate of background cleaner examining pages for eviction consideration.
Filestore Clock Pages HaveRef/sec	Rate of background cleaner examining pages that have a current reference count (are currently in use).
Filestore Clock Pages Valid/sec	Rate of background cleaner examining pages that are valid candidates for eviction.
Filestore Memory Pinned KB	Current filestore memory pinned, in KB.

COUNTER	DESCRIPTION
In-memory Dimension Property File KB	Current size of in-memory dimension property file, in KB.
In-memory Dimension Property File KB/sec	Rate of writes to in-memory dimension property file, in KB.
Potential In-memory Dimension Property File KB	Potential size of in-memory dimension property file, in KB.
Dimension Property Files	Number of dimension property files.
In-memory Dimension Index (Hash) File KB	Size of current in-memory dimension index (hash) file, in KB.
In-memory Dimension Index (Hash) File KB/sec	Rate of writes to in-memory dimension index (hash) file, in KB.
Potential In-memory Dimension Index (Hash) File KB	Potential size of in-memory dimension index (hash) file, in KB.
Dimension Index (Hash) Files	Number of dimension index (hash) files.
In-memory Dimension String File KB	Current size of in-memory dimension string file, in KB.
In-memory Dimension String File KB/sec	Rate of writes to in-memory dimension string file, in KB.
Potential In-memory Dimension String File KB	Potential size of in-memory dimension string file, in KB.
Dimension String Files	Number of dimension string files.
In-memory Map File KB	Current size of in-memory map file, in KB.
In-memory Map File KB/sec	Rate of writes to in-memory map file, in KB.
Potential In-memory Map File KB	Potential size of in-memory map file, in KB.
Map Files	Number of map files.
In-memory Aggregation Map File KB	Current size of in-memory aggregation map file, in KB.
In-memory Aggregation Map File KB/sec	Rate of writes to in-memory aggregation map file, in KB.
Potential In-memory Aggregation Map File KB	Size of potential in-memory aggregation map file, in KB.
Aggregation Map Files	Number of aggregation map files.
In-memory Fact Data File KB	Size of current in-memory fact data file, in KB.
In-memory Fact Data File KB/sec	Rates of writes to in-memory fact data file KB rate.
Potential In-memory Fact Data File KB	Size of potential in-memory fact data file, in KB.
Fact Data Files	Number of fact data files.
In-memory Fact String File KB	Size of current in-memory fact string file, in KB.

COUNTER	DESCRIPTION
In-memory Fact String File KB/sec	Rate of writes to in-memory fact string file, in KB.
Potential In-memory Fact String File KB	Size of potential in-memory fact string file, in KB.
Fact String Files	Number of fact string files.
In-memory Fact Aggregation File KB	Current size of in-memory fact aggregation file, in KB.
In-memory Fact Aggregation File KB/sec	Rate of writes to in-memory fact aggregation file, in KB.
Potential In-memory Fact Aggregation File KB	Size of potential in-memory fact aggregation file, in KB.
Fact Aggregation Files	Number of fact aggregation files.
In-memory Other File KB	Size of current in-memory other file, in KB.
In-memory Other File KB/sec	Rate of writes to in-memory other file, in KB.
Potential In-memory Other File KB	Size of potential in-memory other file, in KB.
Other Files	Number of other files.
VertiPaq Paged KB	Kilobytes of paged memory in use for in-memory data.
VertiPaq Nonpaged KB	Kilobytes of memory locked in the working set for use by the in-memory engine.
VertiPaq Memory-Mapped KB	Kilobytes of pageable memory in use for in-memory data.
Memory Limit Hard KB	Hard memory limit, from configuration file.
Memory Limit VertiPaq KB	In-memory limit, from configuration file.

Proactive Caching

Statistics related to Microsoft Analysis Services Proactive Caching.

COUNTER	DESCRIPTION
Notifications/sec	Rate of notifications from relational database.
Processing Cancellations/sec	Rate of processing cancellations caused by notifications.
Proactive Caching Begin/sec	Rate of proactive caching begin.
Proactive Caching Completion/sec	Rate of proactive caching completion.

Processing Aggregations

Statistics related to Microsoft Analysis Services processing of aggregations in MOLAP data files.

COUNTER	DESCRIPTION
Current partitions	Current number of partitions being processed.
Total partitions	Total number of partitions processed (successfully or otherwise).
Memory size rows	Size of current aggregations in memory. This count is an estimate.
Memory size bytes	Size of current aggregations in memory. This count is an estimate.
Rows merged/sec	Rate of rows merged or inserted into an aggregation.
Rows created/sec	Rate of aggregation rows created.
Temp file rows written/sec	Rate of writing rows to a temporary file. Temporary files are written when aggregations exceed memory limits.
Temp file bytes written/sec	Rate of writing bytes to a temporary file. Temporary files are written when aggregations exceed memory limits.

Processing Indexes

Statistics related to Microsoft Analysis Services processing of indexes for MOLAP data files.

COUNTER	DESCRIPTION
Current partitions	Current number of partitions being processed.
Total partitions	Total number of partitions processed (successfully or otherwise).
Rows/sec	Rate of rows from MOLAP files used to create indexes.
Total rows	Total rows from MOLAP files used to create indexes.

Processing

Statistics related to Microsoft Analysis Services processing of data.

COUNTER	DESCRIPTION
Rows read/sec	Rate of rows read from all relational databases.
Total rows read	Count of rows read from all relational databases.
Rows converted/sec	Rate of rows converted during processing.
Total rows converted	Count of rows converted during processing.
Rows written/sec	Rate of rows written during processing.
Total rows written	Count of rows written during processing.

Storage Engine Query

Statistics related to Microsoft Analysis Services storage engine queries.

COUNTER	DESCRIPTION
Current measure group queries	Current number of measure group queries being actively worked on.
Measure group queries/sec	Rate of measure group queries
Total measure group queries	Total number of queries to measure group.
Current dimension queries	Current number of dimension queries being actively worked on.
Dimension queries/sec	Rate of dimension queries
Total dimension queries.	Total number of dimension queries.
Queries answered/sec	Rate of queries answered.
Total queries answered	Total number of queries answered.
Bytes sent/sec	Rate of bytes sent by server to clients, in response to queries.
Total bytes sent	Total bytes sent by server to clients, in response to queries.
Rows sent/sec	Rate of rows sent by server to clients.
Total rows sent	Total rows sent by server to clients.
Queries from cache direct/sec	Rate of queries answered from cache directly.
Queries from cache filtered/sec	Rate of queries answered by filtering existing cache entry.
Queries from file/sec	Rate of queries answered from files.
Total queries from cache direct	Total number of queries derived directly from cache. Note that this is per partition.
Total queries from cache filtered	Total queries answered by filtering existing cache entries.
Total queries from file	Total number of queries answered from files.
Map reads/sec	Number of logical read operations using the Map file.
Map bytes/sec	Bytes read from the Map file.
Data reads/sec	Number of logical read operations using the Data file.
Data bytes/sec	Bytes read from the Data file.

COUNTER	DESCRIPTION
Avg time/query	Average time per query, in milliseconds. Response time based on queries answered since the last counter measurement.
Network round trips/sec	Rate of network round trips. This includes all client/server communication.
Total network round trips	Total network round trips. This includes all client/server communication.
Flat cache lookups/sec	Rate of flat cache lookups. This includes global, session, and query scope flat caches.
Flat cache hits/sec	Rate of flat cache hits. This includes global, session, and query scope flat caches.
Calculation cache lookups/sec	Rate of calculation cache lookups. This includes global, session, and query scope calculation caches.
Calculation cache hits/sec	Rate of calculation cache hits. This includes global, session, and query scope calculation caches.
Persisted cache lookups/sec	Rate of persisted cache lookups. Persisted caches are created by the MDX script CACHE statement.
Persisted cache hits/sec	Rate of persisted cache hits. Persisted caches are created by the MDX script CACHE statement.
Dimension cache lookups/sec	Rate of dimension cache lookups.
Dimension cache hits/sec	Rate of dimension cache hits.
Measure group cache lookups/sec	Rate of measure group cache lookups.
Measure group cache hits/sec	Rate of measure group cache hits.
Aggregation lookups/sec	Rate of aggregation lookups.
Aggregation hits/sec	Rate of aggregation hits.

Threads


Statistics related to Microsoft Analysis Services threads.

COUNTER	DESCRIPTION
Short parsing idle threads	Number of idle threads in the short parsing thread pool.
Short parsing busy threads	Number of busy threads in the short parsing thread pool.
Short parsing job queue length	Number of jobs in the queue of the short parsing thread pool.
Short parsing job rate	Rate of jobs through the short parsing thread pool.

COUNTER	DESCRIPTION
Long parsing idle threads	Number of idle threads in the long parsing thread pool.
Long parsing busy threads	Number of busy threads in the long parsing thread pool.
Long parsing job queue length	Number of jobs in the queue of the long parsing thread pool.
Long parsing job rate	Rate of jobs through the long parsing thread pool.
Query pool idle threads	Number of idle threads in the query thread pool.
Query pool busy threads	Number of busy threads in the query thread pool.
Query pool job queue length	Number of jobs in the queue of the query thread pool.
Query pool job rate	Rate of jobs through the query thread pool.
Processing pool idle non-I/O threads	Number of idle threads in the processing thread pool dedicated to non-I/O jobs.
Processing pool busy non-I/O threads	Number of threads running non-I/O jobs in the processing thread pool.
Processing pool job queue length	Number of non-I/O jobs in the queue of the processing thread pool.
Processing pool job rate	Rate of non-I/O jobs through the processing thread pool.
Processing pool idle I/O job threads	Number of idle threads for I/O jobs in the processing thread pool.
Processing pool busy I/O job threads	Number of threads running I/O jobs in the processing thread pool.
Processing pool I/O job queue length	Number of I/O jobs in the queue of the processing thread pool.
Processing pool I/O job completion rate	Rate of I/O jobs through the processing thread pool.

Clear the Analysis Services Caches

5/16/2018 • 4 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Analysis Services caches data to boost query performance. This topic provides recommendations for using the XMLA ClearCache command to clear caches that were created in response to an MDX query. The effects of running ClearCache vary depending on whether you are using a tabular or multidimensional model.

When to clear the cache for multidimensional models

For multidimensional databases, Analysis Services builds caches in the formula engine when evaluating a calculation, and in the storage engine for the results of dimension queries and measure group queries. Measure group queries occur when the formula engine needs measure data for a cell coordinate or subcube. Dimension queries occur when querying unnatural hierarchies and when applying autoexists.

Clearing the cache is recommended when conducting performance testing. By clearing the cache between test runs, you ensure that caching does not skew any test results that measure the impact of a query design change.

When to clear the cache for tabular models

Tabular models are generally stored in memory, where aggregations and other calculations are performed at the time a query is executed. As such, the ClearCache command has a limited effect on tabular models. For a tabular model, data may be added to the Analysis Services caches if MDX queries are run against it. In particular, DAX measures referenced by MDX and autoexists operations can cache results in the formula cache and dimension cache respectively. Note however, that unnatural hierarchies and measure group queries do not cache results in the storage engine. Additionally, it is important to recognize that DAX queries do not cache results in the formula and storage engine. To the extent that caches exist as a result of MDX queries, running ClearCache against a tabular model will invalidate any cached data from the system.

Running ClearCache will also clear in-memory caches in the xVelocity in-memory analytics engine (VertiPaq). The xVelocity engine maintains a small set of cached results. Running ClearCache will invalidate these caches in the xVelocity engine.

Finally, running ClearCache will also remove residual data that is left in memory when a tabular model is reconfigured for **DirectQuery** mode. This is particularly important if the model contains sensitive data that is subject to tight controls. In this case, running ClearCache is a precautionary action that you can take to ensure that sensitive data exists only where you expect it to be. Clearing the cache manually is necessary if you are using Management Studio to deploy the model and change the query mode. In contrast, using SQL Server Data Tools to specify **DirectQuery** on the model and partitions will automatically clear the cache when you switch the model to use that query mode.

Compared with recommendations for clearing multidimensional model caches during performance testing, there is no broad recommendation for clearing tabular model caches. If you are not managing the deployment of a tabular model that contains sensitive data, there is no specific administrative task that calls for clearing the cache.

Clear the cache for Analysis Services models

To clear the cache, use XMLA and SQL Server Management Studio. You can clear the cache at the database, cube, dimension or table, or measure group level. The following steps for clearing the cache at the database level apply to both multidimensional models and tabular models.

NOTE

Rigorous performance testing might require a more comprehensive approach to clearing the cache. For instructions on how to flush Analysis Services and file system caches, see the section on clearing caches in the [SQL Server 2008 R2 Analysis Services Operations Guide](#).

For both multidimensional and tabular models, clearing some of these caches can be a two-step process that consists of invalidating the cache when **ClearCache** executes, followed by emptying the cache when the next query is received. A reduction in memory consumption will be evident only after the cache is actually emptied.

Clearing the cache requires that you provide an object identifier to the **ClearCache** statement in an XMLA query. The first step in this topic explains how to get an object identifier.

Step 1: Get the object identifier

1. In Management Studio, right-click an object, select **Properties**, and copy the value from the ID property in the **Properties** pane. This approach works for the database, cube, dimension, or table.
2. To get the measure group ID, right-click the measure group and select **Script Measure Group As**. Choose either **Create** or **Alter**, and send the query to a window. The ID of the measure group will be visible in the object definition. Copy the ID of the object definition.

Step 2: Run the query

1. In Management Studio, right-click a database, point to **New Query**, and select **XMLA**.
2. Copy the following code example into the XMLA query window. Change **DatabaseID** to the ID of the database on the current connection.

```
<ClearCache xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <Object>
    <DatabaseID> Adventure Works DW Multidimensional</DatabaseID>
  </Object>
</ClearCache>
```

Alternatively, you can specify a path of a child object, such as a measure group, to clear the cache for just that object.

```
<ClearCache xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <Object>
    <DatabaseID>Adventure Works DW Multidimensional</DatabaseID>
    <CubeID>Adventure Works</CubeID>
    <MeasureGroupID>Fact Currency Rate</MeasureGroupID>
  </Object>
</ClearCache>
```

3. Press F5 to execute the query. You should see the following result:



```
<return xmlns="urn:schemas-microsoft-com:xml-analysis">
  <root xmlns="urn:schemas-microsoft-com:xml-analysis:empty" />
</return>
```

See Also

[Monitor an Analysis Services Instance](#)

Create Analysis Services Scripts in Management Studio

5/16/2018 • 3 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

SQL Server Management Studio includes script generation features, templates, and editors that you can use to script Analysis Services objects and tasks.

Script Analysis Services Tasks in Management Studio

Scripting tasks in SQL Server Management Studio is accomplished by clicking one of the Script options in a task-oriented dialog box. All of the dialog boxes that you use to perform tasks such as backup or restore database, process an object, or design an aggregation, include a Script option at the top of the dialog box. Selecting one of these options generates an XMLA script based on the information and settings in the dialog box.

By default, the script is generated and placed in an XMLA query editor, but you can also expand the Script option list to direct the script to the Windows Clipboard or a file.

To script an Analysis Services task

1. In SQL Server Management Studio, connect to an instance of Analysis Services.
2. Right-click a database and click **Backup**. This opens the Backup Database dialog box. Specify a backup file name and choose the options you want for this backup.
3. Click **Script** at the top of the dialog box. The Script feature is part of all task-based dialog boxes in Management Studio. It has the following options: **Script Action to New Query Window** to open the query editor window, **Script Action to File** to save the XMLA script to a file, or **Script Action to Clipboard** to save the XMLA script to the Clipboard.

Note that the **Script Action to Job** option that is listed as a script option in Management Studio is not supported for Analysis Services scripts.

4. If you select the default option, **Script Action to New Query Window**, a generated script is placed in an XMLA query window.

You can now close the Backup Database dialog box and edit or run the XMLA script directly.

Script Analysis Services Objects in Management Studio

Scripting objects in SQL Server Management Studio is accomplished by right-clicking an Analysis Services object in SQL Server Management Studio and selecting either **Create to**, **Alter to**, or **Delete to**. Each of these options can be directed to a window or a file, but regardless of where the script is directed to, it will come in the form of a DDL script in an XMLA wrapper. One great advantage to such scripts is that they can be run against any server you point them at. Also, names in the scripts can be changed and run on an iterative basis for mass construction, alteration, or deletion of objects.

Objects that you can script include the elements of an Analysis Services database, including data sources, data source views, cubes, dimensions, mining structures, and roles.

Prerequisites include an understanding of XML for Analysis (XMLA). Fortunately, SQL Server Management Studio has a feature that automatically creates the XMLA script required to create objects, such as cubes. This automation

feature helps reduce the learning curve for XMLA. For more information about how to use XMLA, see [Developing with XMLA in Analysis Services](#). For more information about how to use XMLA, see [Developing with XMLA in Analysis Services](#).

IMPORTANT

When scripting the Role Object, be aware that security permissions are contained by the objects they secure rather than with the security role with which they are associated.

To script Analysis Services objects

1. In SQL Server Management Studio, connect to an instance of Analysis Services.
2. Locate the object for which you want to create a script that either creates, alters, or deletes objects.
3. Right-click the object, point to **Script Cube as**, point to **CREATE To**, **ALTER To**, or **Delete To**, and then click one of the following options: **New Query Editor Window** to open the query editor window, **File** to save the XMLA script to a file, or **Clipboard** to save the XMLA script to the Clipboard.

NOTE


Typically, you would select **File** if you want to create multiple different versions of the file.

See Also

[XMLA Query Editor \(Analysis Services - Multidimensional Data\)](#)

Use Analysis Services Templates in SQL Server Management Studio

5/16/2018 • 13 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

SQL Server Management Studio provides a set of templates to help you quickly create XMLA scripts, DMX or MDX queries, create KPIs in a cube or tabular model, script backup and restore operations, and perform many other tasks. Templates are located in the **Template Explorer** in Management Studio.

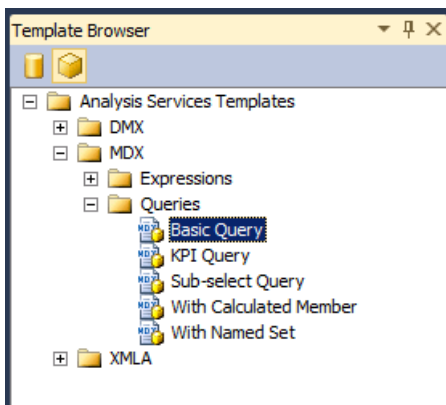
This topic includes a list of the templates for multidimensional models and tabular models, and provides examples of how to build an MDX query and XMLA statement by using the Metadata Explorer and the Template Explorer.

This topic does not cover DMX templates. For examples of how to create data mining queries using the templates, see [Create a DMX Query in SQL Server Management Studio](#) or [Create a Singleton Prediction Query from a Template](#).

Open an Analysis Services Template

All templates for database engine queries and Analysis Services queries and commands are available in Template Explorer.

To open **Template Explorer**, select it from the **View** menu. Next, click the cube icon to see a list of the templates that are available for Analysis Services.



To open a template, right-click the template name and select **Open**, or drag the template into a query window that you already opened. After the query window is open, you can use commands on the toolbar or Query menu to help you build statements:

- To check the syntax of a query, click **Parse**.
- To run a query, click **Execute**.

To stop a query that is running, click **Cancel Executing Query**.

- View the results of a query in the **Results** tab at the bottom of the screen.

Switch to the **Messages** tab to see the number of records returned, errors, query statements, and any other messages that are associated with the execution of the query. For example, if you execute a DAX statement against a model running in Direct Query mode, you can see the Transact-SQL statement that is generated by the xVelocity in-memory analytics engine (VertiPaq).

Build and Run an MDX Query on a Tabular Model using a Template

This example shows you how to create an MDX query in SQL Server Management Studio, using a tabular model database as the data source. To repeat this example on your computer, you can [download the Adventureworks tabular model sample project](#).

WARNING

You cannot use MDX queries against tabular models that have been deployed in Direct Query mode. You can, however, send equivalent queries by using the DAX table queries with the EVALUATE command. For more information, see [DAX Query Parameters](#).

Create an MDX query from a template

1. In SQL Server Management Studio, open the instance that contains the tabular model you want to query. Right-click the database icon, select **New Query**, and then select **MDX**.
2. In Template Browser, in Analysis Services Templates, open **MDX**, and then open **Queries**. Drag **Basic Query** to the query window.
3. Using **Metadata Explorer**, drag the following fields and measures into the query template:
 - a. Replace <row_axis, mdx_set> with **[Product Category].[Product Category Name]**.
 - b. Replace <column_axis, mdx_set> with **[Date].[Calendar Year].[Calendar Year]**.
 - c. Replace <from_clause, mdx_name> with **[Internet Sales]**.
 - d. Replace <where_clause, mdx_set> with **[Measures].[Internet Total Sales]**.
4. You can execute the query as is, but you will probably want to make some changes, such as adding a function to return specific members. For example, type **.members** after **[Product Category].[Product Category Name]**. For more information, see [Using Member Expressions](#).

Create XMLA Script from a Template

The XMLA command templates that are provided in Template Explorer can be used to create scripts for monitoring and updating Analysis Services objects, regardless of whether the instance is in multidimensional and data mining mode, or tabular mode. The **XMLA** templates include samples for the following types of scripts:

- Backup, restore, and synchronize operations
- Cancel specified process or command
- Process an object
- Discover schema rowsets
- Monitor server status, including jobs, connections, transactions, memory, and performance counters

Create a backup command script from a template

1. In SQL Server Management Studio, open the instance that contains the database you want to query. Right-click the database icon, select **New Query**, and then select **XMLA**.

WARNING

You cannot set the context of an XMLA query by changing the restriction list, or by specifying a database in the connection dialog. You must open the XMLA query window from the database that you want to query.

2. Drag the **Backup** template into the empty query window.
3. Double-click the text within the <DatabaseID> element.
4. In Object Explorer, select the database you want to backup, and drag and drop the database between the brackets of the DatabaseID element.
5. Double-click the text within the <File> element. Type the name of the backup file, including the .abf file extension. Specify the full file path if you are not using the default backup location. For more information, see [Backing Up, Restoring, and Synchronizing Databases \(XMLA\)](#).

Generate a Schema Rowset Query using an XMLA Template

The **Template Explorer** contains only one template for schema rowset queries. To use this template, you must be familiar with the requirements of the individual schema rowset that you want to use, including any required elements, and the columns that can be used as restrictions. For more information, see [Analysis Services Schema Rowsets](#).

Note that many of the schema rowsets have also been exposed as Dynamic Management Views (DMV) for simplicity. By using the corresponding DMV, you can query the schema rowset using syntax like that of Transact-SQL. For example, the following queries return the same results, but one is in XML format, and one is in a tabular format. For more information about DMVs, see [Use Dynamic Management Views \(DMVs\) to Monitor Analysis Services](#).

DMV that returns a list of all schema rowsets available as DMVs:

```
SELECT * FROM $system.DISCOVER_SCHEMA_ROWSETS
```

XMLA command that returns list of available schema rowsets:

```
<Discover xmlns="urn:schemas-microsoft-com:xml-analysis">
  <RequestType>DISCOVER_SCHEMA_ROWSETS</RequestType>
  <Restrictions>
    <RestrictionList>
    </RestrictionList>
  </Restrictions>
  <Properties>
    <PropertyList>
    </PropertyList>
  </Properties>
</Discover>
```

Get a list of data sources for a tabular model using a schema rowset query

1. In SQL Server Management Studio, open the instance that contains the database you want to query. Right-click the database icon, select **New Query**, and then select **XMLA**.

WARNING

You cannot set the context of an XMLA query by changing the restriction list, or by specifying a database in the connection dialog. You must open the XMLA query window from the database that you want to query.

2. Open **Template Explorer**, and drag the template, **Discover Schema Rowsets**, into the blank query window.
3. In the template, replace the [RequestType Element \(XMLA\)](#) element with the following text:

```
<RequestType>MDSHEMA_INPUT_DATASOURCES</RequestType>
```

4. Click **Execute**.

Expected results:

```
<CATALOG_NAME>AW Internet Sales Tabular Model_ 24715b71-ea74-4828-aefc-d4c12c15db64</CATALOG_NAME>
<DATASOURCE_NAME>SqlServer localhost AdventureWorksDW2012</DATASOURCE_NAME>
<DATASOURCE_TYPE>Relational</DATASOURCE_TYPE>
<CREATED_ON>2011-10-12T20:27:05.196667</CREATED_ON>
<LAST_SCHEMA_UPDATE>2011-10-12T20:27:05.196667</LAST_SCHEMA_UPDATE>
<DESCRIPTION />
<TIMEOUT>0</TIMEOUT>
<DBMS_NAME>Microsoft SQL Server</DBMS_NAME>
<DBMS_VERSION>11.00.1724</DBMS_VERSION>
```

Analysis Services Template Reference

The following templates are provided for working with Analysis Services databases and the objects within the database, including mining strictures and mining models, cubes, and tabular models:

CATEGORY	ITEM TEMPLATE	DESCRIPTION
DMX\Model Content	Content Query	Demonstrates how to use the DMX SELECT FROM <i><model></i> .CONTENT statement to retrieve the mining model schema rowset content for a specified mining model.
	Continuous Column Values	Demonstrates how to use the DMX SELECT DISTINCT FROM <i><model></i> statement with the DMX RangeMin and RangeMax functions to retrieve a set of values in a specified range from continuous columns in a specified mining model.
	Discrete Column Values	Demonstrates how to use the DMX SELECT DISTINCT FROM <i><model></i> statement retrieve a complete set of values from discrete columns in a specified mining model.
	Drillthrough Query	Demonstrates how to use the DMX SELECT * FROM Model.CASES statement with the DMX IsInNode function to perform a drillthrough query
	Model Attributes	Demonstrates how to use the DMX System.GetModelAttributes function to return a list of attributes used by a model.
	PMML Content	Demonstrates how to use the DMX SELECT * FROM <i><model></i> .PMML statement to retrieve the Predictive Model Markup Language (PMML) representation of the mining model, for algorithms that support this functionality.

CATEGORY	ITEM TEMPLATE	DESCRIPTION
DMX\Model Management	Add Model	Demonstrates how to use the DMX ALTER MINING MODEL STRUCTURE statement to add a mining model
	Clear Model	Demonstrates how to use the DMX DELETE * FROM MINING MODEL statement to delete the content of a specified mining model.
	Clear Structure Cases	Demonstrates how to use the DMX DELETE FROM MINING STRUCTURE statement to clear mining model structure cases
	Clear Structure	Demonstrates how to use the DMX DELETE FROM MINING STRUCTURE statement to clear a mining model structure
	Create from PMML	Demonstrates how to use the DMX CREATE MINING MODEL statement with the FROM PMML clause to create a mining model from a PMML representation.
	Create Structure Nested	Demonstrates how to use the DMX CREATE MINING STRUCTURE statement with a nested column definition list to create a mining model with nested columns.
	Create Structure	Demonstrates how to use the DMX CREATE MINING STRUCTURE statement to create a mining model.
	Drop Model	Demonstrates how to use the DMX DROP MINING MODEL statement to delete an existing mining model.
	Drop Structure	Demonstrates how to use the DMX DROP MINING STRUCTURE statement to delete an existing mining structure.
	Export Model	Demonstrates how to use the DMX EXPORT MINING MODEL statement using the WITH DEPENDENCIES and PASSWORD clauses to export a mining model, including the data source and data source view on which the mining model depends, to a file.

CATEGORY	ITEM TEMPLATE	DESCRIPTION
	Export Structure	Demonstrates how to use the DMX EXPORT MINING STRUCTURE statement using the WITH DEPENDENCIES clause to export a mining structure, including all of the mining models contained by the mining structure and the data source and data source view on which the mining structure depends, to a file.
	Import	Demonstrates how to use the DMX IMPORT FROM statement using the WITH PASSWORD clause to perform an import .
	Rename Model	Demonstrates how to use the DMX RENAME MINING MODEL statement to rename an existing mining model.
	Rename Structure	Demonstrates how to use the DMX RENAME MINING STRUCTRE statement to rename an existing mining structure.
	Train Model	Demonstrates how to use the DMX INSERT INTO MINING MODEL statement to train a mining model inside a previously trained structure.
	Train Nested Structure	Demonstrates how to combine the DMX INSERT INTO MINING STRUCTURE statement with the SHAPE source data query to train a mining model that contains nested columns with data that contains nested tables, retrieved using a query, from an existing data source.
	Train Structure	Demonstrates how to combine the DMX INSERT INTO MINING STRUCTURE statement with the OPENQUERY source data query to train a mining structure.
DMX\Prediction Queries	Base Prediction	Demonstrates how to combine a DMX SELECT FROM <model> PREDICTION JOIN statement with the OPENQUERY source data query to execute a prediction query against a mining model using data, retrieved using a query, from an existing data source.
	Nested Prediction	Demonstrates how to combine a DMX SELECT FROM <model> PREDICTION JOIN statement with the SHAPE and OPENQUERY source data queries to execute a prediction query against a mining model using data that contains nested tables, retrieved using a query, from an existing data source.

CATEGORY	ITEM TEMPLATE	DESCRIPTION
	Nested Singleton Prediction	Demonstrates how to use a DMX SELECT FROM <model> NATURAL PREDICTION JOIN clause to execute a prediction query against a mining model using a single value, explicitly specified in the prediction query, in a column whose name matches a column in the mining model and which contains a set of values in a nested table created using a UNION statement whose names also match to nested columns in the mining model.
	Singleton Prediction	Demonstrates how to use a DMX SELECT FROM <model> NATURAL PREDICTION JOIN statement to execute a prediction query against a mining model using a single value, explicitly specified in the prediction query, in a column whose name matches a column in the mining model.
	Stored Procedure Call	Demonstrates how to use the DMX CALL statement to call a stored procedure
MDX\Expressions	Moving Average-Fixed	Demonstrates how to use the MDX ParallelPeriod and CurrentMember functions with a naturally ordered set to create a calculated measure that provides a moving average of a measure over a fixed number of time periods contained by a hierarchy in a time dimension.
	Moving Average-Variable	Demonstrates how to use the MDX CASE statement within the Avg function to create a calculated measure that provides a moving average of a measure over a variable number of time periods contained by hierarchy in a time dimension.
	Periods to Date	Demonstrates how to use the MDX PeriodsToDate function in a calculated member.
	Ratio to Parent	Demonstrates how to use the MDX Parent function to create a calculated measure that represents a ratio percentage of a measure for each child of a parent member in a specified hierarchy.

CATEGORY	ITEM TEMPLATE	DESCRIPTION
	Ratio to Total	Demonstrates how to use the All member to create a calculated measure that represents a ratio percentage of a measure for each member in a specified hierarchy.
MDX\Queries	Basic Query	Demonstrates a basic MDX SELECT statement from which you can construct an MDX query.
	KPI Query	Demonstrates how to use the MDX KPIValue and KPIGoal functions to retrieve key performance indicator (KPI) information in an MDX query.
	Sub-select Query	Demonstrates how to create a MDX SELECT statement that retrieves information from a subcube defined by another SELECT statement.
	With Calculated Member	Demonstrates how to use the MDX WITH clause in a SELECT statement to define a calculated member for an MDX query.
	With Named Set	Demonstrates how to use the MDX WITH clause in a SELECT statement to define a named for an MDX query.
XMLA\Management	Backup	Demonstrates how to use the XMLA Backup command to back up an Analysis Services database to a file.
	Cancel	Demonstrates how to use the XMLA Cancel command to cancel all running operations on the current session (for users other than administrators or server administrators), database (for administrators), or instance (for server administrators.)
	Create Remote Partition Database	Demonstrates how to use the XMLA Create command with the Analysis Services Scripting Language (ASSL) Database element to create an Analysis Services database and a data source for storing remote partitions.
	Delete	Demonstrates how to use the XMLA Delete command to delete an existing Analysis Services database.

CATEGORY	ITEM TEMPLATE	DESCRIPTION
	Process Dimension	Demonstrates how to use the XMLA Batch command, combined with the Parallel element and the Process command, to update the attributes of a dimension by using a parallel batch operation.
	Process Partition	Demonstrates how to use the XMLA Batch command, combined with the Parallel element and the Process command, to fully process a partition by using a parallel batch operation.
	Restore	Demonstrates how to use the XMLA Restore command to restore an Analysis Services database from an existing backup file.
	Synchronize	Demonstrates how to use the XMLA Synchronize command to synchronize another Analysis Services database with the current Analysis Services database using the SkipMembership option for the SynchronizeSecurity tag.
XMLA\Schema Rowsets	Discover Schema Rowsets	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_SCHEMA_ROWSETS schema rowset.
XMLA\Server Status	Connections	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_CONNECTIONS schema rowset.
	Jobs	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_JOBS schema rowset.
	Locations	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_LOCATIONS schema rowset, specifying the path of the location backup files.
	Locks	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_LOCKS schema rowset.

CATEGORY	ITEM TEMPLATE	DESCRIPTION
	Memory Grant	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_MEMORYGRANT schema rowset.
	Performance Counters	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_PERFORMANCE_COUNTERS schema rowset.
	Sessions	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_SESSIONS schema rowset.
	Traces	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_TRACES schema rowset.
	Transactions	Demonstrates how to use the XMLA Discover method to retrieve the contents of the DISCOVER_TRANSACTIONS schema rowset.

See Also

[Multidimensional Expressions \(MDX\) Reference](#)


[Data Mining Extensions \(DMX\) Reference](#)

[Analysis Services Scripting Language \(ASSL for XMLA\)](#)

[Analysis Services Scripting Language \(ASSL for XMLA\)](#)

Analysis Services Scripts Project in SQL Server Management Studio

5/16/2018 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

In Analysis Services, you can create an Analysis Server Scripts project in SQL Server Management Studio to group related scripts together for development, management, and source control purposes. If no solution is currently loaded in SQL Server Management Studio, creating a new Analysis Server Scripts project automatically creates a new solution. Otherwise, the new Analysis Server Scripts project can be added to the existing solution or created in a new solution.

You use the following basic steps to create an Analysis Server Scripts project in SQL Server Management Studio:

1. On the File menu, point to **New**, and then click **Project**.

Select **Analysis Server Scripts** project template and then specify a name and location for the new project.

2. Right-click **Connections** to create a new connection in the Connections folder of the Analysis Server Scripts project in Solution Explorer.

This folder contains connection strings to Analysis Services instances, against which the scripts contained by the Analysis Server Scripts project can be executed. You can have multiple connections in an Analysis Server Scripts project, and you can choose a connection against which to run a script contained by the project at the time of execution.

3. Right-click **Queries** to create Multidimensional Expressions (MDX), Data Mining Extensions (DMX), and XML for Analysis (XMLA) scripts in the Scripts folder of the Analysis Server Scripts project in Solution Explorer.
4. Right-click on the project, point to **Add**, and then select **Existing Item** to add miscellaneous files, such as text files that contain notes on the project, in the **Miscellaneous** folder of the Analysis Server Scripts project in Solution Explorer. These files are ignored by SQL Server Management Studio.

File Types

A SQL Server Management Studio solution can contain several file types, depending on what projects you included in the solution and what items you included in each project for that solution. For more information about file types for solutions in SQL Server Management Studio, see [Files That Manage Solutions and Projects](#). Typically, the files for each project in a SQL Server Management Studio solution are stored in the solution folder, in a separate folder for each project.

The project folder for an Analysis Server Scripts project can contain the file types listed in the following table.

FILE TYPE	DESCRIPTION
-----------	-------------

FILE TYPE	DESCRIPTION
Analysis Server Scripts project definition file (.ssmsasproj)	<p>Contains metadata about the folders shown in Solution Explorer, as well as information that indicates which folders should display files included in the project.</p> <p>The project definition file also contains the metadata for Analysis Services connections contained in the project, as well as metadata that associates connections with script files included in the project.</p>
DMX script file (.dmx)	Contains a DMX script included in the project.
MDX script file (.mdx)	Contains an MDX script included in the project.
XMLA script file (.xmla)	Contains an XMLA script included in the project.

Analysis Services Templates

When adding new MDX, DMX, or XMLA scripts to an Analysis Server Scripts project, you have the option of using Template Explorer to locate Analysis Services templates, which are a collection of predefined scripts or statements that demonstrate how to perform a specified action. Template Explorer is available on the **View** menu and includes templates for SQL Server, Analysis Services, and SQL Server Compact. For more information, see [Use Analysis Services Templates in SQL Server Management Studio](#).

See Also

[Creating Multidimensional Models Using SQL Server Data Tools \(SSDT\)](#)

[Multidimensional Expressions \(MDX\) Reference](#)

[Data Mining Extensions \(DMX\) Reference](#)

[Analysis Services Scripting Language \(ASSL for XMLA\)](#)

[Analysis Services Scripting Language \(ASSL for XMLA\)](#)

Schedule SSAS Administrative Tasks with SQL Server Agent

5/16/2018 • 7 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Using the SQL Server Agent service, you can schedule Analysis Services administrative tasks to run in the order and times that you need. Scheduled tasks help you automate processes that run on regular or predictable cycles. You can schedule administrative tasks, such as cube processing, to run during times of slow business activity. You can also determine the order in which tasks run by creating job steps within a SQL Server Agent job. For example, you can process a cube and then perform a backup of the cube.

Job steps give you control over flow of execution. If one job fails, you can configure SQL Server Agent to continue to run the remaining tasks or to stop execution. You can also configure SQL Server Agent to send notifications about the success or failure of job execution.

This topic is a walkthrough that shows two ways of using SQL Server Agent to run XMLA script. The first example demonstrates how to schedule processing of a single dimension. Example two shows how to combine processing tasks into a single script that runs on a schedule. To complete this walkthrough, you will need to meet the following prerequisites.

Prerequisites

SQL Server Agent service must be installed.

By default, jobs run under the service account. The default account for SQL Server Agent is NT Service\SQLAgent\$<instancename>. To perform a backup or processing task, this account must be a system administrator on the Analysis Services instance. For more information, see [Grant server admin rights to an Analysis Services instance](#).

You should also have a test database to work with. You can deploy the AdventureWorks multidimensional sample database or a project from the Analysis Services multidimensional tutorial to use in this walkthrough. For more information, see [Install Sample Data and Projects for the Analysis Services Multidimensional Modeling Tutorial](#).

Example 1: Processing a dimension in a scheduled task

This example demonstrates how to create and schedule a job that processes a dimension.

An Analysis Services scheduled task is an XMLA script that is embedded into a SQL Server Agent job. This job is scheduled to run at desired times and frequency. Because the SQL Server Agent is part of SQL Server, you work with both the Database Engine and Analysis Services to create and schedule an administrative task.

Create a script for processing a dimension in a SQL Server Agent job

1. In SQL Server Management Studio, connect to Analysis Services. Open a database folder and find a dimension. Right-click the dimension and select **Process**.
2. In the **Process Dimension** dialog box, in the **Process Options** column under **Object list**, verify that the option for this column is **Process Full**. If it is not, under **Process Options**, click the option, and then select **Process Full** from the drop-down list.
3. Click **Script**.

This step opens an **XML Query** window that contains the XMLA script that processes the dimension.

4. In the **Process Dimension** dialog box, click **Cancel** to close the dialog box.
5. In the **XMLA Query** window, highlight the XMLA script, right-click the highlighted script, and select **Copy**.

This step copies the XMLA script to the Windows Clipboard. You can leave the XMLA script in the Clipboard or paste it into Notepad or another text editor. The following is an example of the XMLA script.

```
<Batch xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Parallel>
    <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Object>
        <DatabaseID>Adventure Works DW Multidimensional</DatabaseID>
        <DimensionID>Dim Account</DimensionID>
      </Object>
      <Type>ProcessFull</Type>
      <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
    </Process>
  </Parallel>
</Batch>
```

Create and schedule the dimension processing job

1. Connect to an instance of the Database Engine and then open Object Explorer.
2. Expand **SQL Server Agent**.
3. Right-click **Jobs** and select **New Job**.
4. In the **New Job** dialog box, enter a job name in **Name**.
5. Under **Select a page**, select **Steps**, and then click **New**.
6. In the **New Job Step** dialog box, enter a step name in **Step Name**.
7. In **Server**, type **localhost** for a default instance of Analysis Services and **localhost\<instance name>** for a named instance.

If you will be running the job from a remote computer, use the server name and instance name where the job will run. Use the format **<server name>** for a default instance, and **<server name>\<instance name>** for a named instance.

8. In **Type**, select **SQL Server Analysis Services Command**.
9. In **Command**, right-click and select **Paste**. The XMLA script that you generated in the previous step should appear in the command window.
10. Click **OK**.
11. Under **Select a page**, click **Schedules**, and then click **New**.
12. In the **New Job Schedule** dialog box, enter a schedule name in **Name**, and then click **OK**.

This step creates a schedule for Sunday at 12:00 AM. The next step shows you how to manually execute the job. You can also specify a schedule that executes the job when you are monitoring it.

13. In the **New Job** dialog box, click **OK**.
14. In **Object Explorer**, expand **Jobs**, right-click the job you created, and then select **Start Job at Step**.

Because the job has only one step, the job executes immediately. If the job contains more than one step, you

can select the step at which the job should start.

15. When the job finishes, click **Close**.

Example 2: Batch processing a dimension and a partition in a scheduled task

The procedures in this example demonstrate how to create and schedule a job that batch processes an Analysis Services database dimension, and at the same time to process a cube partition that depends on the dimension for aggregation. For more information about batch processing of Analysis Services objects, see [Batch Processing \(Analysis Services\)](#).

Create a script for batch processing a dimension and partition in a SQL Server Agent job

1. Using the same database, expand **Dimensions**, right-click the **Customer** dimension, and select **Process**.
2. In the **Process Dimension** dialog box, in **Process Options** column under **Object list**, verify that the option for this column is **Process Full**.
3. Click **Script**.

This step opens an **XML Query** window that contains the XMLA script that processes the dimension.

4. In the **Process Dimension** dialog box, click **Cancel** to close the dialog box.
5. Expand **Cubes**, expand **Adventure Works**, expand **Measure Groups**, expand **Internet Sales**, expand **Partitions**, right-click the last partition in the list, and then select **Process**.
6. In the **Process Partition** dialog box, in the **Process Options** column under **Object list**, verify that the option for this column is **Process Full**.
7. Click **Script**.

This step opens a second **XML Query** window that contains the XMLA script that processes the partition.

8. In the **Process Partition** dialog box, click **Cancel** to close the editor.

At this point you must merge the two scripts, and ensure that the dimension is processed first.

WARNING

If the partition is processed first, the subsequent dimension processing causes the partition to become unprocessed. The partition would then require a second processing to reach a processed state.

9. In the **XMLA Query** window that contains the XMLA script that processes the partition, highlight the code inside the `Batch` and `Parallel` tags, right-click the highlighted script, and select **Copy**.

```
<Process xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Object>
    <DatabaseID> Adventure Works DW Multidimensional</DatabaseID>
    <CubeID>Adventure Works</CubeID>
    <MeasureGroupID>Fact Internet Sales 1</MeasureGroupID>
    <PartitionID> Internet_Sales_2004</PartitionID>
  </Object>
  <Type>ProcessFull</Type>
  <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
</Process>
```

10. Open the **XMLA Query** window that contains the XMLA script that processes the dimension. Right-click

within the script to the left of the `</Process>` tag and select **Paste**.

The following example shows the revised XMLA script.

```
<Batch xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Parallel>
    <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Object>
        <DatabaseID>Adventure Works DW Multidimensional</DatabaseID>
        <DimensionID>Dim Customer</DimensionID>
      </Object>
      <Type>ProcessFull</Type>
      <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
    </Process>
    <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Object>
        <DatabaseID>Adventure Works DW Multidimensional</DatabaseID>
        <CubeID>Adventure Works</CubeID>
        <MeasureGroupID>Fact Internet Sales 1</MeasureGroupID>
        <PartitionID>Internet_Sales_2004</PartitionID>
      </Object>
      <Type>ProcessFull</Type>
      <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
    </Process>
  </Parallel>
</Batch>
```

11. Highlight the revised XMLA script, right-click the highlighted script, and select **Copy**.
12. This step copies the XMLA script to the Windows Clipboard. You can leave the XMLA script in the Clipboard, save it to a file, or paste it into Notepad or another text editor.

Create and schedule the batch processing job

1. Connect to an instance of SQL Server, and then open Object Explorer.
2. Expand **SQL Server Agent**. Start the service if it is not running.
3. Right-click **Jobs** and select **New Job**.
4. In the **New Job** dialog box, enter a job name in **Name**.
5. In **Steps**, click **New**.
6. In the **New Job Step** dialog box, enter a step name in **Step Name**.
7. In **Type**, select **SQL Server Analysis Services Command**.
8. In **Run as**, select the **SQL Server Agent Service Account**. Recall from the Prerequisites section that this account must have administrative permissions on Analysis Services.
9. In **Server**, specify the server name of the Analysis Services instance.
10. In **Command**, right-click and select **Paste**.
11. Click **OK**.
12. In the **Schedules** page, click **New**.
13. In the **New Job Schedule** dialog box, enter a schedule name in **Name**, and then click **OK**.

This step creates a schedule for Sunday at 12:00 AM. The next step shows you how to manually execute the job. You can also select a schedule which will execute the job when you are monitoring it.

14. Click **OK** to close the dialog box.

15. In **Object Explorer**, expand **Jobs**, right-click the job you created, and select **Start Job at Step**.

Because the job has only one step, the job executes immediately. If the job contains more than one step, you can select the step at which the job should start.



16. When the job finishes, click **Close**.

See Also

[Processing Options and Settings \(Analysis Services\)](#)

Automate Analysis Services Administrative Tasks with SSIS

5/16/2018 • 3 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

Microsoft SQL Server Integration Services enables you to automate execution of DDL scripts, cube and mining model processing tasks, and data mining query tasks. Integration Services can be thought of as a collection of control flow and maintenance tasks, which can be linked to form sequential and parallel data processing jobs.

Integration Services is designed to perform data cleaning operations during data processing tasks, and to bring together data from different data sources. When working with cubes and mining models, Integration Services can transform non-numeric data to numeric data, and can ensure that data values fall within expected bounds, thus creating clean data from which to populate fact tables and dimensions.

Integration Services Tasks

There are two main elements in any Integration Services task or job: control flow elements and data flow elements. The control flow elements define the logical ordering of job progression by applying precedence constraints. The data flow elements concern connectivity between the output of a component to the input of the following component, plus any data transform that might operate on that data in between. As for the decision about where the data goes, precedence constraints contain logic to specify which component receives the output. The Integration Services tasks that are most relevant to Microsoft SQL Server Analysis Services include the Execute DDL Task, the Analysis Services Processing Task, and the Data Mining Query Task. For each of these tasks, the Send Mail Task can be used to send the administrator an e-mail message containing the task results.

The Execute DDL Task

The Execute DDL Task in Integration Services enables you to send DDL scripts directly to the Analysis Services server and to run them automatically. This allows the Analysis Services administrator to perform backup, restore, or sync operations from within an Integration Services package. A package is made up of the control and data flow elements described earlier, which all must be **run regularly**, as do other DDL statements that can be added to tasks. Because the tasks discussed here are frequently run at night, it is particularly useful to have packages that can easily be run from any scheduling application. You can schedule a package to be run at any time using Integration Services Agent. For more information about how to implement this task, see [Analysis Services Execute DDL Task](#).

Analysis Services Processing Task

The Analysis Services Processing Task in Integration Services enables you to automatically populate cubes with new information when you make regular updates to your source relational database. You can process at the dimension, cube, or partition level using the Analysis Services Processing Task. The processing itself can be of type **incremental** or **full**, which you select based on your job requirements. Incremental processing adds new data and performs enough recalculation to keep the target up-to-date, whereas full processing drops existing data for a complete reload and recalculation. Full processing takes more time, but is more complete. For more information about how to implement this task, see [Analysis Services Processing Task](#).

Data Mining Query Task

The Data Mining Query Task in Integration Services enables you to extract and store information from mining models. The information is often stored in a relational database and, for example, can be used to isolate a list of potential customers for a targeted marketing campaign. Data mining can identify the value of a customer and the probability that the customer will respond to a particular marketing pitch. You can use the Data Mining Query Task to extract and modify data to a preferred format. For more information about how to implement this task, see [Data Mining Query Task](#).

See Also

[Partition Processing Destination](#)

[Dimension Processing Destination](#)

[Data Mining Query Transformation](#)

[Processing a multidimensional model \(Analysis Services\)](#)

High availability and Scalability in Analysis Services

5/16/2018 • 7 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

This article describes the most commonly used techniques for making Analysis Services databases high available and scalable. While each objective could be addressed separately, in reality they often go hand in hand: a scalable deployment for large query or processing workloads typically comes with expectations of high availability.

The reverse case is not always true, however. High availability, without scale, can be the sole objective when stringent service level agreements exist for mission-critical, but moderate, query workloads.

Techniques for making Analysis Services highly available and scalable tend to be the same for all server modes (Multidimensional, Tabular, and SharePoint integrated mode). Unless specifically noted otherwise, you should assume the information in this article applies to all modes.

Key Points

Because the techniques for availability and scale differ from those of the relational database engine, a short summary of key points is an effective introduction to techniques used with Analysis Services:

- Analysis Services utilizes the high availability and scalability mechanisms built into the Windows server platform: network load balancing (NLB), Window Server Failover Clustering (WSFC), or both.

NOTE

The Always On feature of the relational database engine does not extend to Analysis Services. You cannot configure an Analysis Services instance to run in an Always On availability group.

Although Analysis Services does not run in Always On Availability Groups, it can both retrieve and process data from Always On relational databases. For instructions on how to configure a highly available relational database so that it can be used by Analysis Services, see [Analysis Services with Always On Availability Groups](#).

- High availability, as a sole objective, can be achieved via server redundancy in a failover cluster. Replacement nodes are assumed to have identical hardware and software configuration as the active node. By itself, WSFC gives you high availability, but without scale.
- Scalability, with or without availability, is achieved via NLB over read-only databases. Scalability is usually a concern when query volumes are large or subject to sudden spikes.

Load balancing, coupled with multiple read-only databases, give you both scale and high availability because all nodes are active, and when a server goes down, requests are automatically redistributed among the remaining nodes. When you need both scalability and availability, an NLB cluster is the right choice.

For processing, objectives of high availability and scalability are less of a concern because you control the timing and scope of operations. Processing can be both partial and incremental across portions of a model, although at some point, you will need to process a model in full on a single server to ensure data consistency across all indexes and aggregations. A robust scalable architecture relies on hardware that can accommodate full processing at whatever cadence is required. For large solutions, this work is structured as an independent operation, with its own hardware resources.

Single vs. multi-server configurations

In a regular single-server deployment, processing and query workloads run concurrently, assuming system resources are sufficient for both activities. Analysis Services keeps existing data structures intact for query support while an updated version is being processed in the background. Having sufficient memory and disk space to handle temporary data structures is a hardware requirement that exists for all server modes, although each mode places different demands on system resources and comes with different levels of NUMA-awareness.

Single servers and scalability

A single high-end, multi-core server might provide sufficient scale on its own. On high end system with a large number of cores, RAM, and disk space, you can potentially scale-up within a single system.

For Multidimensional databases, you can adjust server configuration properties to create affinity between processes and processors. See [Thread Pool Properties](#) for more information.

Multi-server deployments

Sometimes operational requirements dictate the use of multiple servers. For example, failover clusters are multi-server by definition, with each node running on identical hardware and software configurations.

Similarly, a serious requirement for high availability of query workloads typically calls for multiple servers. In this scenario, the recommended configuration for Analysis Services is to use a mix of read-only and read-write databases, running on separate instances of Analysis Services, on dedicated hardware. Read-only databases handle query requests. Read-write databases are used for processing. An expanded description of this commonly used technique is provided in the next section.

Virtual machines and high availability

Another strategy for meeting a high availability requirement could include the use of virtual machines. If availability can be satisfied by standing up a replacement server within hours rather than minutes, you might be able to use virtual machines that can be started on demand, and loaded with updated databases retrieved from a central location.

Scalability using read-only and read-write databases

Network load balancing is recommended for high or escalating query and processing workloads. Analysis Services databases in a NLB solution are defined as read-only databases to ensure consistency across queries.

Although the guidance in [Scale-out querying for Analysis Services using read-only databases](#) (published in 2008) is dated, it's still generally valid. While server operating systems and computer hardware have evolved, and references to specific platforms and CPU limits are obsolete, the basic technique of using read-only and read-write databases for large query volumes is unchanged.

The approach can be summarized as follows:

- Use dedicated hardware and instances of Analysis Services to process the database. Set the database to read-only after processing is finished. See [Switch an Analysis Services database between ReadOnly and ReadWrite modes](#) for instructions.
- Use multiple, identical query servers to run copies of the same read-only Analysis Services database. Servers are deployed in an NLB cluster, accessed via one virtual server name that serves as a single point of entry to the cluster.
- Use robocopy to copy an entire data directory from the processing server to each query server and attach the same database in read-only mode to all query servers. You can also use SAN snapshots, synchronize, or any other tool or method you use for moving production databases.

Resource demands for Tabular and Multidimensional workloads

The following table is a high-level summary of how Analysis Services uses system resources for queries and processing, separated out by server mode and storage. This summary might help you understand what to emphasize in a multi-server deployment that handles a distributed workload.

Server and storage mode	Impact on system resource
Tabular in-memory (default) where queries are executed as table scans of in-memory data structures.	Emphasize RAM and CPUs with fast clock speeds.
Tabular in DirectQuery mode, where queries are offloaded to backend relational database servers and processing is limited to constructing metadata in the model.	Focus on relational database performance, lowering network latency, and maximizing throughput. Faster CPUs can also improve performance of the Analysis Services query processor.
Multidimensional models using MOLAP storage	Choose a balanced configuration that accommodates disk IO for loading data quickly and sufficient RAM for cached data.
Multidimensional models using ROLAP storage.	Maximize disk IO and minimize network latency.

Highly availability and redundancy through WSFC

Analysis Services can be installed into an existing Windows Server Failover Cluster (WSFC) to achieve high availability that restores service within the shortest time possible.

Failover clusters provide full access (read and writeback) to the database, but only one node at a time. Secondary databases run on additional nodes in the cluster, as replacement servers if the first node goes down.

The primary advantage of failover clustering is fast recovery from a service failure. This advantage comes with certain limitations. For one, if failover is never needed, dedicated resources in the cluster are idle. Second, in the event of a failover, all connections are lost, with the corresponding loss of uncommitted work. Most client applications should be able to handle this situation; often, hitting the refresh button in the application will bring the results back.

When considering a WSFC, keep the following points in mind:

- Active/Active is not currently supported. Active/Passive (failover) is the only supported WSFC configuration for Analysis Services.
- When clustering Analysis Services, make sure that any nodes participating in the cluster run on identical or highly similar hardware, and that the operational context of each node is the same in terms of operating system version and service packs, Analysis Services version and service packs (or cumulative updates), and server mode.
- Avoid repurposing a Passive node as another workload's Active node. Any short-term gains in computer utilization will be lost in the event of an actual failover situation if the node is unable to handle both workloads.

In-depth instructions and background information for deploying Analysis Services in a failover cluster are provided in this whitepaper: [How to Cluster SQL Server Analysis Services](#). Although written for SQL Server 2012, this guidance still applies to newer versions of Analysis Services.

See Also


[Synchronize Analysis Services Databases](#)

[Forcing NUMA affinity for Analysis Services Tabular Databases](#)

[An Analysis Services Case Study: Using Tabular Models in a Large-scale Commercial Solution](#)

Log operations in Analysis Services

5/16/2018 • 7 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

An Analysis Services instance will log server notifications, errors, and warnings to the msmdsrv.log file – one for each instance you install. Administrators refer to this log for insights into routine and extraordinary events alike. In recent releases, logging has been enhanced to include more information. Log records now include product version and edition information, as well as processor, memory, connectivity, and blocking events. You can review the entire change list at [Logging improvements](#).

Besides the built-in logging feature, many administrators and developers also use tools provided by the Analysis Services community to collect data about server operations, such as **ASTrace**. See [Microsoft SQL Server Community Samples: Analysis Services](#) for the download links.

This topic contains the following sections:

- [Location and types of logs](#)
- [General information on log file configuration settings](#)
- [MSMDSRV service log file](#)
- [Query logs](#)
- [Mini dump \(.mdmp\) files](#)
- [Tips and best practices](#)

NOTE

If you're looking for information about logging, you might also be interested in tracing operations that show processing and query execution paths. Trace objects for ad hoc and sustained tracing (such as auditing cube access) — as well as recommendations on how to best use Flight Recorder, SQL Server Profiler, and xEvents — can be found through the links on this page: [Monitor an Analysis Services Instance](#).

Location and types of logs

Analysis Services provides the logs described below.

FILE NAME OR LOCATION	TYPE	USED FOR	ON BY DEFAULT
Msmdsrv.log	Error log	Routine monitoring and basic troubleshooting	Yes
OlapQueryLog table in a relational database	Query log	Collect inputs for the Usage Optimization Wizard	No
SQLDmp<guid>.mdmp files	Crashes and exceptions	Deep troubleshooting	No

We highly recommend the following link for additional information resources not covered in this topic: [Initial data collection tips from Microsoft Support](#).

General information on log file configuration settings

You can find sections for each log in the msmdsrv.ini server configuration file, located in the \Program Files\Microsoft SQL Server\MSAS13.MSSQLSERVER\OLAP\Config folder. See [Server Properties in Analysis Services](#) for instructions on editing the file.

Where possible, we suggest that you set logging properties in the server properties page of Management Studio. Although in some cases, you must edit the msmdsrv.ini file directly to configure settings that are not visible in the administrative tools.

```
- <Log>
  <File>msmdsrv.log</File>
  <FileBufferSize>0</FileBufferSize>
  <MessageLogs>File;Console;System</MessageLogs>
+ <ErrorLog>
+ <QueryLog>
+ <Exception>
+ <Trace>
+ <FlightRecorder>
</Log>
```

MSMDSRV service log file

Analysis Services logs server operations to the msmdsrv.log file, one per instance, located at \program files\Microsoft SQL Server\<instance>\Olap\Log.

This log file is emptied at each service restart. In previous releases, administrators would sometimes restart the service for the sole purpose of flushing the log file before it could grow so large as to become unusable. This is no longer necessary. Configuration settings, introduced in SQL Server 2012 SP2 and later, give you control over the size of the log file and its history:

- **MaxFileSizeMB** specifies a maximum log file size in megabytes. The default is 256. A valid replacement value must be a positive integer. When **MaxFileSizeMB** is reached, Analysis Services renames the current file as msmdsrv{current timestamp}.log file, and starts a new msmdsrv.log file.
- **MaxNumberFiles** specifies retention of older log files. The default is 0 (disabled). You can change it to a positive integer to keep versions of the log file. When **MaxNumberFiles** is reached, Analysis Services deletes the file with the oldest timestamp in its name.

To use these settings, do the following:

1. Open msmdsrv.ini in NotePad.
2. Copy the following two lines:

```
<MaxFileSizeMB>256</MaxFileSizeMB>
<MaxNumberOfLogFiles>5</MaxNumberOfLogFiles>
```

3. Paste the two lines into the Log section of msmdsrv.ini, below the filename for msmdsrv.log. Both settings must be added manually. There are no placeholders for them in the msmdsrv.ini file.

The changed configuration file should look like the following:

```
<Log>
<File>msmdsrv.log</File>
<MaxFileSizeMB>256</MaxFileSizeMB>
<MaxNumberOfLogFiles>5</MaxNumberOfLogFiles>
<FileBufferSize>0</FileBufferSize>
```

4. Edit the values if those provided differ from what you want.

5. Save the file.
6. Restart the service.

Query logs

The query log is a bit of a misnomer in that it does not log the MDX or DAX query activity of your users. Instead, it collects data about queries generated by Analysis Services, which is subsequently used as data input in the Usage Based Optimization Wizard. The data collected in the query log is not intended for direct analysis. Specifically, the datasets are described in bit arrays, with a zero or a one indicating the parts of dataset is included in the query. Again, this data is meant for the wizard.

For query monitoring and troubleshooting, many developers and administrators use a community tool, **ASTrace**, to monitor queries. You can also use SQL Server Profiler, xEvents, or an Analysis Services trace. See [Monitor an Analysis Services Instance](#) for tracing-related links.

When should you use the query log? We recommend enabling the query log as part of a query performance tuning exercise that includes the Usage Based Optimization Wizard. The query log does not exist until you enable the feature, create the data structures to support it, and set properties used by Analysis Services to locate and populate the log.

To enable the query log, follow these steps:

1. Create a SQL Server relational database to store the query log.
2. Grant the Analysis Services service account sufficient permissions on the database. The account needs permission to create a table, write to the table, and read from the table.
3. In SQL Server Management Studio, right-click **Analysis Services | Properties | General**, set **CreateQueryLogTable** to true.
4. Optionally, change **QueryLogSampling** or **QueryLogTableName** if you want to sample queries at a different rate, or use a different name for the table.

The query log table will not be created until you have run enough MDX queries to meet the sampling requirements. For example, if you keep the default value of 10, you must run at least 10 queries before the table will be created.

Query log settings are server wide. The settings you specify will be used by all databases running on this server.

Name	Value	Current Value	Default Value	Restart
DataMining \ MaxConcurrentPredictionQueries	0	0	0	
Feature \ ComUdfEnabled	false	false	false	
Feature \ LinkFromOtherInstanceEnabled	false	false	false	
Feature \ LinkInsideInstanceEnabled	true	true	true	
Feature \ LinkToOtherInstanceEnabled	false	false	false	
ForceCommit Timeout	30000	30000	30000	
Log \ FlightRecorder \ Enabled	true	true	true	
Log \ QueryLog \ CreateQueryLogTable	true	true	false	
Log \ QueryLog \ QueryLogConnectionString	Provider=SQLNCLI11.1;Data...	Provider=SQLN...		
Log \ QueryLog \ QueryLogSampling	10	10	10	
Log \ QueryLog \ QueryLogTableName	OlapQueryLog	OlapQueryLog	OlapQueryLog	
LogDir	C:\Program Files\Microsoft...	C:\Program File...		yes

After the configuration settings are specified, run an MDX query multiple times. If sampling is set to 10, run the query 11 times. Verify the table is created. In Management Studio, connect to the relational database engine, open the database folder, open the **Tables** folder, and verify that **OlapQueryLog** exists. If you do not immediately see the table, refresh the folder to pick up any changes to its contents.

Allow the query log to accumulate sufficient data for the Usage Based Optimization Wizard. If query volumes are cyclical, capture enough traffic to have a representative set of data. See [Usage Based Optimization Wizard](#) for instructions on how to run the wizard.

See [Configuring the Analysis Services Query Log](#) to learn more about query log configuration. Although the paper is quite old, query log configuration has not changed in recent releases and the information it contains still applies.

Mini dump (.mdmp) files

Dump files capture data used for analyzing extraordinary events. Analysis Services automatically generates mini dumps (.mdmp) in response to a server crash, exception, and some configuration errors. The feature is enabled, but does not send crash reports automatically.

Crash reports are configured through the Exception section in the Msmdsrv.ini file. These settings control memory dump file generation. The following snippet shows the default values:

```
<Exception>
<CreateAndSendCrashReports>1</CreateAndSendCrashReports>
<CrashReportsFolder/>
<SQLDumperFlagsOn>0x0</SQLDumperFlagsOn>
<SQLDumperFlagsOff>0x0</SQLDumperFlagsOff>
<MiniDumpFlagsOn>0x0</MiniDumpFlagsOn>
<MiniDumpFlagsOff>0x0</MiniDumpFlagsOff>
<MinidumpErrorList>0xC1000000, 0xC1000001, 0xC102003F, 0xC1360054, 0xC1360055</MinidumpErrorList>
<ExceptionHandlingMode>0</ExceptionHandlingMode>
<CriticalErrorHandling>1</CriticalErrorHandling>
<MaxExceptions>500</MaxExceptions>
<MaxDuplicateDumps>1</MaxDuplicateDumps>
</Exception>
```

Configure Crash Reports

Unless otherwise directed by Microsoft Support, most administrators use the default settings. This older KB article is still used to provide instruction on how to configure dump files: [How to configure Analysis Services to generate memory dump files](#).

The configuration setting most likely to be modified is the **CreateAndSendCrashReports** setting used to determine whether a memory dump file will be generated.

VALUE	DESCRIPTION
0	Turns off the memory dump file. All other settings under the Exception section are ignored.
1	(Default) Enables, but does not send, the memory dump file.
2	Enables and automatically sends an error report to Microsoft.

CrashReportsFolder is the location of the dump files. By default, an .mdmp file and associated log records can be found in the \Olap\Log folder.

SQLDumperFlagsOn is used to generate a full dump. By default, full dumps are not enabled. You can set this property to **0x34**.

The following links provide more background:

- [Looking Deeper into SQL Server using Minidumps](#)

- [How to create a user mode dump file](#)
- [How to use the Sqldumper.exe utility to generate a dump file in SQL Server](#)

Tips and best practices

This section is a recap of the tips mentioned throughout this article.

- Configure the msmdsrv.log file to control the size and number of msmdsrv log file. The settings are not enabled by default, so be sure to add them as post-installation step. See [MSMDSRV service log file](#) in this topic.
- Review this blog post from Microsoft Customer Support to learn what resources they use to get information about server operations: [Initial Data Collection](#)
- Use ATrace2012, rather than a query log, to find out who is querying cubes. The query log is typically used to provide input into the Usage Based Optimization Wizard, and the data it captures is not easy to read or interpret. ATrace2012 is a community tool, widely used, that captures query operations. See [Microsoft SQL Server Community Samples: Analysis Services](#).

See Also

[Analysis Services Instance Management](#)

[Introduction to Monitoring Analysis Services with SQL Server Profiler](#)

[Server Properties in Analysis Services](#)

Database Consistency Checker (DBCC) for Analysis Services

5/16/2018 • 19 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server Analysis Services  Azure Analysis Services

DBCC provides on-demand database validation for Multidimensional and Tabular databases on an Analysis Services instance. You can execute DBCC in an MDX or XMLA query window in SQL Server Management Studio (SSMS) and trace the DBCC output in either SQL Server Profiler or xEvent sessions in SSMS.

The command takes an object definition and returns either an empty result set or detailed error information if the object is corrupted. In this article, you'll learn how to run the command, interpret results, and address any problems that arise.

For Tabular databases, consistency checks performed by DBCC are equivalent to the built-in validation that occurs automatically every time you reload, synchronize, or restore a database. In contrast, consistency checks for Multidimensional databases happen only when you run DBCC on demand.

The range of validation checks will vary by mode, with Tabular databases subject to a broader range of checks. Characteristics of a DBCC workload also varies by server mode. Check operations on Multidimensional databases involve reading data from disk, constructing temporary indexes for comparison against actual indexes -- all of which takes significantly longer to complete.

Command syntax for DBCC uses the object metadata specific to the type of database you are checking:

- Multidimensional + pre-SQL Server 2016 Tabular 1100 or 1103 compatibility level databases are described in Multidimensional modeling constructs like **cubeID**, **measuregroupID**, and **partitionID**.
- Metadata for new Tabular model databases at compatibility level 1200 and higher consist of descriptors like **TableName** and **PartitionName**.

DBCC for Analysis Services will execute on any Analysis Services database at any compatibility level, as long as the database is running on a SQL Server 2016 instance. Just make sure you're using the right command syntax for each database type.

NOTE

If you're familiar with [DBCC \(Transact-SQL\)](#), you'll quickly notice that the DBCC in Analysis Services has a much narrower scope. DBCC in Analysis Services is a single command that reports exclusively on data corruption across the database or on individual objects. If you have other tasks in mind, such as collecting information, try using AMO PowerShell or XMLA scripts instead. See [Monitor an Analysis Services Instance](#) for links to more information.

Permission requirements

You must be an Analysis Services database or server administrator (a member of the server role) to run the command. See [Grant database permissions \(Analysis Services\)](#) or [Grant server admin rights to an Analysis Services instance](#) for instructions.

Command syntax

Tabular databases at the 1200 and higher compatibility levels use tabular metadata for object definitions. The complete DBCC syntax for a tabular database created at a SQL Server 2016 functional level is illustrated in the

following example.

Key differences between the two syntaxes include a newer XMLA namespace, no <Object> element, and no <Model> element (there is still only one model per database).

```
<DBCC xmlns="http://schemas.microsoft.com/analysiservices/2014/engine">
  <DatabaseID>MyTabular1200DB_7811b5d8-c407-4203-8793-12e16c3d1b9b</DatabaseID>
  <TableName>FactSales</TableName>
  <PartitionName>FactSales_4</PartitionName>
</DBCC>
```

You can omit lower-level objects, such as table or partition names, to check the entire schema.

You can get object names and DatabaseID from Management Studio, through the property page of each object.

Command syntax for Multidimensional and Tabular 110x databases

DBCC uses identical syntax for multidimensional as well as tabular 1100 and 1103 databases. You can run DBCC against specific database objects, including the entire database. See [Object Element \(XMLA\)](#) for more information about the object definition.

```
<DBCC xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Object>
    <DatabaseID>AdventureWorksDW2014Multidimensional-EE</DatabaseID>
    <CubeID>Adventure Works</CubeID>
    <MeasureGroupID>Fact Internet Sales_1</MeasureGroupID>
    <PartitionID>Internet_Sales_2006</PartitionID>
  </Object>
</DBCC>
```

To run DBCC on objects higher up the object chain, delete any lower-level object ID elements you don't need:

```
<DBCC xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Object>
    <DatabaseID>AdventureWorksDW2014Multidimensional-EE</DatabaseID>
    <CubeID>Adventure Works</CubeID>
  </Object>
</DBCC>
```

For tabular 110x databases, the object definition syntax is modeled after the syntax of Process command (specifically, in how tables are mapped to dimensions and measure groups).

- **CubeID** maps to the model ID, which is **Model**.
- **MeasureGroupID** maps to a table ID.
- **PartitionID** maps to a partition ID.

Usage

In SQL Server Management Studio, you can invoke DBCC using either an MDX or XMLA query window. Additionally, you can use either SQL Server 2017 Profiler or Analysis Services xEvents to view DBCC output. Note that SSAS DBCC messages are not reported to the Windows application event log or the msmdsrv.log file.

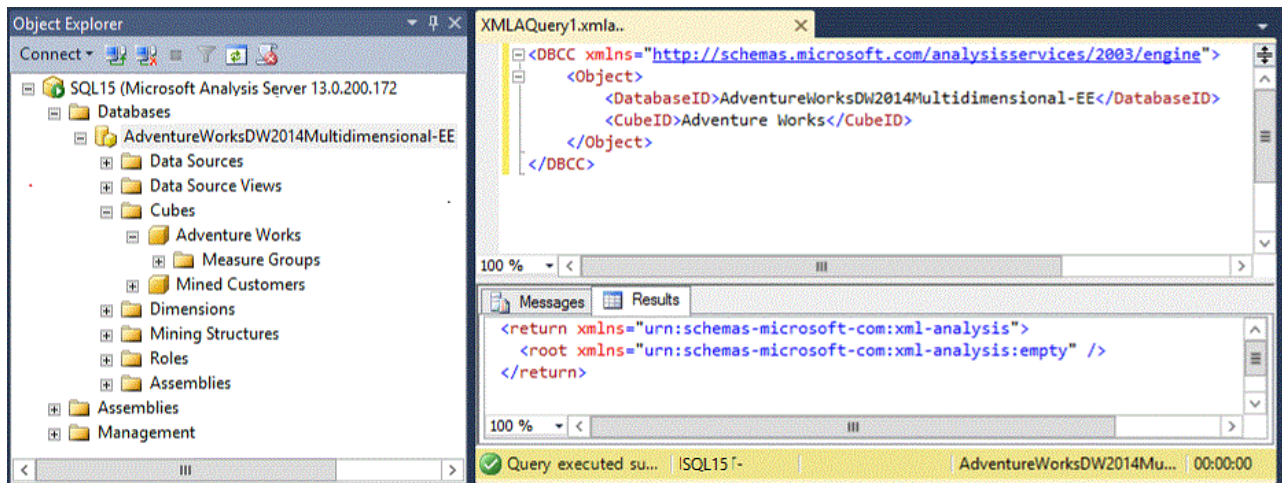
DBCC checks for physical data corruption, as well as logical data corruption that occur when orphaned members exist in a segment. A database must be processed before you can run DBCC. It skips remote, empty, or unprocessed partitions.

The command runs in a read transaction, and can thus be kicked out by force commit timeout. Partition checks are run in parallel.

A service restart might be required to pick up any corruption errors that have occurred since the last service restart. Reconnecting to the server is not enough to pick up the changes.

Run DBCC commands in Management Studio

For ad hoc queries, open an MDX or XMLA query window in SQL Server Management Studio. To do this, right-click the database | **New Query | XMLA**) to run the command and read the output.



The Results tab will indicate an empty result set (as shown in the screenshot) if no problems were detected.

The Messages tab provides detail information but is not always reliable for smaller databases. Status messages are sometimes trimmed, indicating the command completed, but without the status check messages on each object. A typical message report might look similar to the one shown below.

Messages reported from DBCC for the cube validation check

```
Executing the query ...
READS, 0
READ_KB, 0
WRITES, 0
WRITE_KB, 0
CPU_TIME_MS, 0
ROWS_SCANNED, 0
ROWS_RETURNED, 0

<DBCC xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
<Object>
<DatabaseID>AdventureWorksDW2014Multidimensional-EE</DatabaseID>
<CubeID>Adventure Works</CubeID>
</Object>
</DBCC>
Started checking segment indexes for the 'Internet_Sales_2011' partition.
Started checking segment indexes for the 'Internet_Sales_2012' partition.
Finished checking segment indexes for the 'Internet_Sales_2011' partition.
Started checking segment indexes for the 'Internet_Sales_2013' partition.
Finished checking segment indexes for the 'Internet_Sales_2012' partition.
Started checking segment indexes for the 'Internet_Sales_2014' partition.
Started checking segment indexes for the 'Internet_Orders_2011' partition.
Finished checking segment indexes for the 'Internet_Sales_2014' partition.
Started checking segment indexes for the 'Internet_Orders_2012' partition.
Started checking segment indexes for the 'Internet_Orders_2013' partition.
Finished checking segment indexes for the 'Internet_Orders_2012' partition.
...
Run complete
```

Output when running DBCC against an earlier version of Analysis Services

DBCC is only supported on databases running on a SQL Server 2017 instance. Running the command on older systems will return this error.

```
Executing the query ...  
The DBCC element at line 7, column 87 (namespace http://schemas.microsoft.com/analysisservices/2003/engine)  
cannot appear under Envelope/Body/Execute/Command.  
Execution complete
```

Trace DBCC output in SQL Server Profiler 2016

You can view DBCC output in a Profiler trace that includes Progress Reports events (Progress Report Begin, Progress Report Current, Progress Report End, and Progress Report Error).

1. Start a trace. See [Use SQL Server Profiler to Monitor Analysis Services](#) for help on how to use SQL Server Profiler with Analysis Services.
2. Choose **Command Begin** and **Command End** plus any or all of the **Progress Report** Events.
3. Run the DBCC command in Management Studio in either an XMLA or MDX query window, using the syntax provided in a previous section.
4. In SQL Server Profiler, DBCC activity is indicated through **Command** events having an event subclass of DBCC:

EventClass	EventSubclass	ActivityID	ApplicationName
Command Begin	12 - Batch		
Command Begin	32 - DBCC	374D0361...	Microsoft SQL Server Management Studio - Query
Command End	32 - DBCC	374D0361...	Microsoft SQL Server Management Studio - Query

Event code 32 is DBCC execution.

Event code 64 is a DBCC progress report on individual objects.

Event code 63 is a segment check for multidimensional objects.

For both event subclasses, review **TextData** values for messages returned by DBCC.

Status messages start with "Checking consistency of <object>", "Started checking <object>", or "Finished checking <object>".

NOTE

In CTP 3.0, objects are identified by internal names. For example, a Categories hierarchy is articulated as H\$Categories-<objectID>. Internal names should be replaced by user friendly names in an upcoming CTP.

Error messages are listed below.

Trace DBCC output in an xEvent session in SSMS

Extended events sessions can use both profiler events or xEvents. Refer to the previous section for guidance on adding **Command** and **Progress Report** events.

1. Start a session by right-clicking a database > **Management** > **Extended Events** > **Sessions** > **New Session**. See [Monitor Analysis Services with SQL Server Extended Events](#) for more information.
2. Choose any or all of the **Progress Report** Events for the Profiler event category or **RequestProgress** events for the PureXevent category.
3. Run the DBCC command in Management Studio in either an XMLA or MDX query window, using the syntax provided in a previous section.

4. In SSMS, refresh the Sessions folder. Right-click the session name > **Watch Live Data**.
5. Review TextData values for messages returned by DBCC. TextData is a property of an event field and shows status and error messages returned by the event.

Status messages start with "Checking consistency of <object>", "Started checking <object>", or "Finished checking <object>".

Error messages are listed below.

Reference: Consistency checks and errors for Multidimensional databases

For multidimensional databases, only partition indexes are validated. During execution, DBCC builds a temporary index for each partition and compares it with the persisted index on disk. Building a temporary index requires reading all data from the partition data on disk and then holding the temporary index in memory for comparison. Given the additional workload, your server might experience significant disk IO and memory consumption while running a DBCC execution.

Detection of Multidimensional index corruption includes the following checks. Errors in this table appear in xEvent or Profiler traces for failures at the object level.

Object	DBCC check description	Error on failure
Partition Index	Check segment statistics and indexes. Compares the ID of each member in the temporary partition index against the partition statistics stored on disk. If a member is found in the temporary index with a data ID value outside the range stored for the partition index statistics on disk, then the statistics for the index are considered corrupt.	The partition segment statistics are corrupted.
Partition Index	Validates metadata. Verifies that each member in the temporary index can be found in the index header file for the segment on disk.	The partition segment is corrupted.
Partition Index	Scan segments to look for physical corruptions. Reads the index file on disk for each member in the temporary index and verifies that the size of the index records match, and that the same data pages are flagged as having records for the current member.	The partition segment is corrupted.

Reference: Consistency checks and errors for Tabular databases

The following table is list of all consistency checks performed on tabular objects, alongside errors that are raised if the check indicates corruption. Errors in this table appear in xEvent or Profiler traces for failures at the object level.

Object	DBCC check description	Error on failure
Database	Checks count of tables in the database. A value less than zero indicates corruption.	There is corruption in the storage layer. The collection of tables in the '% {parent/}' database is corrupt.
Database	Checks internal structure used to track Referential Integrity and throw s an error if the size is incorrect.	Database files failed to pass consistency checks.
Table	Checks internal value used to determine if table is a Dimension or Fact table. A value that falls outside the known range indicates corruption.	Database consistency checks (DBCC) failed while checking the table statistics.
Table	Checks that the number of partitions in the segment map for the table matches the number of partitions defined for the table.	There is corruption in the storage layer. The collection of partitions in the '% {parent/}' table is corrupt.
Table	If a tabular database was created or imported from PowerPivot for Excel 2010 and has a partition count greater than one, an error will be raised, as partition support was added in later versions and this would indicate corruption.	Database consistency checks (DBCC) failed while checking the segment map.
Partition	Verifies for each partition that the number segments of data and the record count for each segment of data in the segment matches the values stored in the index for the segment.	Database consistency checks (DBCC) failed while checking the segment map.
Partition	Raise an error if the number of total records, segments, or records per segment is not valid (less than zero), or the number of segments doesn't match the calculated number of segments needed based on total record count.	Database consistency checks (DBCC) failed while checking the segment map.
Relationship	Raise an error if the structure used to store data about the relationship contains no records or if the name of the table used in the relationship is empty.	Database consistency checks (DBCC) failed while checking relationships.
Relationship	<p>Verify that the name of the primary table, primary column, foreign table, and foreign column are set and that the columns and tables involved in the relationship can be accessed.</p> <p>Verify that the column types involved are valid and that the index of Primary Key-Foreign Key values results in a valid lookup structure.</p>	Database consistency checks (DBCC) failed while checking relationships.

Hierarchy	Raise an error if the sort order for the hierarchy isn't a recognized value.	Database consistency checks (DBCC) failed while checking the '%{hier/}' hierarchy.
Hierarchy	<p>The checks performed on the hierarchy depend on the internal type of hierarchy mapping scheme used.</p> <p>All hierarchies are checked for correct processed state, that the hierarchy store exists, and that where applicable, data structures used for a data-ID-to-hierarchy-position conversion exists.</p> <p>Assuming all these checks pass, the hierarchy structure is walked to verify that each position in the hierarchy points to the correct member. If any of these tests fail, an error is raised.</p>	Database consistency checks (DBCC) failed while checking the '%{hier/}' hierarchy.
User defined Hierarchy	<p>Checks that the hierarchy level names are set.</p> <p>If the hierarchy has been processed, check that the internal hierarchy data store has the correct format. Verify that the internal hierarchy store doesn't contain any invalid data values.</p> <p>If the hierarchy is marked as unprocessed, confirm that this state applies to old data structures and that all levels of the hierarchy are marked as empty.</p>	Database consistency checks (DBCC) failed while checking the '%{hier/}' hierarchy.
Column	Raise an error if the encoding used for the column is not set to a known value.	Database consistency checks (DBCC) failed while checking the column statistics.
Column	Check whether the column was compressed by the in-memory engine or not.	Database consistency checks (DBCC) failed while checking the column statistics.
Column	Check the compression type on the column for known values.	Database consistency checks (DBCC) failed while checking the column statistics.
Column	When the column "tokenization" is not set to a known value, raise an error.	Database consistency checks (DBCC) failed while checking the column statistics.
Column	If the id range stored for a columns data dictionary does not match the number of values in the data dictionary or is outside the allowed range, raise an error.	Database consistency checks (DBCC) failed while checking the data dictionary.

Column	Check that the number of data segments for a column matches the number of data segments for the table to which it belongs.	There is corruption in the storage layer. The collection of segments in the '% {parent/}' column is corrupt.
Column	Check that the number of Partitions for a data column matches the number of partitions for the data segment map for the column.	Database consistency checks (DBCC) failed while checking the segment map.
Column	Verify that the number of records in a column segment matches the record count stored in the index for that column segment.	There is corruption in the storage layer. The collection of segments in the '% {parent/}' column is corrupt.
Column	If a column has no segment statistics, raise an error.	Database consistency checks (DBCC) failed while checking the segment statistics.
Column	If a column has no compression information or segment storage, raise an error.	Database files failed to pass consistency checks.
Column	Report an error if segment statistics for a column don't match the actual column values for Minimum Data ID, Maximum Data ID, number of Distinct values, number of rows, or presence of NULL values.	Database consistency checks (DBCC) failed while checking the segment statistics.
ColumnSegment	If the minimum data ID or maximum data ID is less than the system reserved value for NULL, mark the column segment information as corrupt.	Database consistency checks (DBCC) failed while checking the segment statistics.
ColumnSegment	If there are no rows for this segment, the minimum and maximum data values for the column should be set to the system reserved value for NULL. If the value is not null, raise an error.	Database consistency checks (DBCC) failed while checking the segment statistics.
ColumnSegment	If the column has rows and at least one non-null value, check that the minimum and maximum data id for the column is greater than the system reserved value for NULL.	Database consistency checks (DBCC) failed while checking the segment statistics.
Internal	Verify that the store tokenization hint is set and that if the store is processed, there are valid pointers for internal tables. If the store is not processed, verify all the pointers are null. If not, return a generic DBCC error.	Database files failed to pass consistency checks.
DBCC Database	Raise an error if the database schema has no tables or one or more tables cannot be accessed.	There is corruption in the storage layer. The collection of tables in the '% {parent/}' database is corrupt.

DBCC Database	Raise an error if a table is marked as temporary or has an unknown type.	A bad table type was encountered.
DBCC Database	Raise an error if the number of relationships for a table has a negative value, or if any table has a relationship defined and a corresponding relationship structure cannot be found.	There is corruption in the storage layer. The collection of relationships in the '% {parent/}' table is corrupt.
DBCC Database	If the compatibility level for the database is 1050 (SQL Server 2008 R2/PowerPivot v1.0) and the number of relationships exceeds the number of tables in the model, mark the database as corrupted.	Database files failed to pass consistency checks.
DBCC Table	For the table under validation, check if the number of columns is less than zero and raise an error if true. An error also occurs if the column store for a column in the table is NULL.	There is corruption in the storage layer. The collection of columns in the '% {parent/}' table is corrupt.
DBCC Partition	Checks the table that the partition being validated belongs to, and if the number of columns for the table is less than zero, it indicates the Columns collection is corrupt for the table. An error will also occur if the column store for a column in the table is NULL.	There is corruption in the storage layer. The collection of columns in the '% {parent/}' table is corrupt.
DBCC Partition	Loops through each column for the selected partition and checks that each segment for the partition has a valid link to a column segment structure. If any segment has a NULL link, the partition is considered corrupt.	There is corruption in the storage layer. The collection of segments in the '% {parent/}' column is corrupt.
Column	Returns an error if the column type is not valid.	A bad segment type was encountered.
Column	Returns an error if any column has a negative count for the number of segments in a column, or if the pointer to the column segment structure for a segment has a NULL link.	There is corruption in the storage layer. The collection of segments in the '% {parent/}' column is corrupt.
DBCC Command	The DBCC Command will report multiple status messages as it proceeds through the DBCC operation. It will report a status message before starting that includes the database, table, or column name of the object, and again after finishing each object check.	<p>Checking consistency of the <objectname> <objecttype>. Phase: pre-check.</p> <p>Checking consistency of the <objectname> <objecttype>. Phase: post-check.</p>

Common resolutions for error conditions

The following errors appear in SQL Server Management Studio or in msmdsrv.log files. These errors appear when one or more checks fail to pass. Depending on the error, the recommended resolution is to either reprocess an

object, delete and redeploy a solution, or restore the database.

ERROR	ISSUE	RESOLUTION
Errors in the metadata manager The object reference '<objectID>' is not valid. It does not match the structure of the metadata class hierarchy.	malformed command	Check the command syntax. Most likely, you included a lower level object without specifying one or more of its parent objects.
Errors in the metadata manager Either the <object> with the ID of '<objectID>' does not exist in the <parentobject> with the ID of '<parentobjectID>', or the user does not have permissions to access the object.	Index corruption (multidimensional)	Reprocess the object and any dependent objects.
Error occurred during consistency check of the partition An error occurred while checking consistency of the <partition-name> partition of the <measure-group-name> measure group for the <cube-name> cube from the <database-name> database. Please re-process the partition or indexes in order to fix the corruption.	Index corruption (multidimensional)	Reprocess the object and any dependent objects.
Partition segment statistics corrupted	Index corruption (multidimensional)	Reprocess the object and any dependent objects.
Partition segment is corrupted	Metadata corruption (multidimensional or tabular)	Delete and redeploy the project, or restore from a backup and reprocess. See How to handle corruption in Analysis Services databases (blog) for instructions.
Table metadata corruption Table <table-name> metadata file is corrupt. The main table is not found under DataFileList node.	Metadata corruption (tabular only)	Delete and redeploy the project, or restore from a backup and reprocess. See How to handle corruption in Analysis Services databases (blog) for instructions.
Corruption in storage layer Corruption in storage layer: collection of <type-name> in <parent-name> <parent-type> is corrupt.	Metadata corruption (tabular only)	Delete and redeploy the project, or restore from a backup and reprocess. See How to handle corruption in Analysis Services databases (blog) for instructions.
System table is missing System table <table-name> is missing.	Object corruption (tabular only)	Reprocess the object and any dependent objects

ERROR	ISSUE	RESOLUTION
Table statistics are corrupt Statistics of table System table <table-name> is missing.	Metadata corruption (tabular only)	Delete and redeploy the project, or restore from a backup and reprocess. See How to handle corruption in Analysis Services databases (blog) for instructions.

Disable automatic consistency checks on database load operations through the msmdsrv.ini configuration file

Although its not recommended, you can disable the built-in database consistency checks that occur automatically on database load events (on tabular databases only). To do this, you will need to modify a configuration setting in the msmdsrv.ini file:

```
<ConfigurationSettings>
  <Vertipaq />
    <DisableConsistencyChecks />
```

This setting is not present in the configuration file and must be added manually.

Valid values are as follows:

- **-2** (default) DBCC is enabled. If the server can logically resolve the error with a high degree of certainty, a fix will be applied automatically. Otherwise, an error will be logged.
- **-1** DBCC is partially enabled. It is enabled for RESTORE and on pre-commit validations that check database state at the end of a transaction.
- **0** DBCC is partially enabled. Database consistency checks are performed during RESTORE, IMAGELOAD, LOCALCUBELOAD, and ATTACH operations.
- **1** DBCC is disabled. Data integrity checks are disabled, however deserialization checks will still occur.

NOTE

This setting has no impact on DBCC when running the command on demand.

See Also

[Process Database, Table, or Partition \(Analysis Services\)](#)
[Processing a multidimensional model \(Analysis Services\)](#)
[Monitor an Analysis Services Instance](#)
[Compatibility Level for Tabular models in Analysis Services](#)
[Server Properties in Analysis Services](#)