

Table of Contents

Analysis Services Documentation

Overview

- What is Azure Analysis Services?

Get started

- Create a server in Azure portal

- Create a server with Azure PowerShell

- Deploy a model from Visual Studio

- Create a model in Azure portal (preview)

Concepts

- Supported data sources

- Connecting to on-premises data sources

- Connecting to servers

- Authentication and user permissions

- Compatibility level

- High availability

How-to

- Install and configure an on-premises data gateway

- Import a Power BI Desktop file

- Backup and restore

- Refresh with REST API

- Configure scale-out

- Automate administrative tasks

 - Create service principal - Azure portal

 - Create service principal - PowerShell

 - Add a service principal to server administrator role

Server management

 - Configure a server name alias

 - Manage a server

 - Manage database users

[Manage server administrators](#)

[Monitor server metrics](#)

[Setup diagnostic logging](#)

[Use gateway for data sources on VNet](#)

Connect

[Get client libraries](#)

[Connect with Excel](#)

[Connect with Power BI](#)

[Create an .odc file](#)

Tutorials

[1 - Add a sample model](#)

[Data modeling - Adventure Works tutorial](#)

Reference

[Azure PowerShell](#)

[REST](#)

[DAX](#)

[Power Query M](#)

[Tabular Model Scripting Language \(TMSL\)](#)

Resources

[Samples](#)

[Analysis Services team blog](#)

[Azure Analysis Services forum](#)

[Azure Roadmap](#)

[Azure status](#)

[Feedback](#)

[Pricing](#)

[Pricing calculator](#)

[Stackoverflow](#)

[Videos](#)

Learn how to set up and manage data modeling with Analysis Services. Tutorials and other documentation show you how to create an enterprise-grade BI solution using tabular data in the cloud or on-premises.

[Learn about Analysis Services](#)

[Azure Video Library](#)

[Create an Analysis Services server](#)

Reference

Command-Line

[PowerShell](#)

REST

[REST API Reference](#)

Others

[DAX](#)

[Power Query M](#)

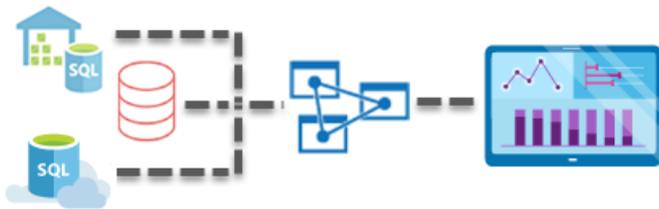
What is Azure Analysis Services?

4/24/2018 • 8 min to read • [Edit Online](#)



Azure Analysis Services provides enterprise-grade data modeling in the cloud. It is a fully managed platform as a service (PaaS), integrated with Azure data platform services.

With Analysis Services, you can mashup and combine data from multiple data sources, define metrics, and secure your data in a single, trusted semantic data model. The data model provides an easier and faster way for your users to browse massive amounts of data with client applications like Power BI, Excel, Reporting Services, third-party, and custom apps.



Check out [this video](#) to learn how Azure Analysis Services fits in with Microsoft's overall BI capabilities, and how you can benefit from getting your data models into the cloud.

Built on SQL Server Analysis Services

Azure Analysis Services is compatible with many great features already in SQL Server Analysis Services Enterprise Edition. Azure Analysis Services supports tabular models at the 1200 and 1400 [compatibility levels](#). Partitions, row-level security, bi-directional relationships, and translations are all supported. In-memory and DirectQuery modes mean lightning fast queries over massive and complex datasets.

Tabular models offer rapid development and are highly customizable. For developers, tabular models include the Tabular Object Model (TOM) to describe model objects. TOM is exposed in JSON through the [Tabular Model Scripting Language \(TMSL\)](#) and the AMO data definition language through the [Microsoft.AnalysisServices.Tabular](#) namespace.

Better with Azure

Azure Analysis Services integrates with many Azure services enabling you to build sophisticated analytics solutions. Integration with [Azure Active Directory](#) provides secure, role-based access to your critical data. Integrate with [Azure Data Factory](#) pipelines by including an activity that loads data into the model. [Azure Automation](#) and [Azure Functions](#) can be used for lightweight orchestration of models using custom code.

Get up and running quickly

In Azure portal, you can [create a server](#) within minutes. And, with Azure Resource Manager [templates](#) and PowerShell, you can provision servers using a declarative template. With a single template, you can deploy multiple services along with other Azure components such as storage accounts and Azure Functions.

Once you have a server created, you can create a tabular model right in Azure portal. With the new (preview) [Web](#)

[designer feature](#), you can connect to an Azure SQL Database, Azure SQL Data Warehouse data source, or import a Power BI Desktop .pbix file. Relationships between tables are created automatically, and you can create measures or edit the model.bim file in json format right from your browser.

Scale to your needs

The right tier when you need it

Azure Analysis Services is available in Developer, Basic, and Standard tiers. Within each tier, plan costs vary according to processing power, QPUs, and memory size. When you create a server, you select a plan within a tier. You can change plans up or down within the same tier, or upgrade to a higher tier, but you cannot downgrade from a higher tier to a lower tier.

Go up, down, or pause your server. Use the Azure portal or have total control on-the-fly by using PowerShell. You only pay for what you use. To learn more about the different plans and tiers, and use the pricing calculator to determine the right plan for you, see [Azure Analysis Services Pricing](#).

Scale-out resources for fast query responses

With Azure Analysis Services scale-out, client queries are distributed among multiple *query replicas* in a query pool. Query replicas have synchronized copies of your tabular models. By spreading the query workload, response times during high query workloads can be reduced. Model processing operations can be separated from the query pool, ensuring client queries are not adversely affected by processing operations. You can create a query pool with up to seven additional query replicas (eight total, including your server).

Just like with changing your tier, you can scale-out query replicas according to your needs. Configure scale-out in the portal or by using REST APIs. To learn more, see [Azure Analysis Services scale-out](#).

Keep your data close

Azure Analysis Services servers can be created in the following [Azure regions](#):

AMERICAS	EUROPE	ASIA PACIFIC
Brazil South Canada Central East US 2 North Central US South Central US West Central US West US	North Europe UK South West Europe	Australia Southeast Japan East Southeast Asia West India

New regions are being added all the time, so this list might be incomplete. You choose a location when you create your server in Azure portal or by using Azure Resource Manager templates. To get the best performance, choose a location nearest your largest user base. Assure [high availability](#) by deploying your models on redundant servers in multiple regions.

Migrate your existing tabular models

If you already have existing on-premises SQL Server Analysis Services model solutions, you can migrate to Azure Analysis Services without significant changes. To migrate, you can use SSDT to deploy your model to your server. Or, in SSMS, you can use backup and restore or TMSL.

If you have on-premises data sources, you need to install and configure an [On-premises data gateway](#). If you have roles and role members already configured, your roles migrate, but you have to readd role members by using SSMS or PowerShell.

Connect to popular data sources

Azure Analysis Services supports [connecting to data sources](#) on-premises in your organization and in the cloud. Combine data from both on-premises and cloud data sources for a hybrid solution.

New tabular 1400 models use the modern Get Data feature in SSDT, based on the M formula query language. With Get Data, you have more data transformation and mashup features, and the ability to create and edit your own advanced M formula language queries. For example, with tabular 1400 models, you can model on data files in Azure Blob Storage.

Use the tools you already know



SQL Server Data Tools (SSDT) for Visual Studio

Develop and deploy models with the free [SQL Server Data Tools \(SSDT\) for Visual Studio](#). SSDT includes Analysis Services project templates that get you up and going quickly. SSDT now includes the modern Get Data datasource query and mashup functionality for tabular 1400 models. If you're familiar with Get Data in Power BI Desktop and Excel 2016, you already know how easy it is to create highly customized data source queries.

Sql Server Management Studio

Manage your servers and model databases by using [SQL Server Management Studio \(SSMS\)](#). Connect to your servers in the cloud. Run TMSL scripts right from the XMLA query window, and automate tasks by using TMSL scripts. New features and functionality happen fast - SSMS is updated monthly.

PowerShell

Server resource management tasks like creating servers, suspending or resuming server operations, or changing the service level (tier) use Azure Resource Manager (AzureRM) cmdlets. Other tasks for managing databases such as adding or removing role members, processing, or running TMSL scripts use cmdlets in the SqlServer module. Both AzureRM and SqlServer modules are available in the [PowerShell gallery](#).

Your data is secure



Authentication

User authentication for Azure Analysis services is handled by [Azure Active Directory \(AAD\)](#). When attempting to log in to an Azure Analysis Services database, users use an organization account identity with access to the database they are trying to access. These user identities must be members of the default Azure Active Directory for the subscription where the Azure Analysis Services server resides. To learn more, see [Authentication and user permissions](#).

Data security

Azure Analysis Services utilizes Azure Blob storage to persist storage and metadata for Analysis Services databases. Data files within Blob are encrypted using Azure Blob Server Side Encryption (SSE). When using Direct Query mode, only metadata is stored. The actual data is accessed from the data source at query time.

Firewall

Azure Analysis Services Firewall blocks all client connections other than those specified in rules. Configure rules specifying allowed IP addresses by individual client IPs or by range. Power BI (service) connections can also be allowed or blocked.

On-premises data sources

Secure access to data residing on-premises in your organization is achieved by installing and configuring an [On-premises data gateway](#). Gateways provide access to data for both Direct Query and in-memory modes. When an Azure Analysis Services model connects to an on-premises data source, a query is created along with the encrypted credentials for the on-premises data source. The gateway cloud service analyzes the query and pushes the request to an Azure Service Bus. The on-premises gateway polls the Azure Service Bus for pending requests. The gateway then gets the query, decrypts the credentials, and connects to the data source for execution. The results are then sent from the data source, back to the gateway and then on to the Azure Analysis Services database.

Azure Analysis Services is governed by the [Microsoft Online Services Terms](#) and the [Microsoft Online Services Privacy Statement](#). To learn more about Azure Security, see the [Microsoft Trust Center](#).

Supports the latest client tools



Modern data exploration and visualization tools like Power BI, Excel, SQL Server 2017 Reporting Services, and third-party tools are all supported, providing users with highly interactive and visually rich insights into your model data.

Clients use MSOLAP, AMO, or ADOMD [client libraries](#) to connect to Analysis Services servers. Microsoft client applications like Power BI Desktop and Excel install all three client libraries. But keep in mind, depending on the version or frequency of updates, client libraries may not be the latest versions required by Azure Analysis Services. The same applies to custom applications or other interfaces such as AsCmd, TOM, ADOMD.NET. These applications typically require manually installing the libraries as part of a package.

Get help

Documentation

Azure Analysis Services is simple to set up and to manage. You can find all the info you need to create and manage your server services here. When creating a data model to deploy to your server, it's much the same as it is for creating a data model you deploy to an on-premises server. There's an extensive library of conceptual, procedural, tutorials, and reference articles at [SQL Server Analysis Services help](#).

Videos

Check out helpful videos at [Azure Analysis Services on Channel 9](#).

Blogs

Things are changing rapidly. You can always get the latest information on the [Analysis Services team blog](#) and [Azure blog](#).

Community

Analysis Services has a vibrant community of users. Join the conversation on [Azure Analysis Services forum](#).

Feedback

Have suggestions or feature requests? Be sure to leave your comments on [Azure Analysis Services Feedback](#).

Have suggestions about the documentation? You can add comments using Livefyre at the bottom of each article.

Next steps

Learn how to [create a server](#) in Azure.

Once you have a server created, [add a sample model](#).

Create an Analysis Services server in Azure portal

4/24/2018 • 1 min to read • [Edit Online](#)

This article walks you through creating an Analysis Services server resource in your Azure subscription.

Before you begin, you need:

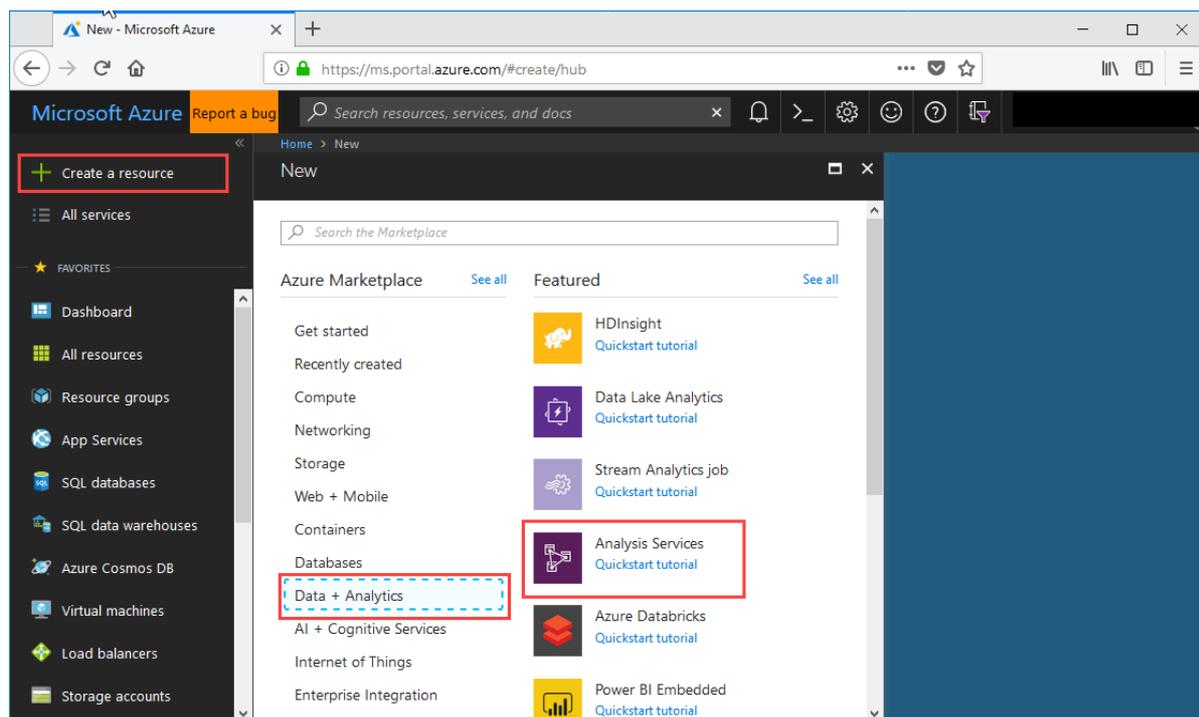
- **Azure subscription:** Visit [Azure Free Trial](#) to create an account.
- **Azure Active Directory:** Your subscription must be associated with an Azure Active Directory tenant. And, you need to be signed in to Azure with an account in that Azure Active Directory. To learn more, see [Authentication and user permissions](#).

Log in to the Azure portal

Log in to the [Azure portal](#)

Create a server

1. Click + **Create a resource** > **Data + Analytics** > **Analysis Services**.



2. In **Analysis Services**, fill in the required fields, and then press **Create**.

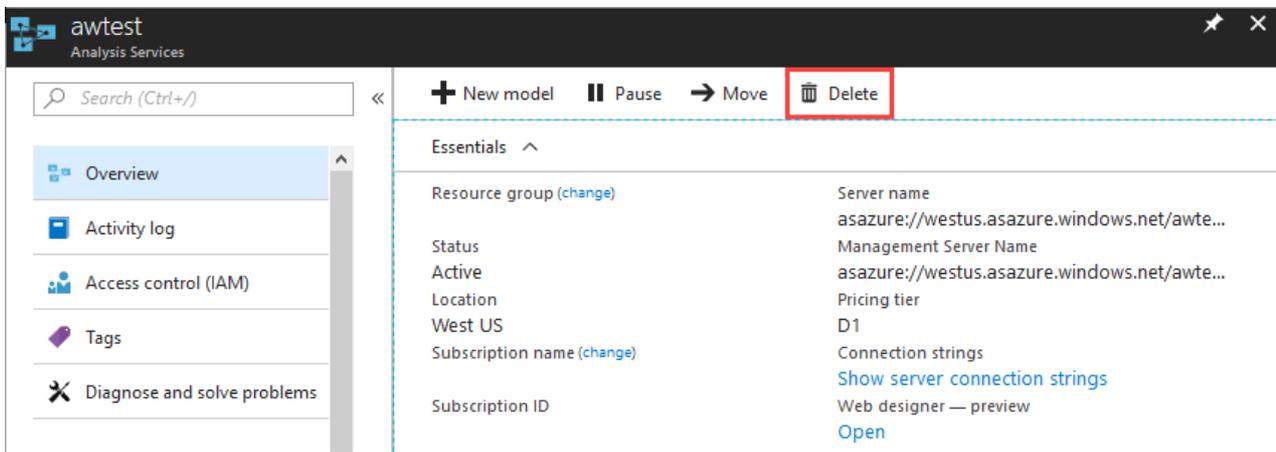
- **Server name:** Type a unique name used to reference the server.
- **Subscription:** Select the subscription this server will be associated with.
- **Resource group:** Create a new resource group or select one you already have. Resource groups are designed to help you manage a collection of Azure resources. To learn more, see [resource groups](#).
- **Location:** This Azure datacenter location hosts the server. Choose a location nearest your largest user base.
- **Pricing tier:** Select a pricing tier. If you are testing and intend to install the sample model database, select the free **D1** tier. To learn more, see [Azure Analysis Services pricing](#).
 - **Administrator:** By default, this will be the account you are logged in with. You can choose a different account from your Azure Active Directory.
 - **Backup Storage setting:** Optional. If you already have a [storage account](#), you can specify it as the default for model database backup. You can also specify [backup and restore](#) settings later.
 - **Storage key expiration:** Optional. Specify a storage key expiration period.

3. Click **Create**.

Create usually takes under a minute. If you selected **Add to Portal**, navigate to your portal to see your new server. Or, navigate to **All services** > **Analysis Services** to see if your server is ready.

Clean up resources

When no longer needed, delete your server. In your server's **Overview**, click **Delete**.



Next steps

[Add a sample data model](#) to your server.

[Install an On-premises data gateway](#) if your data model connects to on-premises data sources.

[Deploy a tabular model project](#) from Visual Studio.

Create an Azure Analysis Services server by using PowerShell

4/18/2018 • 1 min to read • [Edit Online](#)

This quickstart describes using PowerShell from the command line to create an Azure Analysis Services server in an [Azure resource group](#) in your Azure subscription.

This task requires Azure PowerShell module version 4.0 or later. To find the version, run

```
Get-Module -ListAvailable AzureRM
```

. To install or upgrade, see [Install Azure PowerShell module](#).

NOTE

Creating a server might result in a new billable service. To learn more, see [Analysis Services pricing](#).

Before you begin

To complete this quickstart, you need:

- **Azure subscription:** Visit [Azure Free Trial](#) to create an account.
- **Azure Active Directory:** Your subscription must be associated with an Azure Active Directory tenant and you must have an account in that directory. To learn more, see [Authentication and user permissions](#).

Import AzureRm.AnalysisServices module

To create a server in your subscription, you use the [AzureRM.AnalysisServices](#) component module. Load the AzureRm.AnalysisServices module into your PowerShell session.

```
Import-Module AzureRM.AnalysisServices
```

Sign in to Azure

Sign in to your Azure subscription by using the [Connect-AzureRmAccount](#) command. Follow the on-screen directions.

```
Connect-AzureRmAccount
```

Create a resource group

An [Azure resource group](#) is a logical container where Azure resources are deployed and managed as a group. When you create your server, you must specify a resource group in your subscription. If you do not already have a resource group, you can create a new one by using the [New-AzureRmResourceGroup](#) command. The following example creates a resource group named `myResourceGroup` in the West US region.

```
New-AzureRmResourceGroup -Name "myResourceGroup" -Location "West US"
```

Create a server

Create a new server by using the [New-AzureRmAnalysisServicesServer](#) command. The following example creates a server named myServer in myResourceGroup, in the West US region, at the D1 tier, and specifies philipc@adventureworks.com as a server administrator.

```
New-AzureRmAnalysisServicesServer -ResourceGroupName "myResourceGroup" -Name "myServer" -Location West US -Sku D1 -Administrator "philipc@adventure-works.com"
```

Clean up resources

You can remove the server from your subscription by using the [Remove-AzureRmAnalysisServicesServer](#) command. If you continue with other quickstarts and tutorials in this collection, do not remove your server. The following example removes the server created in the previous step.

```
Remove-AzureRmAnalysisServicesServer -Name "myServer" -ResourceGroupName "myResourceGroup"
```

Next steps

[Manage Azure Analysis Services with PowerShell](#) [Deploy a model from SSDT](#) [Create a model in Azure portal](#)

Deploy a model from SSDT

4/13/2018 • 1 min to read • [Edit Online](#)

Once you've created a server in your Azure subscription, you're ready to deploy a tabular model database to it. You can use SQL Server Data Tools (SSDT) to build and deploy a tabular model project you're working on.

Prerequisites

To get started, you need:

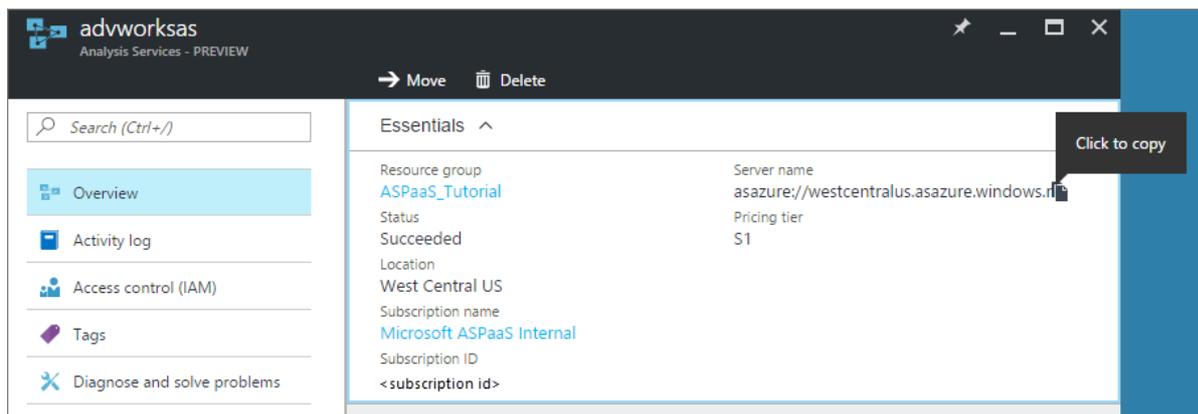
- **Analysis Services server** in Azure. To learn more, see [Create an Azure Analysis Services server](#).
- **Tabular model project** in SSDT or an existing tabular model at the 1200 or higher compatibility level. Never created one? Try the [Adventure Works Internet sales tabular modeling tutorial](#).
- **On-premises gateway** - If one or more data sources are on-premises in your organization's network, you need to install an [On-premises data gateway](#). The gateway is necessary for your server in the cloud connect to your on-premises data sources to process and refresh data in the model.

TIP

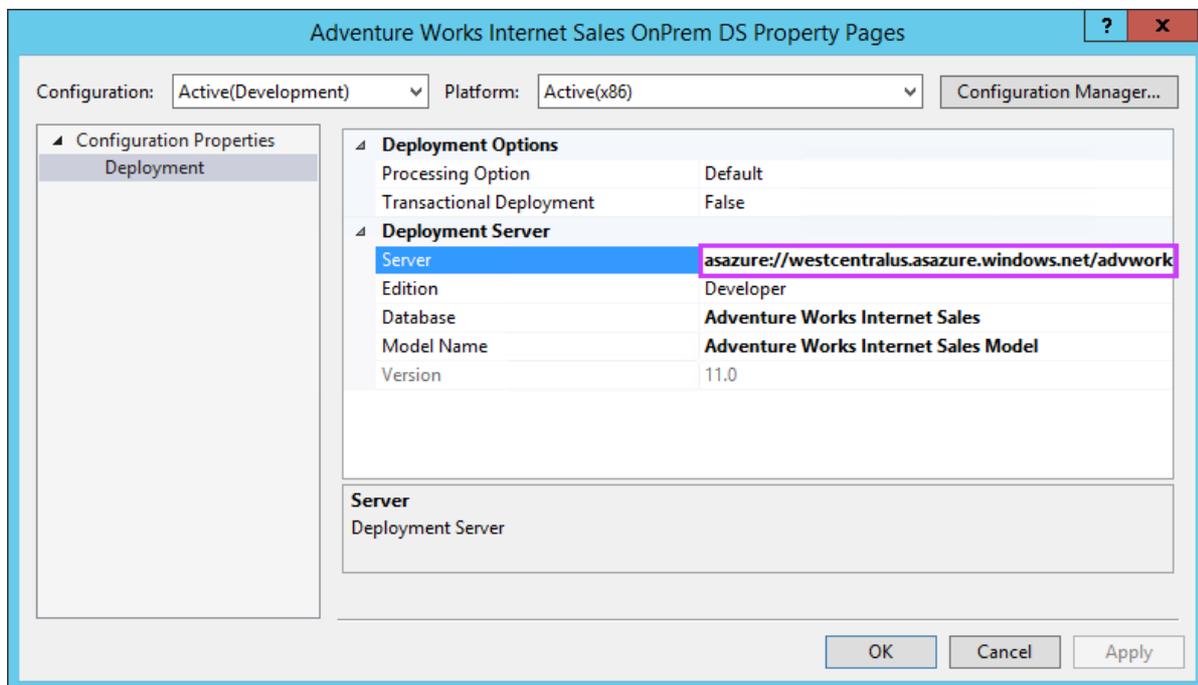
Before you deploy, make sure you can process the data in your tables. In SSDT, click **Model > Process > Process All**. If processing fails, you cannot successfully deploy.

To deploy a tabular model from SSDT

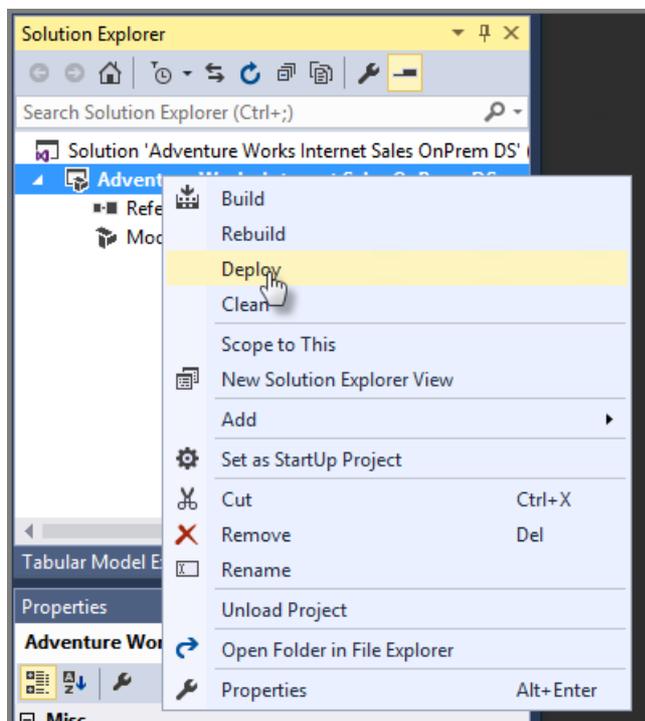
1. Before you deploy, you need to get the server name. In **Azure portal > server > Overview > Server name**, copy the server name.



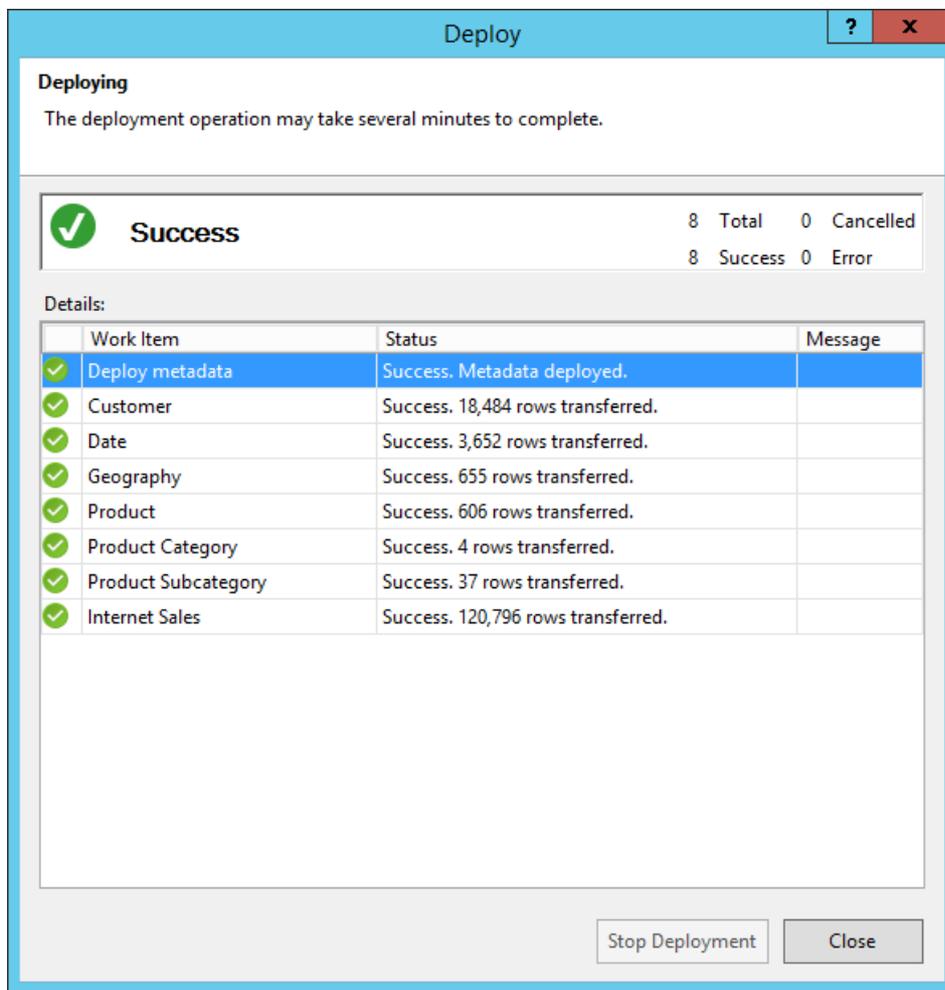
2. In SSDT > **Solution Explorer**, right-click the project > **Properties**. Then in **Deployment > Server** paste the server name.



3. In **Solution Explorer**, right-click **Properties**, then click **Deploy**. You may be prompted to sign in to Azure.



Deployment status appears in both the Output window and in Deploy.



That's all there is to it!

Troubleshooting

If deployment fails when deploying metadata, it's likely because SSDT couldn't connect to your server. Make sure you can connect to your server using SSMS. Then make sure the Deployment Server property for the project is correct.

If deployment fails on a table, it's likely because your server couldn't connect to a data source. If your data source is on-premises in your organization's network, be sure to install an [On-premises data gateway](#).

Next steps

Now that you have your tabular model deployed to your server, you're ready to connect to it. You can [connect to it with SSMS](#) to manage it. And, you can [connect to it using a client tool](#) like Power BI, Power BI Desktop, or Excel, and start creating reports.

Create a model in Azure portal

4/24/2018 • 1 min to read • [Edit Online](#)

The Azure Analysis Services web designer (preview) feature in Azure portal provides a quick and easy way to create and edit tabular models and query model data right in your browser.

Keep in mind, the web designer is **preview**. Functionality is limited. For more advanced model development and testing, it's best to use Visual Studio (SSDT) and SQL Server Management Studio (SSMS).

Before you begin

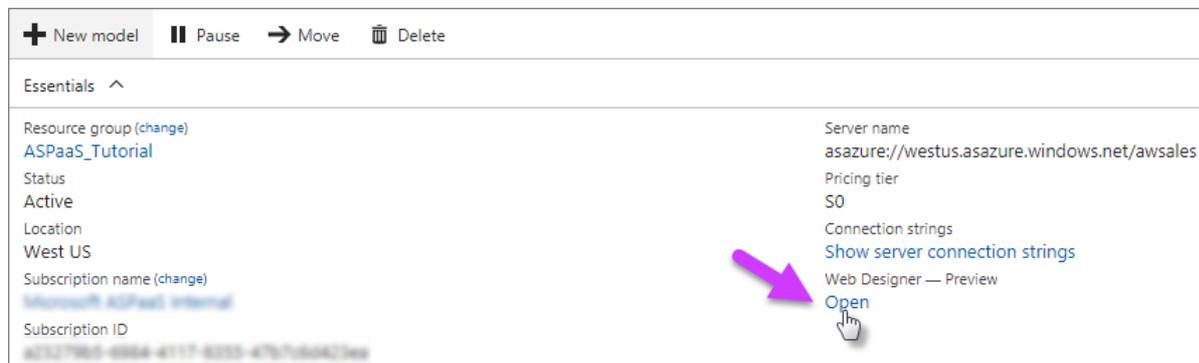
- An Azure Analysis Services server at the Standard or Developer tier. New models created by using the Web designer are DirectQuery, supported only by these tiers.
- An Azure SQL Database, Azure SQL Data Warehouse, or Power BI Desktop (.pbix) file as a datasource. New models created from Power BI Desktop files support Azure SQL Database and Azure SQL Data Warehouse.
- A SQL Server account and password for connecting to Azure SQL Database or Azure SQL Data Warehouse data sources.

Sign in to the Azure portal

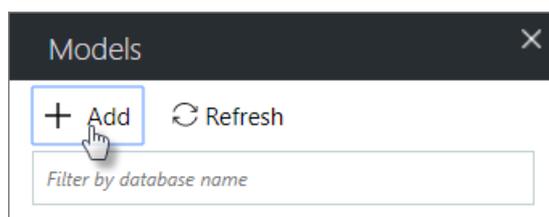
Sign in to the [Azure portal](#).

To create a new tabular model

1. In your server **Overview** > **Web designer**, click **Open**.



2. In **Web designer** > **Models**, click + **Add**.



3. In **New model**, type a model name, and then select a data source.

New model

Model Name

Regional sales

Where is your data?

Azure SQL Database

Azure SQL Data Warehouse

Power BI Desktop file

4. In **Connect**, enter the connection properties. Username and password must be a SQL Server account.

Connect

Connection option

Use SQL database from current subscription

SQL Server

adworkssql

Username

lizav

Password

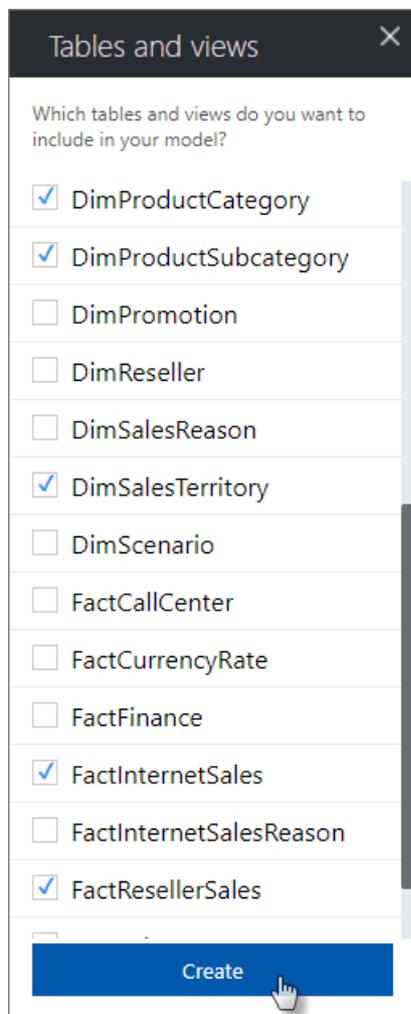
.....

Database

adworksdw

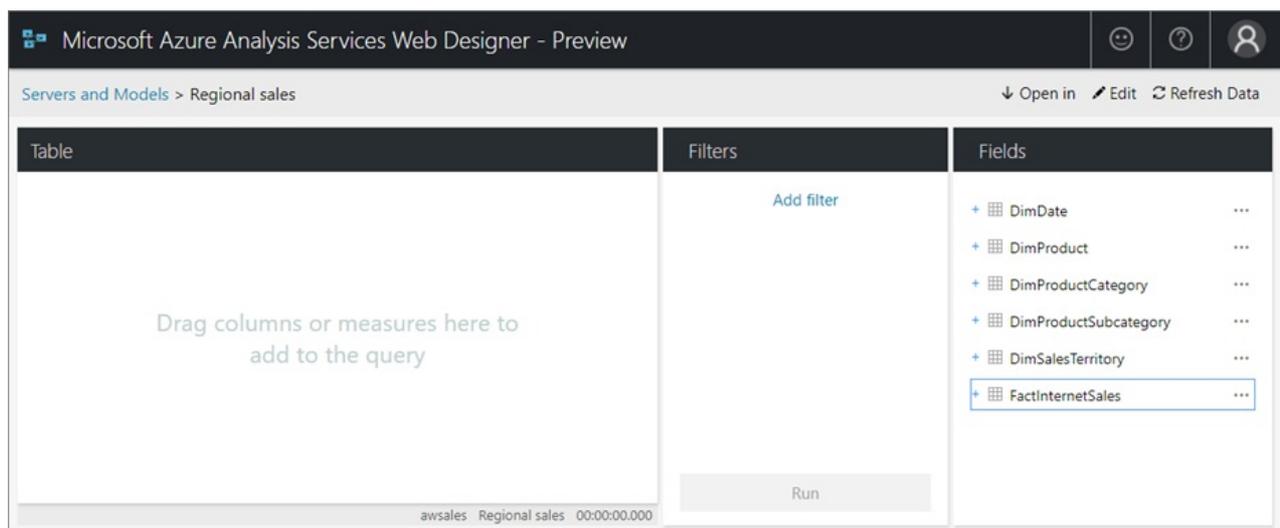
Next

5. In **Tables and views**, select the tables to include in your model, and then click **Create**. Relationships are created automatically between tables with a key pair.



Your new model appears in your browser. From here, you can:

- Query model data by dragging fields to the query designer and adding filters.
- Create new measures in tables.
- Edit model metadata by using the json editor.
- Open the model in Visual Studio (SSDT), Power BI Desktop, or Excel.



NOTE

When you edit model metadata or create new measures in your browser, you're saving those changes to your model in Azure. If you're also working on your model in SSDT, Power BI Desktop, or Excel, your model can get out of sync.

Next steps

[Manage database roles and users](#)

[Connect with Excel](#)

Data sources supported in Azure Analysis Services

4/13/2018 • 2 min to read • [Edit Online](#)

Data sources and connectors shown in Get Data or Import Wizard in Visual Studio are shown for both Azure Analysis Services and SQL Server Analysis Services. However, not all data sources and connectors shown are supported in Azure Analysis Services. The types of data sources you can connect to depend on many factors such as model compatibility level, available data connectors, authentication type, providers, and On-premises data gateway support.

Azure data sources

DATASOURCE	IN-MEMORY	DIRECTQUERY
Azure SQL Database	Yes	Yes
Azure SQL Data Warehouse	Yes	Yes
Azure Blob Storage*	Yes	No
Azure Table Storage*	Yes	No
Azure Cosmos DB*	Yes	No
Azure Data Lake Store*	Yes	No
Azure HDInsight HDFS*	Yes	No
Azure HDInsight Spark*	Yes	No

* Tabular 1400 models only.

Provider

In-memory and DirectQuery models connecting to Azure data sources use .NET Framework Data Provider for SQL Server.

On-premises data sources

Connecting to on-premises data sources from and Azure AS server require an On-premises gateway. When using a gateway, 64-bit providers are required.

In-memory and DirectQuery

DATASOURCE	IN-MEMORY PROVIDER	DIRECTQUERY PROVIDER
SQL Server	SQL Server Native Client 11.0, Microsoft OLE DB Provider for SQL Server, .NET Framework Data Provider for SQL Server	.NET Framework Data Provider for SQL Server

DATASOURCE	IN-MEMORY PROVIDER	DIRECTQUERY PROVIDER
SQL Server Data Warehouse	SQL Server Native Client 11.0, Microsoft OLE DB Provider for SQL Server, .NET Framework Data Provider for SQL Server	.NET Framework Data Provider for SQL Server
Oracle	Microsoft OLE DB Provider for Oracle, Oracle Data Provider for .NET	Oracle Data Provider for .NET
Teradata	OLE DB Provider for Teradata, Teradata Data Provider for .NET	Teradata Data Provider for .NET

In-memory only

DATASOURCE
Access Database
Active Directory*
Analysis Services
Analytics Platform System
Dynamics CRM*
Excel workbook
Exchange*
Folder*
JSON document*
Lines from binary*
MySQL Database
OData Feed*
ODBC query
OLE DB
Postgre SQL Database*
SAP HANA*
SAP Business Warehouse*
SharePoint*

DATASOURCE
Sybase Database
XML table*

* Tabular 1400 models only.

Specifying a different provider

Data models in Azure Analysis Services may require different data providers when connecting to certain data sources. In some cases, tabular models connecting to data sources using native providers such as SQL Server Native Client (SQLNCLI11) may return an error. If using native providers other than SQLOLEDB, you may see error message: **The provider 'SQLNCLI11.1' is not registered.** Or, if you have a DirectQuery model connecting to on-premises data sources and you use native providers, you may see error message: **Error creating OLE DB row set. Incorrect syntax near 'LIMIT'.**

When migrating an on-premises SQL Server Analysis Services tabular model to Azure Analysis Services, it may be necessary to change the provider.

To specify a provider

1. In SSDT > **Tabular Model Explorer** > **Data Sources**, right-click a data source connection, and then click **Edit Data Source**.
2. In **Edit Connection**, click **Advanced** to open the Advance properties window.
3. In **Set Advanced Properties** > **Providers**, then select the appropriate provider.

Impersonation

In some cases, it may be necessary to specify a different impersonation account. Impersonation account can be specified in Visual Studio (SSDT) or SSMS.

For on-premises data sources:

- If using SQL authentication, impersonation should be Service Account.
- If using Windows authentication, set Windows user/password. For SQL Server, Windows authentication with a specific impersonation account is supported only for in-memory data models.

For cloud data sources:

- If using SQL authentication, impersonation should be Service Account.

Next steps

[On-premises gateway](#)
[Manage your server](#)

Connecting to on-premises data sources with Azure On-premises Data Gateway

4/24/2018 • 9 min to read • [Edit Online](#)

The on-premises data gateway acts as a bridge, providing secure data transfer between on-premises data sources and your Azure Analysis Services servers in the cloud. In addition to working with multiple Azure Analysis Services servers in the same region, the latest version of the gateway also works with Azure Logic Apps, Power BI, Power Apps, and Microsoft Flow. You can associate multiple services in the same region with a single gateway.

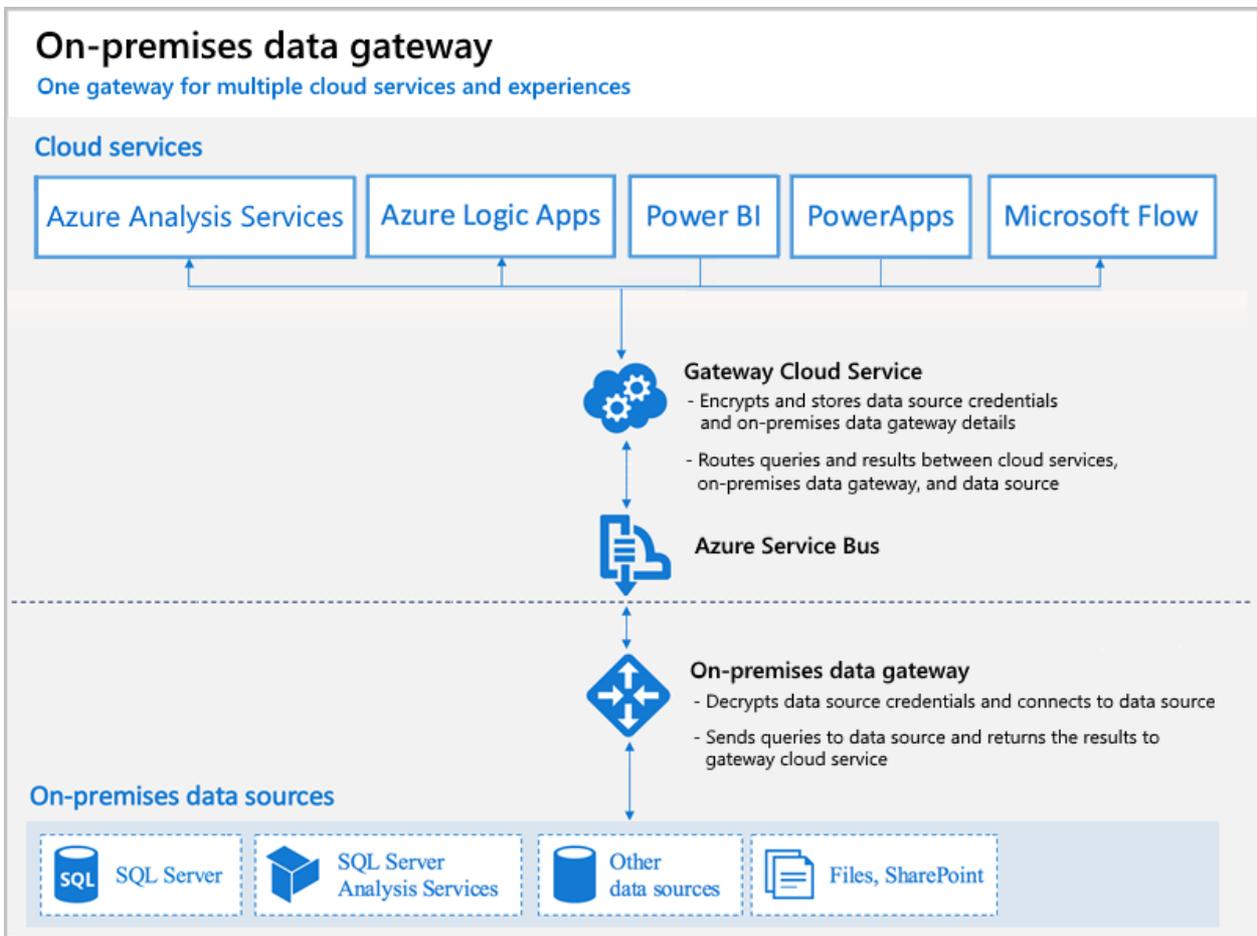
Getting setup with the gateway the first time is a four-part process:

- **Download and run setup** - This step installs a gateway service on a computer in your organization. You also sign in to Azure using an account in your [tenant's](#) Azure AD. Azure B2B (guest) accounts are not supported.
- **Register your gateway** - In this step, you specify a name and recovery key for your gateway and select a region, registering your gateway with the Gateway Cloud Service. Your gateway resource can be registered in any region, but we recommend it be in the same region as your Analysis Services servers.
- **Create a gateway resource in Azure** - In this step, you create a gateway resource in your Azure subscription.
- **Connect your servers to your gateway resource** - Once you have a gateway resource in your subscription, you can begin connecting your servers to it. You can connect multiple servers and other resources to it.

To get started right away, see [Install and configure on-premises data gateway](#).

How it works

The gateway you install on a computer in your organization runs as a Windows service, **On-premises data gateway**. This local service is registered with the Gateway Cloud Service through Azure Service Bus. You then create a gateway resource Gateway Cloud Service for your Azure subscription. Your Azure Analysis Services servers are then connected to your gateway resource. When models on your server need to connect to your on-premises data sources for queries or processing, a query and data flow traverses the gateway resource, Azure Service Bus, the local on-premises data gateway service, and your data sources.



Queries and data flow:

1. A query is created by the cloud service with the encrypted credentials for the on-premises data source. It's then sent to a queue for the gateway to process.
2. The gateway cloud service analyzes the query and pushes the request to the [Azure Service Bus](#).
3. The on-premises data gateway polls the Azure Service Bus for pending requests.
4. The gateway gets the query, decrypts the credentials, and connects to the data sources with those credentials.
5. The gateway sends the query to the data source for execution.
6. The results are sent from the data source, back to the gateway, and then onto the cloud service and your server.

Windows Service account

The on-premises data gateway is configured to use `NT SERVICE\PBIgwService` for the Windows service logon credential. By default, it has the right of Logon as a service; in the context of the machine that you are installing the gateway on. This credential is not the same account used to connect to on-premises data sources or your Azure account.

If you encounter issues with your proxy server due to authentication, you may want to change the Windows service account to a domain user or managed service account.

Ports

The gateway creates an outbound connection to Azure Service Bus. It communicates on outbound ports: TCP 443 (default), 5671, 5672, 9350 through 9354. The gateway does not require inbound ports.

We recommend you whitelist the IP addresses for your data region in your firewall. You can download the [Microsoft Azure Datacenter IP list](#). This list is updated weekly.

NOTE

The IP Addresses listed in the Azure Datacenter IP list are in CIDR notation. To learn more, see [Classless Inter-Domain Routing](#).

The following are the fully qualified domain names used by the gateway.

DOMAIN NAMES	OUTBOUND PORTS	DESCRIPTION
*.powerbi.com	80	HTTP used to download the installer.
*.powerbi.com	443	HTTPS
*.analysis.windows.net	443	HTTPS
*.login.windows.net	443	HTTPS
*.servicebus.windows.net	5671-5672	Advanced Message Queuing Protocol (AMQP)
*.servicebus.windows.net	443, 9350-9354	Listeners on Service Bus Relay over TCP (requires 443 for Access Control token acquisition)
*.frontend.clouddatahub.net	443	HTTPS
*.core.windows.net	443	HTTPS
login.microsoftonline.com	443	HTTPS
*.msftncsi.com	443	Used to test internet connectivity if the gateway is unreachable by the Power BI service.
*.microsoftonline-p.com	443	Used for authentication depending on configuration.

Forcing HTTPS communication with Azure Service Bus

You can force the gateway to communicate with Azure Service Bus by using HTTPS instead of direct TCP; however, doing so can greatly reduce performance. You can modify the *Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config* file by changing the value from `AutoDetect` to `Https`. This file is typically located at `C:\Program Files\On-premises data gateway`.

```
<setting name="ServiceBusSystemConnectivityModeString" serializeAs="String">  
  <value>Https</value>  
</setting>
```

Tenant level administration

There is currently no single place where tenant administrators can manage all the gateways that other users have installed and configured. If you're a tenant administrator, it's recommended you ask users in your organization to add you as an administrator to every gateway they install. This allows you to manage all the gateways in your organization through the Gateway Settings page or through [PowerShell commands](#).

Frequently asked questions

General

Q: Do I need a gateway for data sources in the cloud, such as Azure SQL Database?

A: No. A gateway is necessary for connecting to on-premises data sources only.

Q: Does the gateway have to be installed on the same machine as the data source?

A: No. The gateway just needs the capability to connect to the server, typically on the same network.

Q: Why do I need to use a work or school account to sign in?

A: You can only use an organizational work or school account when you install the on-premises data gateway. And, that account must be in the same tenant as the subscription you are configuring the gateway resource in. Your sign-in account is stored in a tenant that's managed by Azure Active Directory (Azure AD). Usually, your Azure AD account's user principal name (UPN) matches the email address.

Q: Where are my credentials stored?

A: The credentials that you enter for a data source are encrypted and stored in the Gateway Cloud Service. The credentials are decrypted at the on-premises data gateway.

Q: Are there any requirements for network bandwidth?

A: It's recommended your network connection has good throughput. Every environment is different, and the amount of data being sent affects the results. Using ExpressRoute could help to guarantee a level of throughput between on-premises and the Azure datacenters. You can use the third-party tool Azure Speed Test app to help gauge your throughput.

Q: What is the latency for running queries to a data source from the gateway? What is the best architecture?

A: To reduce network latency, install the gateway as close to the data source as possible. If you can install the gateway on the actual data source, this proximity minimizes the latency introduced. Consider the datacenters too. For example, if your service uses the West US datacenter, and you have SQL Server hosted in an Azure VM, your Azure VM should be in the West US too. This proximity minimizes latency and avoids egress charges on the Azure VM.

Q: How are results sent back to the cloud?

A: Results are sent through the Azure Service Bus.

Q: Are there any inbound connections to the gateway from the cloud?

A: No. The gateway uses outbound connections to Azure Service Bus.

Q: What if I block outbound connections? What do I need to open?

A: See the ports and hosts that the gateway uses.

Q: What is the actual Windows service called?

A: In Services, the gateway is called On-premises data gateway service.

Q: Can the gateway Windows service run with an Azure Active Directory account?

A: No. The Windows service must have a valid Windows account. By default, the service runs with the Service SID, NT SERVICE\PBIEgwService.

Q: How do I takeover a gateway?

A: To takeover a gateway (by running Setup/Change in Control Panel > Programs), you need to be an Owner for the gateway resource in Azure and have the recovery key. Gateway resource Owners are configurable in Access Control.

High availability and disaster recovery

Q: What options are available for disaster recovery?

A: You can use the recovery key to restore or move a gateway. When you install the gateway, specify the recovery key.

Q: What is the benefit of the recovery key?

A: The recovery key provides a way to migrate or recover your gateway settings after a disaster.

Troubleshooting

Q: Why don't I see my gateway in the list of gateway instances when trying to create the gateway resource in Azure?

A: There are two possible reasons. First is a resource is already created for the gateway in the current or some other subscription. To eliminate that possibility, enumerate resources of the type **On-premises Data Gateways** from the portal. Make sure to select all the subscriptions when enumerating all the resources. Once the resource is created, the gateway does not appear in the list of gateway instances in the Create Gateway Resource portal experience. The second possibility is that the Azure AD identity of the user who installed the gateway is different from the user signed in to Azure portal. To resolve, sign in to the portal using the same account as the user who installed the gateway.

Q: How can I see what queries are being sent to the on-premises data source?

A: You can enable query tracing, which includes the queries that are sent. Remember to change query tracing back to the original value when done troubleshooting. Leaving query tracing turned on creates larger logs.

You can also look at tools that your data source has for tracing queries. For example, you can use Extended Events or SQL Profiler for SQL Server and Analysis Services.

Q: Where are the gateway logs?

A: See Logs later in this article.

Update to the latest version

Many issues can surface when the gateway version becomes outdated. As good general practice, make sure that you use the latest version. If you haven't updated the gateway for a month or longer, you might consider installing the latest version of the gateway, and see if you can reproduce the issue.

Error: Failed to add user to group. (-2147463168 PBIEgwService Performance Log Users)

You might get this error if you try to install the gateway on a domain controller, which isn't supported. Make sure that you deploy the gateway on a machine that isn't a domain controller.

Logs

Log files are an important resource when troubleshooting.

Enterprise gateway service logs

```
C:\Users\PBIEgwService\AppData\Local\Microsoft\On-premises data gateway\<yyyymmdd>.<Number>.log
```

Configuration logs

```
C:\Users\<username>\AppData\Local\Microsoft\On-premises data gateway\GatewayConfigurator.log
```

Event logs

You can find the Data Management Gateway and PowerBIGateway logs under **Application and Services Logs**.

Telemetry

Telemetry can be used for monitoring and troubleshooting. By default

To turn on telemetry

1. Check the On-premises data gateway client directory on the computer. Typically, it is **%systemdrive%\Program Files\On-premises data gateway**. Or, you can open a Services console and check the Path to executable: A property of the On-premises data gateway service.
2. In the Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config file from client directory.

Change the SendTelemetry setting to true.

```
<setting name="SendTelemetry" serializeAs="String">  
    <value>true</value>  
</setting>
```

3. Save your changes and restart the Windows service: On-premises data gateway service.

Next steps

- [Install and configure on-premises data gateway.](#)
- [Manage Analysis Services](#)
- [Get data from Azure Analysis Services](#)

Connecting to servers

4/24/2018 • 1 min to read • [Edit Online](#)

This article describes connecting to a server by using data modeling and management applications like SQL Server Management Studio (SSMS) or SQL Server Data Tools (SSDT). Or, with client reporting applications like Microsoft Excel, Power BI Desktop, or custom applications. Connections to Azure Analysis Services use HTTPS.

Client libraries

[Get the latest Client libraries](#)

All connections to a server, regardless of type, require updated AMO, ADOMD.NET, and OLEDB client libraries to connect to and interface with an Analysis Services server. For SSMS, SSDT, Excel 2016, and Power BI, the latest client libraries are installed or updated with monthly releases. However, in some cases, it's possible an application may not have the latest. For example, when policies delay updates, or Office 365 updates are on the Deferred Channel.

Server name

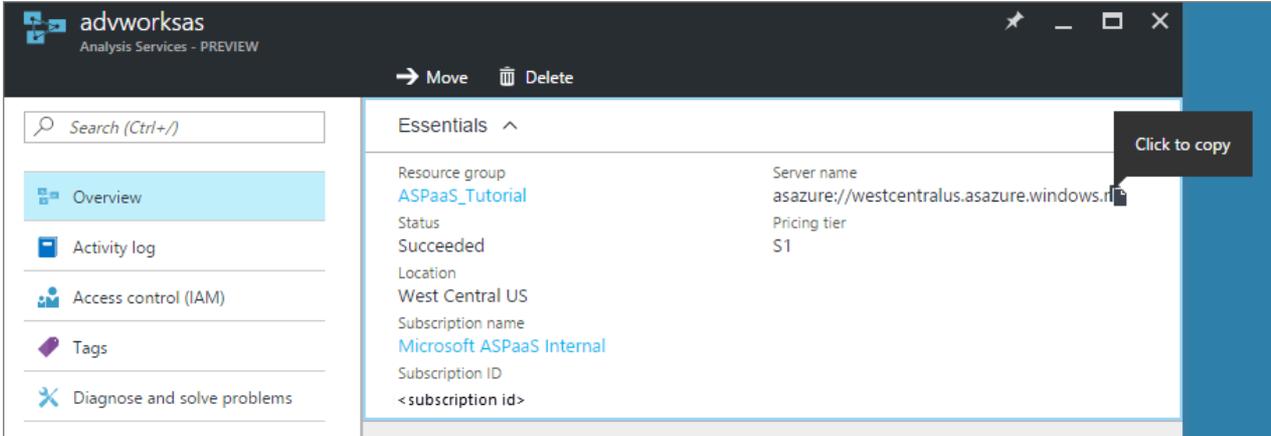
When you create an Analysis Services server in Azure, you specify a unique name and the region where the server is to be created. When specifying the server name in a connection, the server naming scheme is:

```
<protocol>://<region>/<servername>
```

Where protocol is string **asazure**, region is the Uri where the server was created (for example, westus.asazure.windows.net) and servername is the name of your unique server within the region.

Get the server name

In **Azure portal** > server > **Overview** > **Server name**, copy the entire server name. If other users in your organization are connecting to this server too, you can share this server name with them. When specifying a server name, the entire path must be used.



The screenshot shows the Azure portal interface for an Analysis Services server. The 'Essentials' section displays the following information:

Resource group	ASPaaS_Tutorial	Server name	asazure://westcentralus.asazure.windows.net
Status	Succeeded	Pricing tier	S1
Location	West Central US		
Subscription name	Microsoft ASPaaS Internal		
Subscription ID	<subscription id>		

Connection string

When connecting to Azure Analysis Services using the Tabular Object Model, use the following connection string formats:

Integrated authentication picks up the Azure Active Directory credential cache if available. If not, the Azure login window is shown.

```
"Provider=MSOLAP;Data Source=<Azure AS instance name>;"
```

Azure Active Directory authentication with username and password

```
"Provider=MSOLAP;Data Source=<Azure AS instance name>;User ID=<user name>;Password=<password>;Persist Security Info=True; Impersonation Level=Impersonate;"
```

Windows authentication (Integrated security)

Use the Windows account running the current process.

```
"Provider=MSOLAP;Data Source=<Azure AS instance name>; Integrated Security=SSPI;Persist Security Info=True;"
```

Connect using an .odc file

With older versions of Excel, users can connect to an Azure Analysis Services server by using an Office Data Connection (.odc) file. To learn more, see [Create an Office Data Connection \(.odc\) file](#).

Next steps

[Connect with Excel](#)

[Connect with Power BI](#)

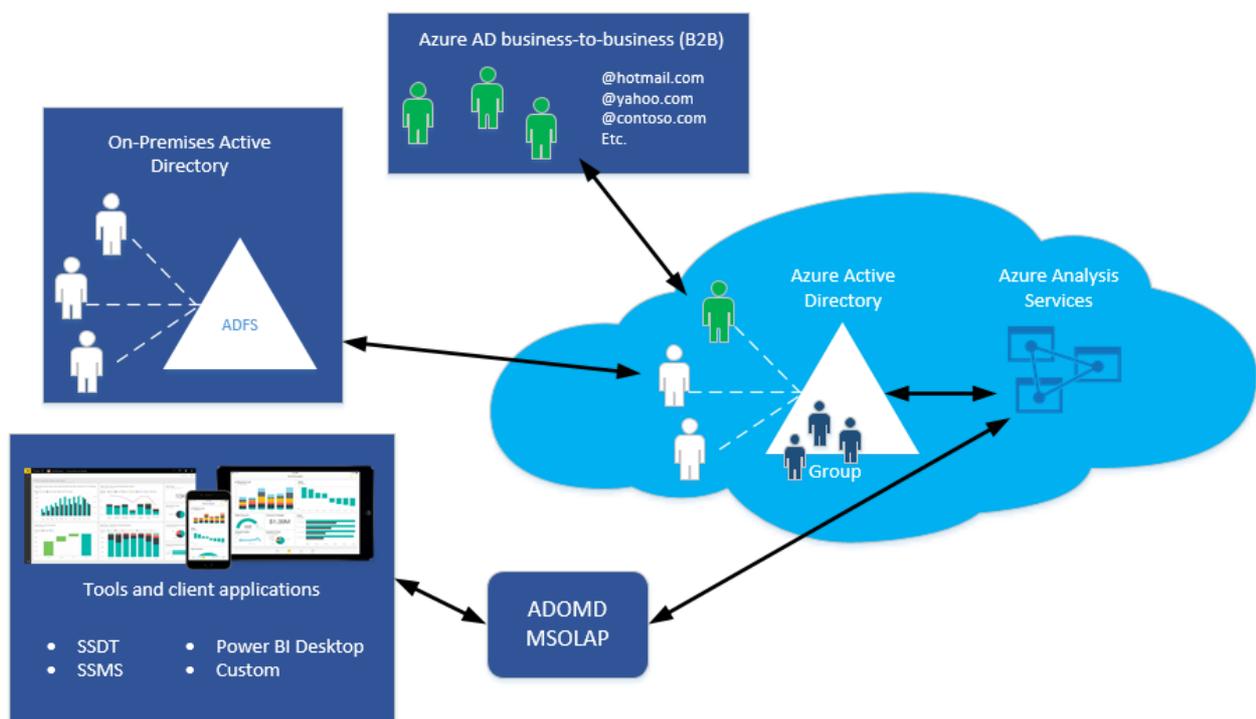
[Manage your server](#)

Authentication and user permissions

4/13/2018 • 5 min to read • [Edit Online](#)

Azure Analysis Services uses Azure Active Directory (Azure AD) for identity management and user authentication. Any user creating, managing, or connecting to an Azure Analysis Services server must have a valid user identity in an [Azure AD tenant](#) in the same subscription.

Azure Analysis Services supports [Azure AD B2B collaboration](#). With B2B, users from outside an organization can be invited as guest users in an Azure AD directory. Guests can be from another Azure AD tenant directory or any valid email address. Once invited and the user accepts the invitation sent by email from Azure, the user identity is added to the tenant directory. Those identities can be added to security groups or as members of a server administrator or database role.



Authentication

All client applications and tools use one or more of the Analysis Services [client libraries](#) (AMO, MSOLAP, ADOMD) to connect to a server.

All three client libraries support both Azure AD interactive flow, and non-interactive authentication methods. The two non-interactive methods, Active Directory Password and Active Directory Integrated Authentication methods can be used in applications utilizing AMOMD and MSOLAP. These two methods never result in pop-up dialog boxes.

Client applications like Excel and Power BI Desktop, and tools like SSMS and SSDT install the latest versions of the libraries when updated to the latest release. Power BI Desktop, SSMS, and SSDT are updated monthly. Excel is [updated with Office 365](#). Office 365 updates are less frequent, and some organizations use the deferred channel, meaning updates are deferred up to three months.

Depending on the client application or tool you use, the type of authentication and how you sign in may be different. Each application may support different features for connecting to cloud services like Azure Analysis Services.

Power BI Desktop, SSDT, and SSMS support Active Directory Universal Authentication, an interactive method that also supports Azure Multi-Factor Authentication (MFA). Azure MFA helps safeguard access to data and applications while providing a simple sign-in process. It delivers strong authentication with several verification options (phone call, text message, smart cards with pin, or mobile app notification). Interactive MFA with Azure AD can result in a pop-up dialog box for validation. **Universal Authentication is recommended.**

If signing in to Azure by using a Windows account, and Universal Authentication is not selected or available (Excel), [Active Directory Federation Services \(AD FS\)](#) is required. With Federation, Azure AD and Office 365 users are authenticated using on-premises credentials and can access Azure resources.

SQL Server Management Studio (SSMS)

Azure Analysis Services servers support connections from [SSMS V17.1](#) and higher by using Windows Authentication, Active Directory Password Authentication, and Active Directory Universal Authentication. In general, it's recommended you use Active Directory Universal Authentication because:

- Supports interactive and non-interactive authentication methods.
- Supports Azure B2B guest users invited into the Azure AS tenant. When connecting to a server, guest users must select Active Directory Universal Authentication when connecting to the server.
- Supports Multi-Factor Authentication (MFA). Azure MFA helps safeguard access to data and applications with a range of verification options: phone call, text message, smart cards with pin, or mobile app notification. Interactive MFA with Azure AD can result in a pop-up dialog box for validation.

SQL Server Data Tools (SSDT)

SSDT connects to Azure Analysis Services by using Active Directory Universal Authentication with MFA support. Users are prompted to sign in to Azure on the first deployment. Users must sign in to Azure with an account with server administrator permissions on the server they are deploying to. When signing in to Azure the first time, a token is assigned. SSDT caches the token in-memory for future reconnects.

Power BI Desktop

Power BI Desktop connects to Azure Analysis Services using Active Directory Universal Authentication with MFA support. Users are prompted to sign in to Azure on the first connection. Users must sign in to Azure with an account that is included in a server administrator or database role.

Excel

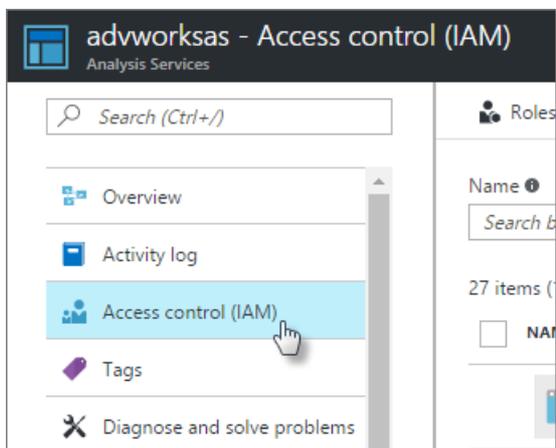
Excel users can connect to a server by using a Windows account, an organization ID (email address), or an external email address. External email identities must exist in the Azure AD as a guest user.

User permissions

Server administrators are specific to an Azure Analysis Services server instance. They connect with tools like Azure portal, SSMS, and SSDT to perform tasks like adding databases and managing user roles. By default, the user that creates the server is automatically added as an Analysis Services server administrator. Other administrators can be added by using Azure portal or SSMS. Server administrators must have an account in the Azure AD tenant in the same subscription. To learn more, see [Manage server administrators](#).

Database users connect to model databases by using client applications like Excel or Power BI. Users must be added to database roles. Database roles define administrator, process, or read permissions for a database. It's important to understand database users in a role with administrator permissions is different than server administrators. However, by default, server administrators are also database administrators. To learn more, see [Manage database roles and users](#).

Azure resource owners. Resource owners manage resources for an Azure subscription. Resource owners can add Azure AD user identities to Owner or Contributor Roles within a subscription by using **Access control** in Azure portal, or with Azure Resource Manager templates.



Roles at this level apply to users or accounts that need to perform tasks that can be completed in the portal or by using Azure Resource Manager templates. To learn more, see [Role-Based Access Control](#).

Database roles

Roles defined for a tabular model are database roles. That is, the roles contain members consisting of Azure AD users and security groups that have specific permissions that define the action those members can take on a model database. A database role is created as a separate object in the database, and applies only to the database in which that role is created.

By default, when you create a new tabular model project, the model project does not have any roles. Roles can be defined by using the Role Manager dialog box in SSDT. When roles are defined during model project design, they are applied only to the model workspace database. When the model is deployed, the same roles are applied to the deployed model. After a model has been deployed, server and database administrators can manage roles and members by using SSMS. To learn more, see [Manage database roles and users](#).

Next steps

[Manage access to resources with Azure Active Directory groups](#)

[Manage database roles and users](#)

[Manage server administrators](#)

[Role-Based Access Control](#)

Compatibility level for Analysis Services tabular models

4/13/2018 • 1 min to read • [Edit Online](#)

Compatibility level refers to release-specific behaviors in the Analysis Services engine. Changes to the compatibility level typically coincide with major releases of SQL Server. These changes are also implemented in Azure Analysis Services to maintain parity between both platforms. Compatibility level changes also affect features available in your tabular models. For example, DirectQuery and tabular object metadata have different implementations depending on the compatibility level.

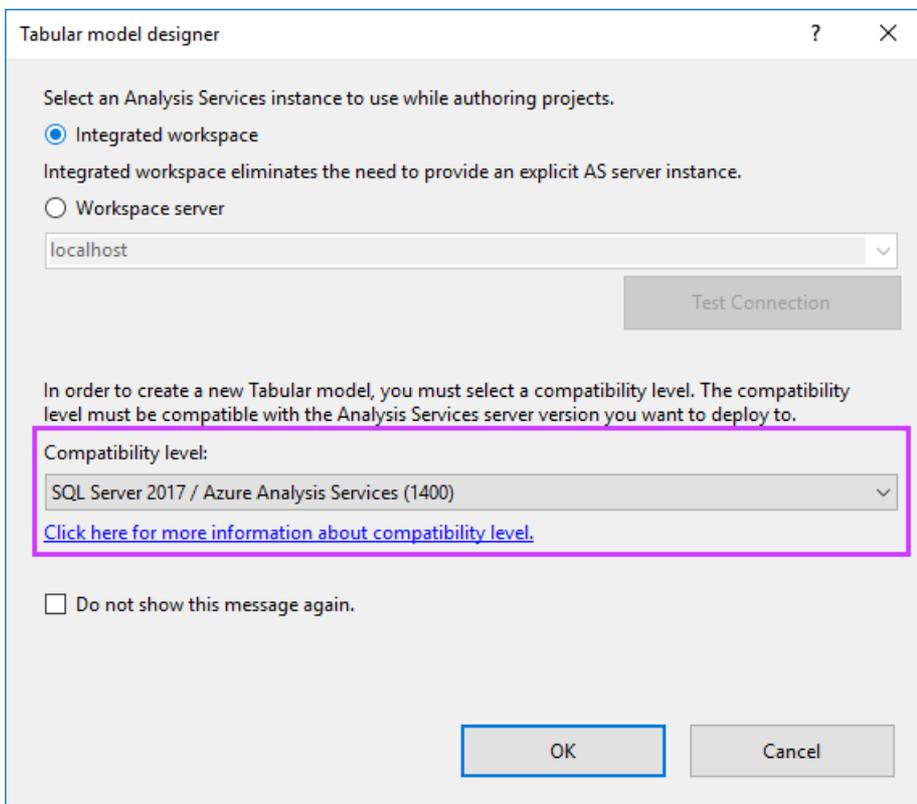
Azure Analysis Services supports tabular models at the 1200 and 1400 compatibility levels.

The latest compatibility level is 1400. This level coincides with SQL Server 2017 Analysis Services. Major features in the 1400 compatibility level include:

- New infrastructure for data connectivity and import into tabular models with support for TOM APIs and TMSL scripting. This new feature enables support for additional data sources such as Azure Blob storage.
- Data transformation and data mashup capabilities by using Get Data and M expressions.
- Measures support a Detail Rows property with a DAX expression. This property enables client tools like Microsoft Excel to drill down to detailed data from an aggregated report. For example, when users view total sales for a region and month, they can view the associated order details.
- Object-level security for table and column names, in addition to the data within them.
- Enhanced support for ragged hierarchies.
- Performance and monitoring improvements.

Set compatibility level

When creating a new tabular model project in SSDT, you can specify the compatibility level on the **Tabular model designer** dialog.



If you select the **Do not show this message again** option, all subsequent projects use the compatibility level you specified as the default. You can change the default compatibility level in SSDT in **Tools > Options**.

To upgrade an existing tabular model project in SSDT, set the **Compatibility Level** property in the model **Properties** window. Keep in-mind, upgrading the compatibility level is irreversible.

Check compatibility level for a tabular model database in SQL Server Management Studio

In SSMS, right-click the database name > **Properties** > **Compatibility Level**.

Check supported compatibility level for a server in SSMS

In SSMS, right-click the server name > **Properties** > **Supported Compatibility Level**.

This property specifies the highest compatibility level of a database that will run on the server (excluding preview). The supported compatibility level cannot be changed.

Next steps

[Create a model in Azure portal](#)

[Manage Analysis Services](#)

Analysis Services high availability

4/13/2018 • 1 min to read • [Edit Online](#)

This article describes assuring high availability for Azure Analysis Services servers.

Assuring high availability during a service disruption

While rare, an Azure data center can have an outage. When an outage occurs, it causes a business disruption that might last a few minutes or might last for hours. High availability is most often achieved with server redundancy. With Azure Analysis Services, you can achieve redundancy by creating additional, secondary servers in one or more regions. When creating redundant servers, to assure the data and metadata on those servers is in-sync with the server in a region that has gone offline, you can:

- Deploy models to redundant servers in other regions. This method requires processing data on both your primary server and redundant servers in-parallel, assuring all servers are in-sync.
- [Backup](#) databases from your primary server and restore on redundant servers. For example, you can automate nightly backups to Azure storage, and restore to other redundant servers in other regions.

In either case, if your primary server experiences an outage, you must change the connection strings in reporting clients to connect to the server in a different regional datacenter. This change should be considered a last resort and only if a catastrophic regional data center outage occurs. It's more likely a data center outage hosting your primary server would come back online before you could update connections on all clients.

To avoid having to change connection strings on reporting clients, you can create a server [alias](#) for your primary server. If the primary server goes down, you can change the alias to point to a redundant server in another region. You can automate alias to server name by coding an endpoint health check on the primary server. If the health check fails, the same endpoint can direct to a redundant server in another region.

Related information

[Backup and restore](#)

[Manage Azure Analysis Services](#)

[Alias server names](#)

Install and configure an on-premises data gateway

4/13/2018 • 3 min to read • [Edit Online](#)

An on-premises data gateway is required when one or more Azure Analysis Services servers in the same region connect to on-premises data sources. To learn more about the gateway, see [On-premises data gateway](#).

Prerequisites

Minimum Requirements:

- .NET 4.5 Framework
- 64-bit version of Windows 7 / Windows Server 2008 R2 (or later)

Recommended:

- 8 Core CPU
- 8 GB Memory
- 64-bit version of Windows 2012 R2 (or later)

Important considerations:

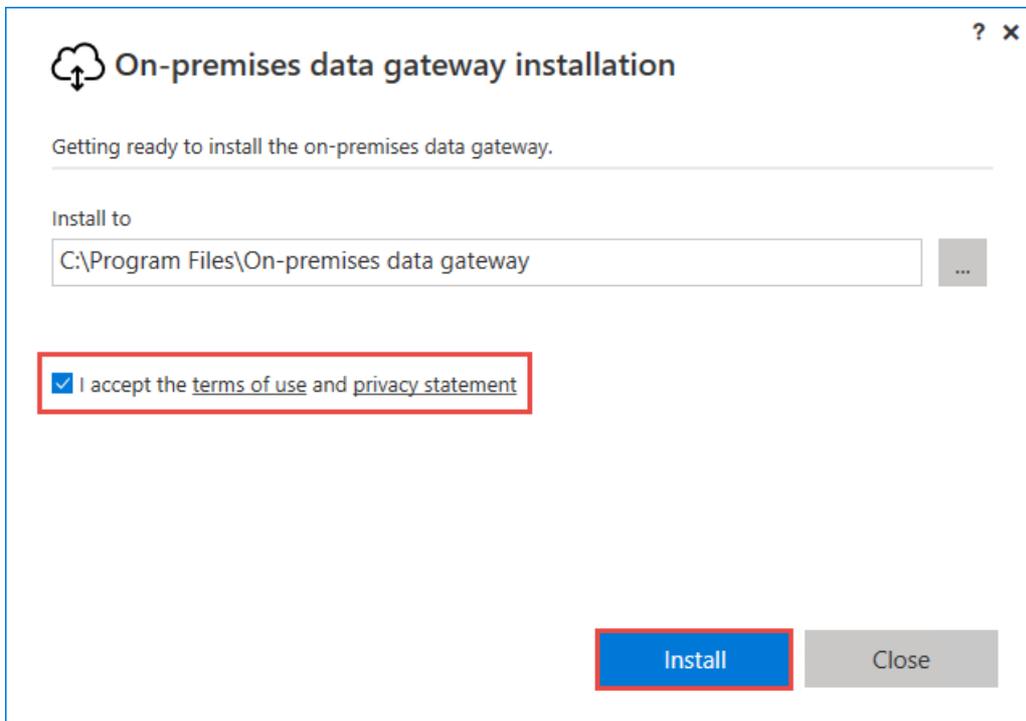
- During setup, when registering your gateway with Azure, the default region for your subscription is selected. You can choose a different region. If you have servers in more than one region, you must install a gateway for each region.
- The gateway cannot be installed on a domain controller.
- Only one gateway can be installed on a single computer.
- Install the gateway on a computer that remains on and does not go to sleep.
- Do not install the gateway on a computer wirelessly connected to your network. Performance can be diminished.
- Sign in to Azure with an account in Azure AD for the same [tenant](#) as the subscription you are registering the gateway in. Azure B2B (guest) accounts are not supported when installing and registering a gateway.
- The (unified) gateway described here is not supported in Azure Government, Azure Germany, and Azure China sovereign regions. Use **Dedicated On-premises gateway for Azure Analysis Services**, installed from your server's **Quick Start** in the portal.

Download

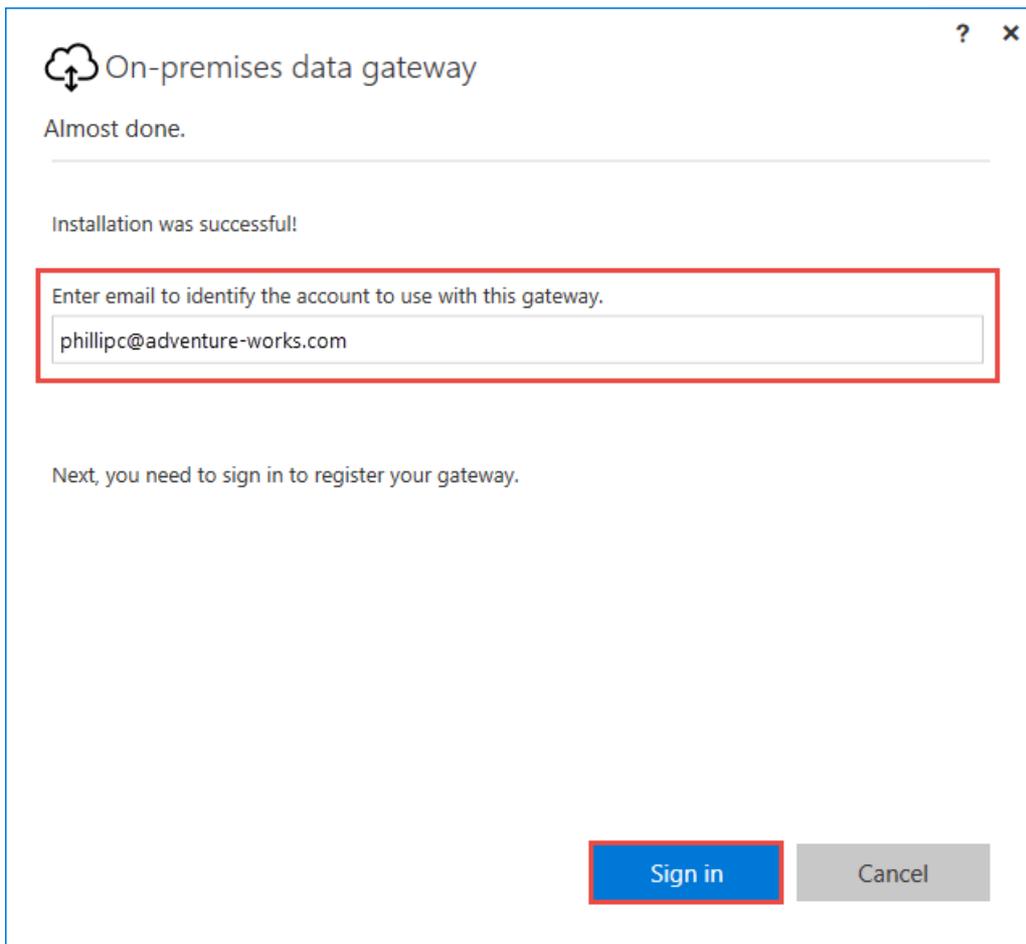
[Download the gateway](#)

Install

1. Run setup.
2. Select a location, accept the terms, and then click **Install**.



3. Sign in to Azure. The account must be in your tenant's Azure Active Directory. This account is used for the gateway administrator. Azure B2B (guest) accounts are not supported when installing and registering the gateway.



NOTE

If you sign in with a domain account, it's mapped to your organizational account in Azure AD. Your organizational account is used as the gateway administrator.

Register

In order to create a gateway resource in Azure, you must register the local instance you installed with the Gateway Cloud Service.

1. Select **Register a new gateway on this computer.**

On-premises data gateway

You are signed in as phillipc@adventure-works.com and are ready to register the gateway.

Register a new gateway on this computer.

Migrate, restore, or takeover an existing gateway.

- Move a gateway to a new computer
- Recover a damaged gateway
- Take ownership of a gateway

The old gateway will be disconnected.

Next Cancel

2. Type a name and recovery key for your gateway. By default, the gateway uses your subscription's default region. If you need to select a different region, select **Change Region**.

IMPORTANT

Save your recovery key in a safe place. The recovery key is required in-order to takeover, migrate, or restore a gateway.

On-premises data gateway

You are signed in as phillipc@adventure-works.com and are ready to register the gateway.

New on-premises data gateway name
West Central US Gateway

Recovery key (8 character minimum)
.....

ⓘ This key is needed to restore the gateway and can't be changed. Record it in a safe place.

Confirm recovery key
.....

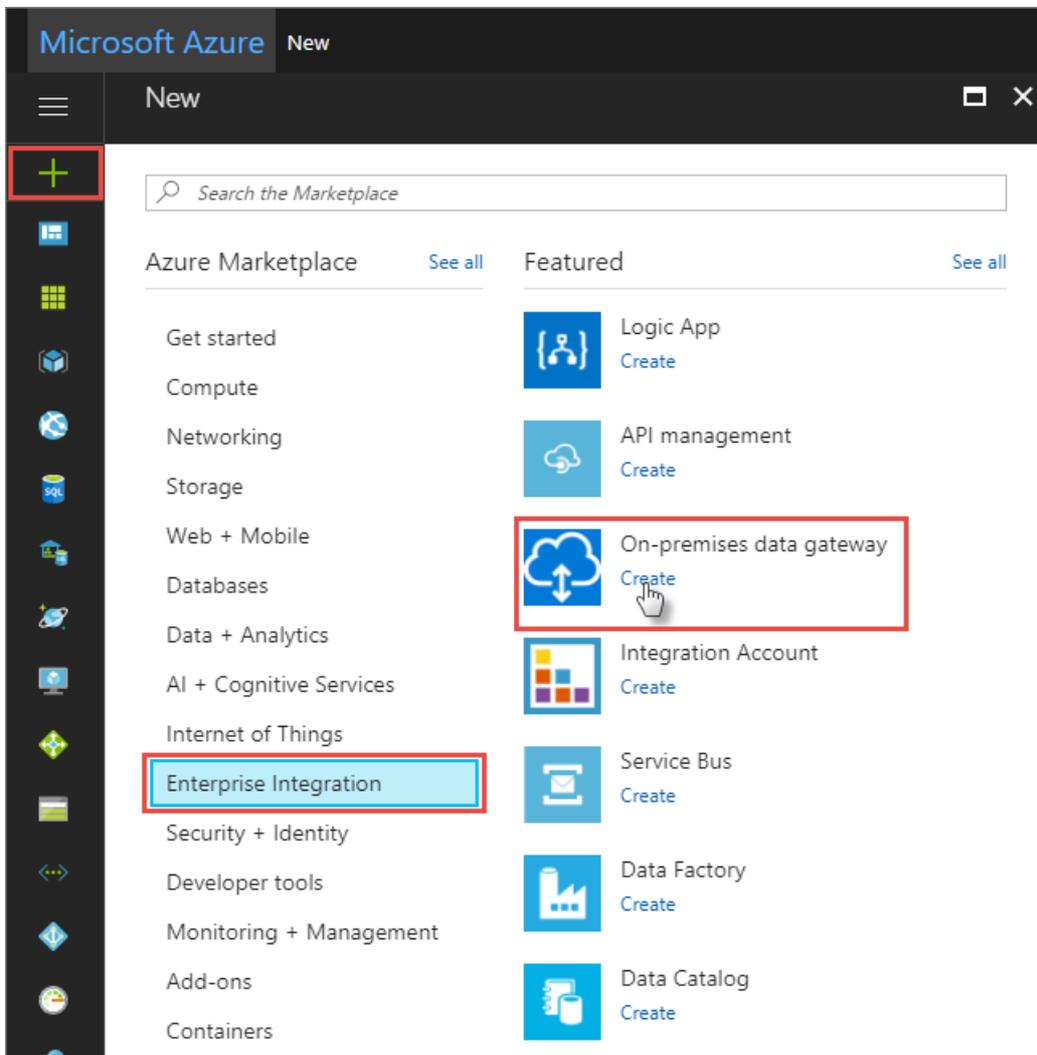
We'll use this region to connect the gateway to cloud services: [West Central US Change Region](#)

<< Back Configure Cancel

Create an Azure gateway resource

After you've installed and registered your gateway, you need to create a gateway resource in your Azure subscription. Sign in to Azure with the same account you used when registering the gateway.

1. In Azure portal, click **Create a new service** > **Enterprise Integration** > **On-premises data gateway** > **Create**.



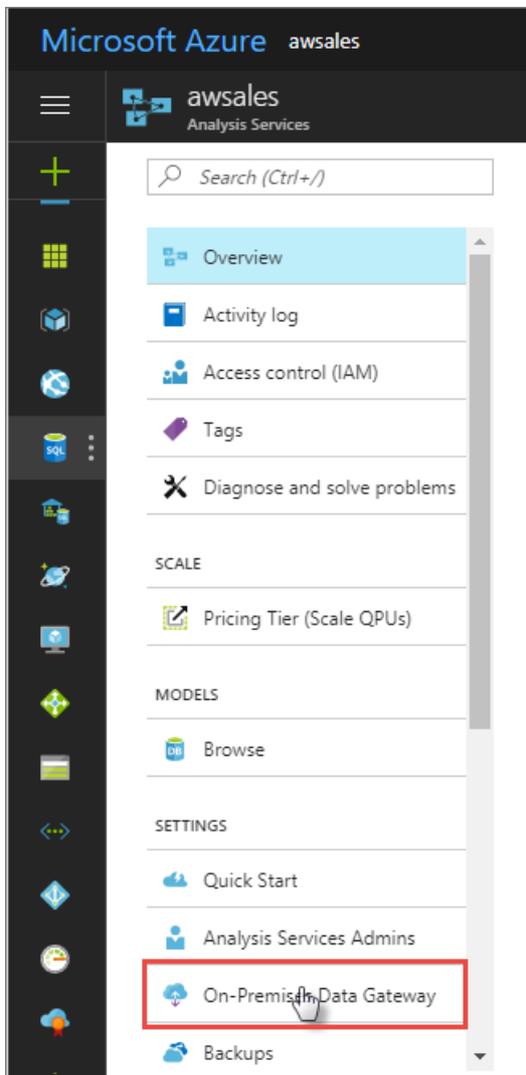
2. In **Create connection gateway**, enter these settings:

- **Name:** Enter a name for your gateway resource.
- **Subscription:** Select the Azure subscription to associate with your gateway resource.
The default subscription is based on the Azure account that you used to sign in.
- **Resource group:** Create a resource group or select an existing resource group.
- **Location:** Select the region you registered your gateway in.
- **Installation Name:** If your gateway installation isn't already selected, select the gateway registered.

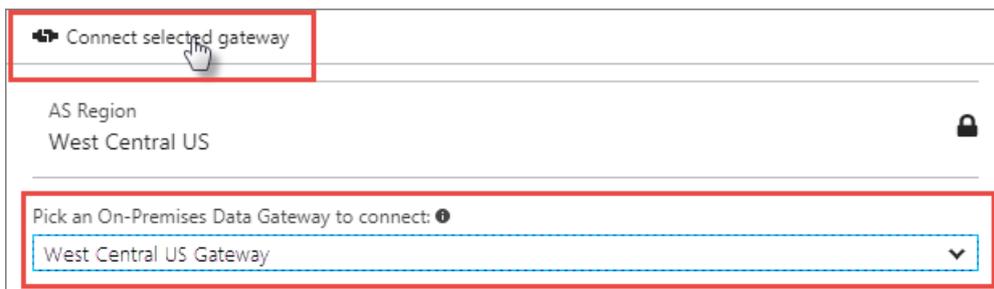
When you're done, click **Create**.

Connect servers to the gateway resource

1. In your Azure Analysis Services server overview, click **On-Premises Data Gateway**.



2. In **Pick an On-Premises Data Gateway to connect**, select your gateway resource, and then click **Connect selected gateway**.



NOTE

If your gateway does not appear in the list, your server is likely not in the same region as the region you specified when registering the gateway.

That's it. If you need to open ports or do any troubleshooting, be sure to check out [On-premises data gateway](#).

Next steps

- [Manage Analysis Services](#)
- [Get data from Azure Analysis Services](#)

Import a Power BI Desktop file

4/13/2018 • 1 min to read • [Edit Online](#)

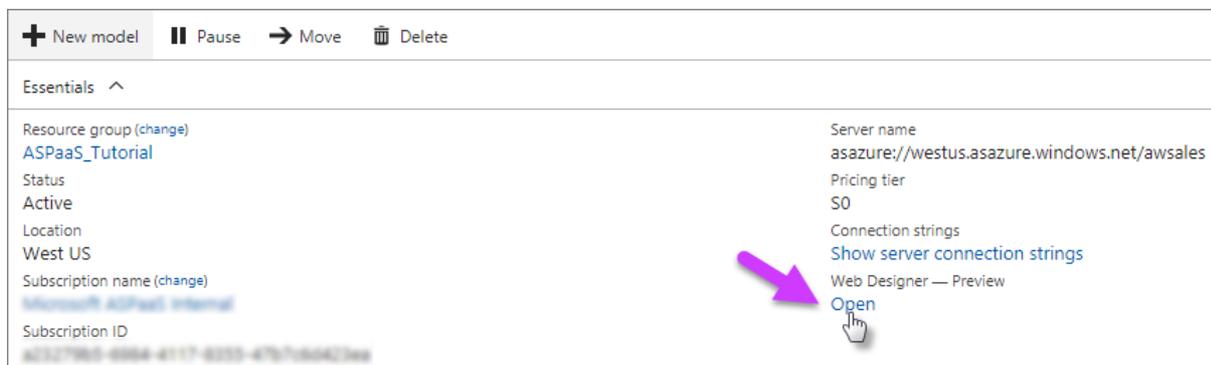
You can create a new model in Azure AS by importing a Power BI Desktop file (pbix) file. Model metadata, cached data, and datasource connections are imported. Reports and visualizations are not imported.

Restrictions

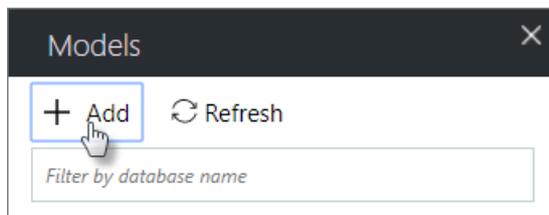
- The pbix model can connect to Azure SQL Database and Azure SQL Data Warehouse data sources only.
- The pbix model cannot have live or DirectQuery connections.
- Import may fail if your pbix data model contains metadata not supported in Analysis Services.

To import from pbix

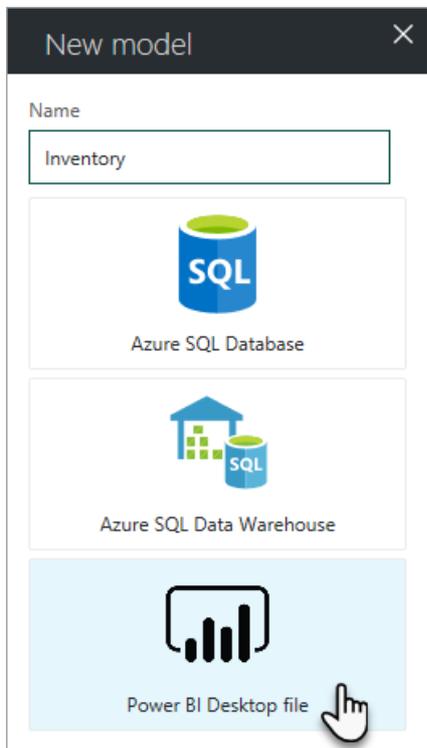
1. In your server's **Overview** > **Web designer**, click **Open**.



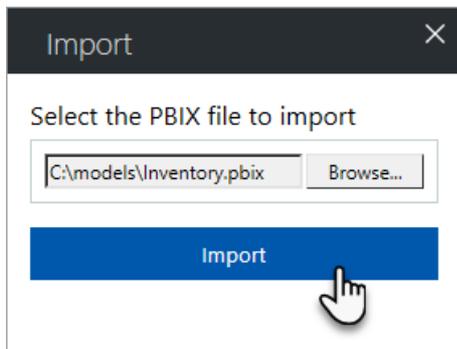
2. In **Web designer** > **Models**, click **+ Add**.



3. In **New model**, type a model name, and then select Power BI Desktop file.



4. In **Import**, locate and select your file.



See also

[Create a model in Azure portal](#)

[Connect to Azure Analysis Services](#)

Backup and restore

4/13/2018 • 2 min to read • [Edit Online](#)

Backing up tabular model databases in Azure Analysis Services is much the same as for on-premises Analysis Services. The primary difference is where you store your backup files. Backup files must be saved to a container in an [Azure storage account](#). You can use a storage account and container you already have, or they can be created when configuring storage settings for your server.

NOTE

Creating a storage account can result in a new billable service. To learn more, see [Azure Storage Pricing](#).

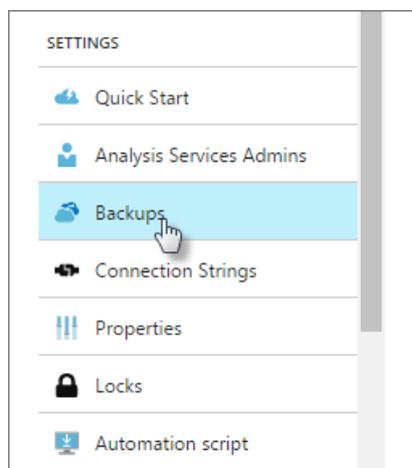
Backups are saved with an abf extension. For in-memory tabular models, both model data and metadata are stored. For DirectQuery tabular models, only model metadata is stored. Backups can be compressed and encrypted, depending on the options you choose.

Configure storage settings

Before backing up, you need to configure storage settings for your server.

To configure storage settings

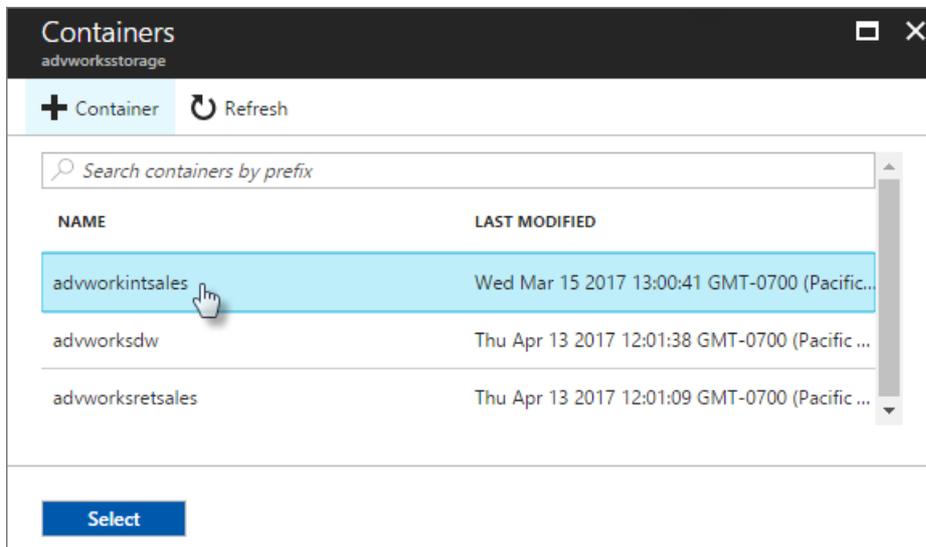
1. In Azure portal > **Settings**, click **Backup**.



2. Click **Enabled**, then click **Storage Settings**.



3. Select your storage account or create a new one.
4. Select a container or create a new one.



5. Save your backup settings.



Backup

To backup by using SSMS

1. In SSMS, right-click a database > **Back Up**.
2. In **Backup Database** > **Backup file**, click **Browse**.
3. In the **Save file as** dialog, verify the folder path, and then type a name for the backup file.
4. In the **Backup Database** dialog, select options.

Allow file overwrite - Select this option to overwrite backup files of the same name. If this option is not selected, the file you are saving cannot have the same name as a file that already exists in the same location.

Apply compression - Select this option to compress the backup file. Compressed backup files save disk space, but require slightly higher CPU utilization.

Encrypt backup file - Select this option to encrypt the backup file. This option requires a user-supplied password to secure the backup file. The password prevents reading of the backup data any other means than a restore operation. If you choose to encrypt backups, store the password in a safe location.

5. Click **OK** to create and save the backup file.

PowerShell

Use [Backup-ASDatabase](#) cmdlet.

Restore

When restoring, your backup file must be in the storage account you've configured for your server. If you need to move a backup file from an on-premises location to your storage account, use [Microsoft Azure Storage Explorer](#) or the [AzCopy](#) command-line utility.

NOTE

If you're restoring from an on-premises server, you must remove all the domain users from the model's roles and add them back to the roles as Azure Active Directory users.

To restore by using SSMS

1. In SSMS, right-click a database > **Restore**.
2. In the **Backup Database** dialog, in **Backup file**, click **Browse**.
3. In the **Locate Database Files** dialog, select the file you want to restore.
4. In **Restore database**, select the database.
5. Specify options. Security options must match the backup options you used when backing up.

PowerShell

Use [Restore-ASDatabase](#) cmdlet.

Related information

[Azure storage accounts](#)

[High availability](#)

[Manage Azure Analysis Services](#)

Asynchronous refresh with the REST API

4/24/2018 • 6 min to read • [Edit Online](#)

By using any programming language that supports REST calls, you can perform asynchronous data-refresh operations on your Azure Analysis Services tabular models. This includes synchronization of read-only replicas for query scale-out.

Data-refresh operations can take some time depending on a number of factors including data volume, level of optimization using partitions, etc. These operations have traditionally been invoked with existing methods such as using [TOM](#) (Tabular Object Model), [PowerShell](#) cmdlets, or [TMSL](#) (Tabular Model Scripting Language). However, these methods can require often unreliable, long-running HTTP connections.

The REST API for Azure Analysis Services enables data-refresh operations to be carried out asynchronously. By using the REST API, long-running HTTP connections from client applications aren't necessary. There are also other built-in features for reliability, such as auto retries and batched commits.

Base URL

The base URL follows this format:

```
https://<rollout>.asazure.windows.net/servers/<serverName>/models/<resource>/
```

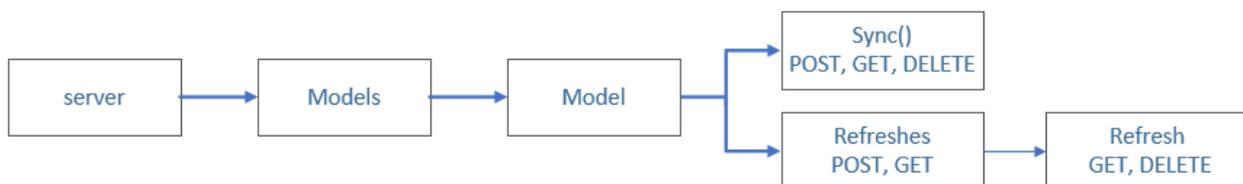
For example, consider a model named AdventureWorks on a server named myserver, located in the West US Azure region. The server name is:

```
asazure://westus.asazure.windows.net/myserver
```

The base URL for this server name is:

```
https://westus.asazure.windows.net/servers/myserver/models/AdventureWorks/
```

By using the base URL, resources and operations can be appended based on the following parameters:



- Anything that ends in **s** is a collection.
- Anything that ends with **()** is a function.
- Anything else is a resource/object.

For example, you can use the POST verb on the Refreshes collection to perform a refresh operation:

```
https://westus.asazure.windows.net/servers/myserver/models/AdventureWorks/refreshes
```

Authentication

All calls must be authenticated with a valid Azure Active Directory (OAuth 2) token in the Authorization header and must meet the following requirements:

- The token must be either a user token or an application service principal.
- The token must have the correct audience set to `https://*.asazure.windows.net`.
- The user or application must have sufficient permissions on the server or model to make the requested call. The permission level is determined by roles within the model or the admin group on the server.

IMPORTANT

Currently, **server admin** role permissions are necessary.

POST /refreshes

To perform a refresh operation, use the POST verb on the /refreshes collection to add a new refresh item to the collection. The Location header in the response includes the refresh ID. The client application can disconnect and check the status later if required because it is asynchronous.

Only one refresh operation is accepted at a time for a model. If there's a current running refresh operation and another is submitted, the 409 Conflict HTTP status code is returned.

The body may resemble the following:

```
{
  "Type": "Full",
  "CommitMode": "transactional",
  "MaxParallelism": 2,
  "RetryCount": 2,
  "Objects": [
    {
      "table": "DimCustomer",
      "partition": "DimCustomer"
    },
    {
      "table": "DimDate"
    }
  ]
}
```

Parameters

Specifying parameters is not required. The default is applied.

NAME	TYPE	DESCRIPTION	DEFAULT
Type	Enum	The type of processing to perform. The types are aligned with the TMSL refresh command types: full, clearValues, calculate, dataOnly, automatic, add, and defragment.	automatic

NAME	TYPE	DESCRIPTION	DEFAULT
CommitMode	Enum	Determines if objects will be committed in batches or only when complete. Modes include: default, transactional, partialBatch.	transactional
MaxParallelism	Int	This value determines the maximum number of threads on which to run processing commands in parallel. This value aligned with the MaxParallelism property that can be set in the TMSL Sequence command or using other methods.	10
RetryCount	Int	Indicates the number of times the operation will retry before failing.	0
Objects	Array	An array of objects to be processed. Each object includes: "table" when processing the entire table or "table" and "partition" when processing a partition. If no objects are specified, the whole model is refreshed.	Process the entire model

CommitMode is equal to partialBatch. It's used when doing an initial load of large datasets that could take hours. If the refresh operation fails after successfully committing one or more batches, the successfully committed batches will remain committed (it will not roll back successfully committed batches).

NOTE

At time of writing, the batch size is the MaxParallelism value, but this value could change.

GET /refreshes/<refreshId>

To check the status of a refresh operation, use the GET verb on the refresh ID. Here's an example of the response body. If the operation is in progress, **inProgress** is returned in status.

```

{
  "startTime": "2017-12-07T02:06:57.1838734Z",
  "endTime": "2017-12-07T02:07:00.4929675Z",
  "type": "full",
  "status": "succeeded",
  "currentRefreshType": "full",
  "objects": [
    {
      "table": "DimCustomer",
      "partition": "DimCustomer",
      "status": "succeeded"
    },
    {
      "table": "DimDate",
      "partition": "DimDate",
      "status": "succeeded"
    }
  ]
}

```

GET /refreshes

To get a list of historical refresh operations for a model, use the GET verb on the /refreshes collection. Here's an example of the response body.

NOTE

At time of writing, the last 30 days of refresh operations are stored and returned, but this number could change.

```

[
  {
    "refreshId": "1344a272-7893-4afa-a4b3-3fb87222fdac",
    "startTime": "2017-12-09T01:58:04.76",
    "endTime": "2017-12-09T01:58:12.607",
    "status": "succeeded"
  },
  {
    "refreshId": "474fc5a0-3d69-4c5d-adb4-8a846fa5580b",
    "startTime": "2017-12-07T02:05:48.32",
    "endTime": "2017-12-07T02:05:54.913",
    "status": "succeeded"
  }
]

```

DELETE /refreshes/<refreshId>

To cancel an in-progress refresh operation, use the DELETE verb on the refresh ID.

POST /sync

Having performed refresh operations, it may be necessary to synchronize the new data with replicas for query scale-out. To perform a synchronize operation for a model, use the POST verb on the /sync function. The Location header in the response includes the sync operation ID.

GET /sync status

To check the status of a sync operation, use the GET verb passing the operation ID as a parameter. Here's an

example of the response body:

```
{
  "operationId": "cd5e16c6-6d4e-4347-86a0-762bdf5b4875",
  "database": "AdventureWorks2",
  "UpdatedAt": "2017-12-09T02:44:26.18",
  "StartedAt": "2017-12-09T02:44:20.743",
  "syncstate": 2,
  "details": null
}
```

Values for `syncstate`:

- 0: Replicating. Database files are being replicated to a target folder.
- 1: Rehydrating. The database is being rehydrated on read-only server instance(s).
- 2: Completed. The sync operation completed successfully.
- 3: Failed. The sync operation failed.
- 4: Finalizing. The sync operation has completed but is performing cleanup steps.

Code sample

Here's a C# code sample to get you started, [RestApiSample on GitHub](#).

To use the code sample

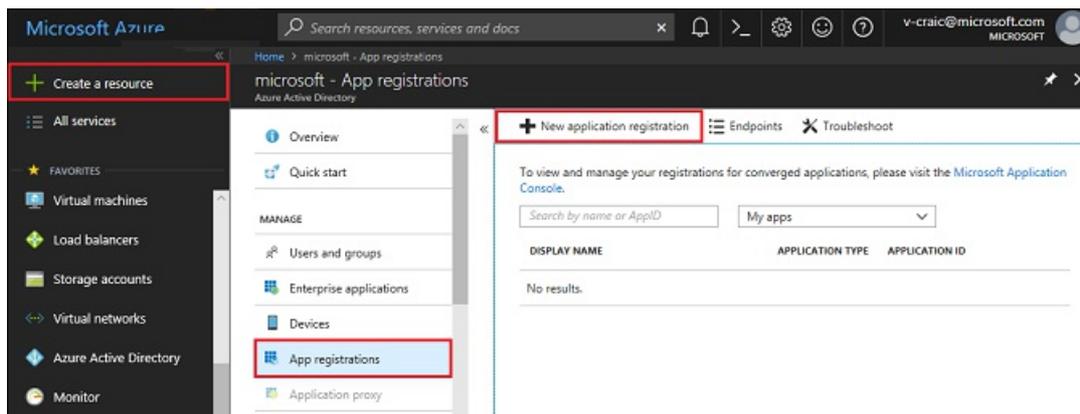
1. Clone or download the repo. Open the RestApiSample solution.
2. Find the line `client.BaseAddress = ...` and provide your [base URL](#).

The code sample can use interactive login, username/password, or [service principal](#).

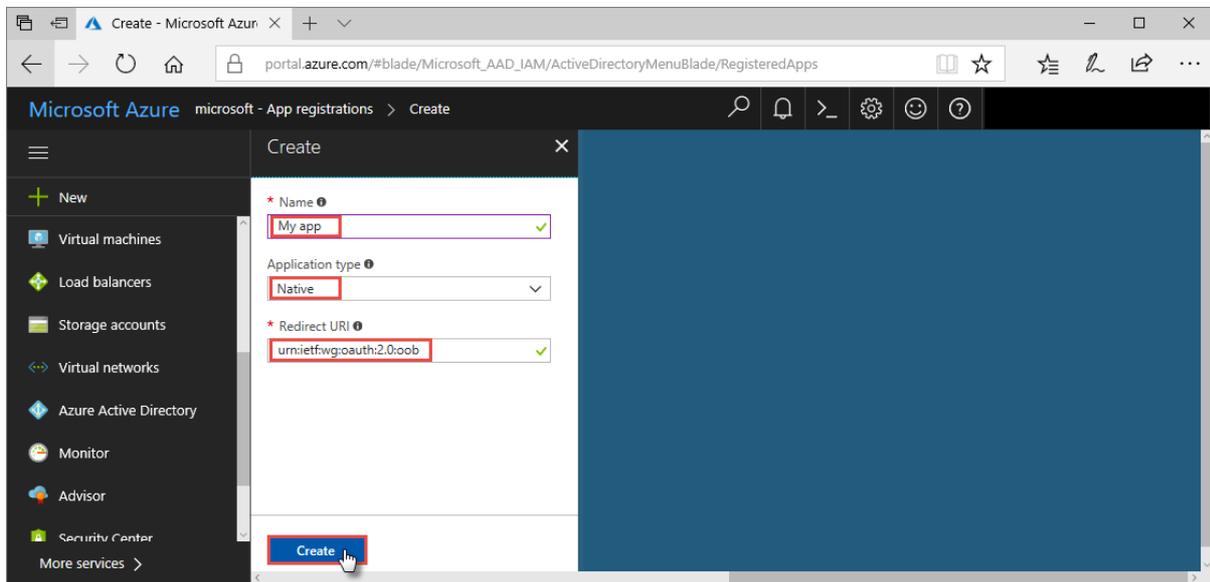
Interactive login or username/password

This form of authentication requires an Azure application be created with the necessary API permissions assigned.

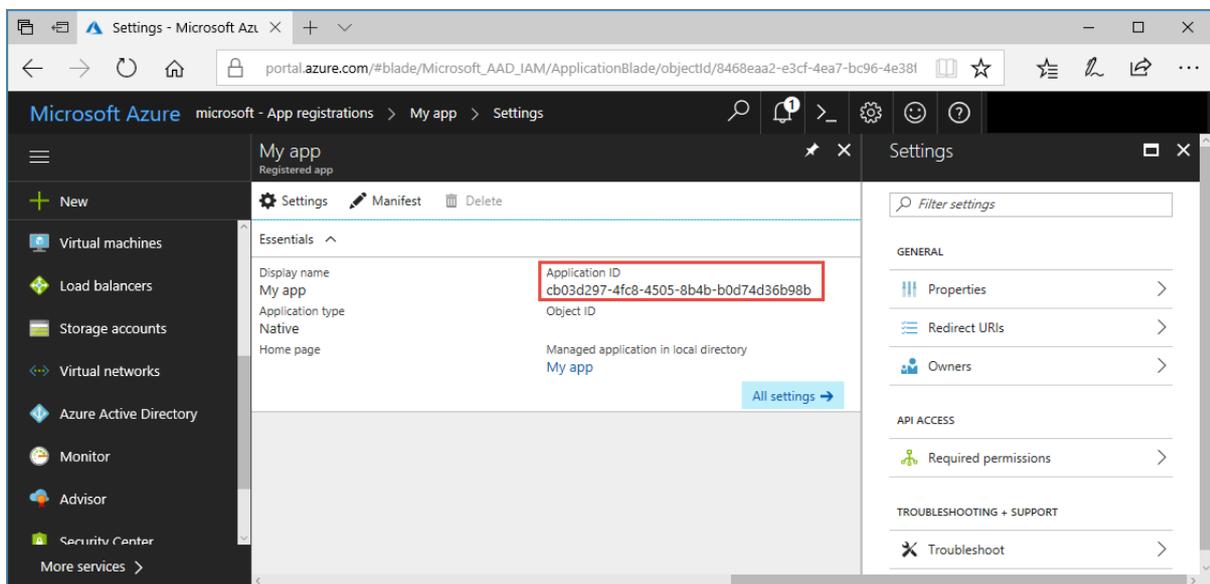
1. In Azure portal, click **Create a resource** > **Azure Active Directory** > **App registrations** > **New application registration**.



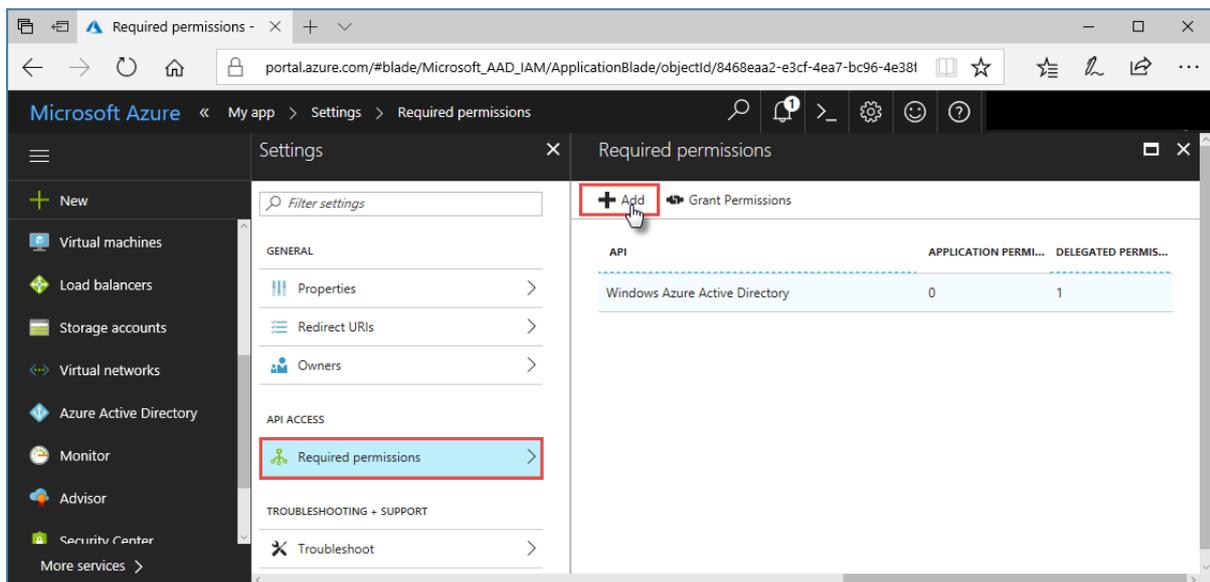
2. In **Create**, type a name, select **Native** application type. For **Redirect URI**, enter `urn:ietf:wg:oauth:2.0:oob`, and then click **Create**.



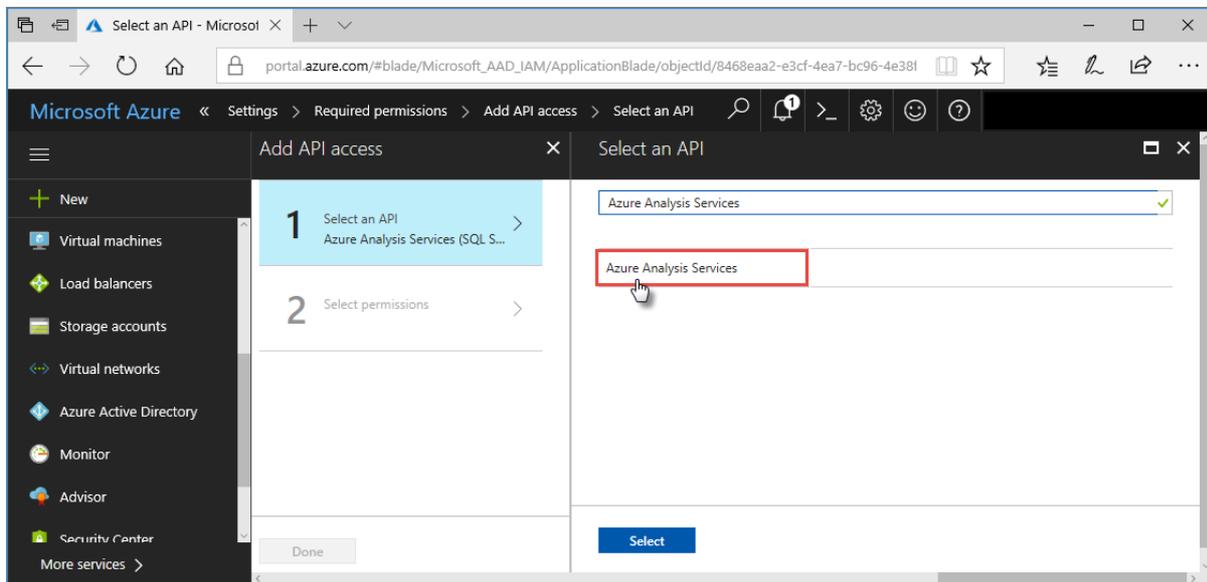
3. Select your app and then copy and save the **Application ID**.



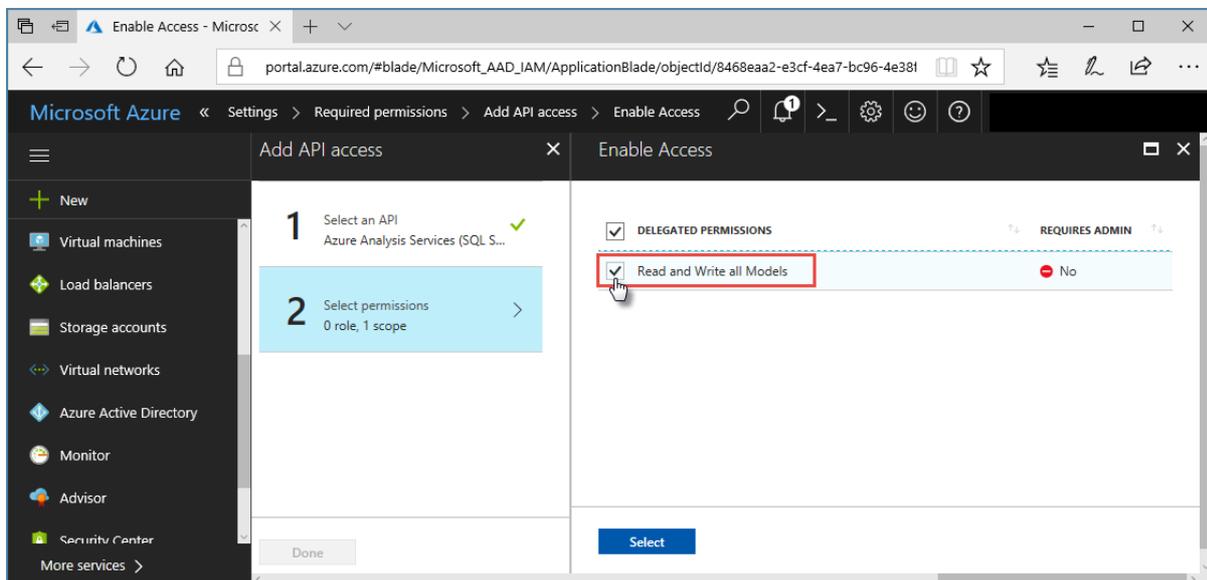
4. In **Settings**, click **Required permissions** > **Add**.



5. In **Select an API**, type **Azure Analysis Services** into the search box, and then select it.



6. Select **Read and Write all Models**, and then click **Select**. When both are selected, click **Done** to add the permissions. It may take a few minutes to propagate.



7. In the code sample, find the **UpdateToken()** method. Observe the contents of this method.
8. Find **string clientID = ...**, and then enter the **Application ID** you copied from step 3.
9. Run the sample.

Service principal

See [Create service principal - Azure portal](#) and [Add a service principal to the server administrator role](#) for more info on how to set up a service principal and assign the necessary permissions in Azure AS. Once you've completed the steps, complete the following additional steps:

1. In the code sample, find **string authority = ...**, replace **common** with your organization's tenant ID.
2. Comment/uncomment so the ClientCredential class is used to instantiate the cred object. Ensure the <App ID> and <App Key> values are accessed in a secure way or use certificate-based authentication for service principals.
3. Run the sample.

See also

[Samples](#)
[REST API](#)

Azure Analysis Services scale-out

4/17/2018 • 3 min to read • [Edit Online](#)

With scale-out, client queries can be distributed among multiple *query replicas* in a query pool, reducing response times during high query workloads. You can also separate processing from the query pool, ensuring client queries are not adversely affected by processing operations. Scale-out can be configured in Azure portal or by using the Analysis Services REST API.

How it works

In a typical server deployment, one server serves as both processing server and query server. If the number of client queries against models on your server exceeds the Query Processing Units (QPU) for your server's plan, or model processing occurs at the same time as high query workloads, performance can decrease.

With scale-out, you can create a query pool with up to seven additional query replicas (eight total, including your server). You can scale the number of query replicas to meet QPU demands at critical times and you can separate a processing server from the query pool at any time. All query replicas are created in the same region as your server.

Regardless of the number of query replicas you have in a query pool, processing workloads are not distributed among query replicas. A single server serves as the processing server. Query replicas serve only queries against the models synchronized between each replica in the query pool.

When processing operations are completed, a synchronization must be performed between the processing server and the query replica servers. When automating processing operations, it's important to configure a synchronization operation upon successful completion of processing operations. Synchronization can be performed manually in the portal, or by using PowerShell or REST API.

NOTE

Scale-out is available for servers in the Standard pricing tier. Each query replica is billed at the same rate as your server.

NOTE

Scale-out does not increase the amount of available memory for your server. To increase memory, you need to upgrade your plan.

Monitor QPU usage

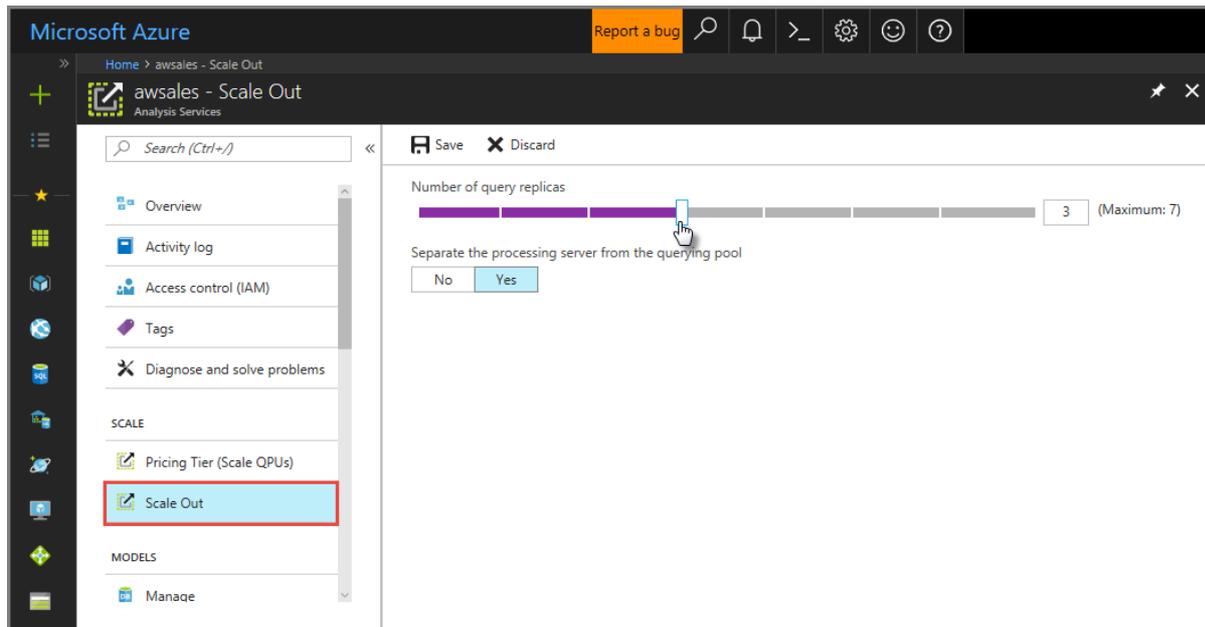
To determine if scale-out for your server is necessary, monitor your server in Azure portal by using Metrics. If your QPU regularly maxes out, it means the number of queries against your models is exceeding the QPU limit for your plan. The Query pool job queue length metric also increases when the number of queries in the query thread pool queue exceeds available QPU. To learn more, see [Monitor server metrics](#).

Configure scale-out

In Azure portal

1. In the portal, click **Scale-out**. Use the slider to select the number of query replica servers. The number of replicas you choose is in addition to your existing server.

2. In **Separate the processing server from the querying pool**, select yes to exclude your processing server from query servers.



3. Click **Save** to provision your new query replica servers.

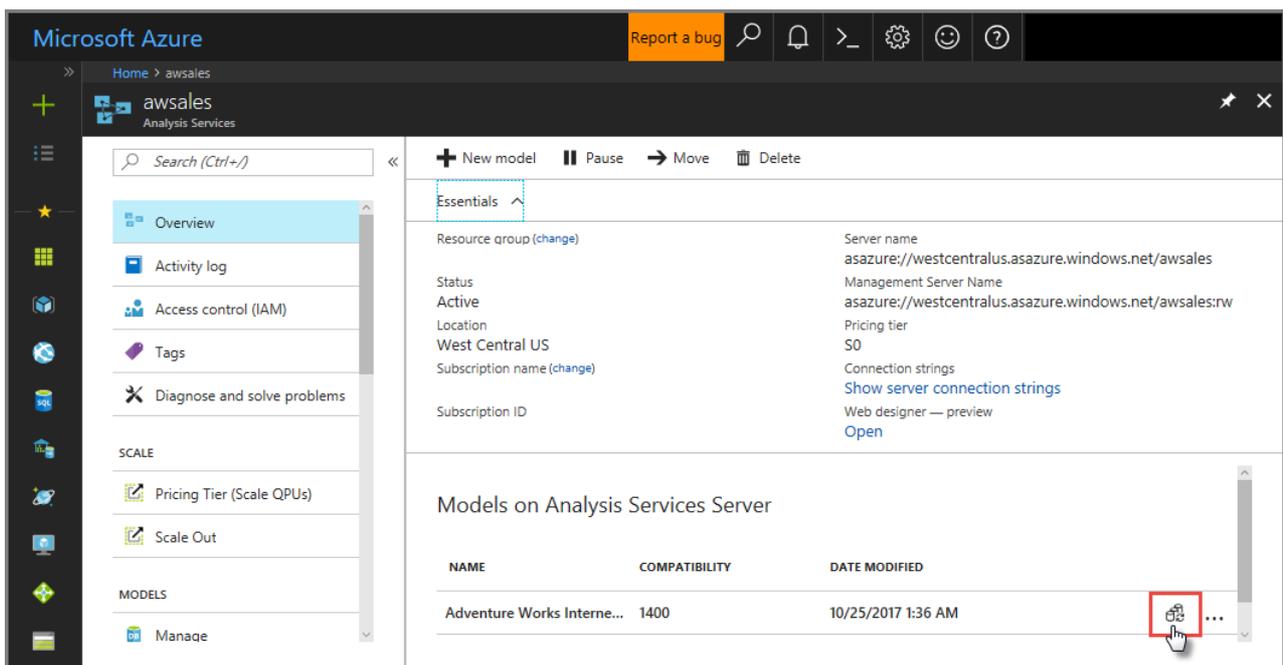
Tabular models on your primary server are synchronized with the replica servers. When synchronization is complete, the query pool begins distributing incoming queries among the replica servers.

Synchronization

When you provision new query replicas, Azure Analysis Services automatically replicates your models across all replicas. You can also perform a manual synchronization by using the portal or REST API. When you process your models, you should perform a synchronization so updates are synchronized among your query replicas.

In Azure portal

In **Overview** > model > **Synchronize model**.



REST API

Use the **sync** operation.

Synchronize a model

```
POST https://<region>.asazure.windows.net/servers/<servername>:rw/models/<modelname>/sync
```

Get sync status

```
GET https://<region>.asazure.windows.net/servers/<servername>:rw/models/<modelname>/sync
```

PowerShell

Before using PowerShell, [install or update the latest AzureRM module](#).

To set the number of query replicas, use [Set-AzureRmAnalysisServicesServer](#). Specify the optional

```
-ReadOnlyReplicaCount
```

 parameter.

To run sync, use [Sync-AzureAnalysisServicesInstance](#).

Connections

On your server's Overview page, there are two server names. If you haven't yet configured scale-out for a server, both server names work the same. Once you configure scale-out for a server, you need to specify the appropriate server name depending on the connection type.

For end-user client connections like Power BI Desktop, Excel, and custom apps, use **Server name**.

For SSMS, SSDT, and connection strings in PowerShell, Azure Function apps, and AMO, use **Management server name**. The management server name includes a special `:rw` (read-write) qualifier. All processing operations occur on the management server.

The screenshot shows the Azure Analysis Services Overview page. At the top, there are action buttons: '+ New model', '|| Pause', '→ Move', and '🗑️ Delete'. Below these is a section titled 'Essentials' with a dropdown arrow. The main content is divided into two columns. The left column lists properties: 'Resource group (change)', 'Status: Active', 'Location: West Central US', 'Subscription name (change)', and 'Subscription ID'. The right column lists server details: 'Server name: asazure://westcentralus.asazure.windows.net/awsales', 'Management Server Name: asazure://westcentralus.asazure.windows.net/awsales:rw' (highlighted with a red box), 'Pricing tier: S0', 'Connection strings: Show server connection strings', and 'Web designer — preview: Open' (highlighted with a dashed blue box).

Related information

[Monitor server metrics](#)

[Manage Azure Analysis Services](#)

Use portal to create an Azure Active Directory application and service principal that can access resources

4/12/2018 • 5 min to read • [Edit Online](#)

When you have code that needs to access or modify resources, you must set up an Azure Active Directory (AD) application. You assign the required permissions to AD application. This approach is preferable to running the app under your own credentials because you can assign permissions to the app identity that are different than your own permissions. Typically, these permissions are restricted to exactly what the app needs to do.

This article shows you how to perform those steps through the portal. It focuses on a single-tenant application where the application is intended to run within only one organization. You typically use single-tenant applications for line-of-business applications that run within your organization.

IMPORTANT

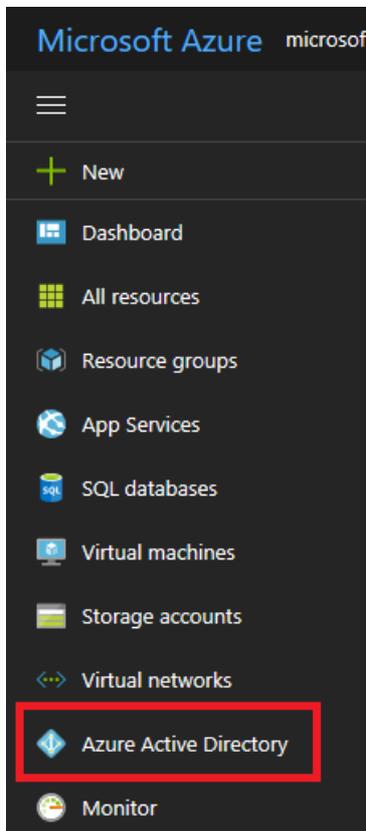
Instead of creating a service principal, consider using Azure AD Managed Service Identity for your application identity. Azure AD MSI is a public preview feature of Azure Active Directory that simplifies creating an identity for code. If your code runs on a service that supports Azure AD MSI and accesses resources that support Azure Active Directory authentication, Azure AD MSI is a better option for you. To learn more about Azure AD MSI, including which services currently support it, see [Managed Service Identity for Azure resources](#).

Required permissions

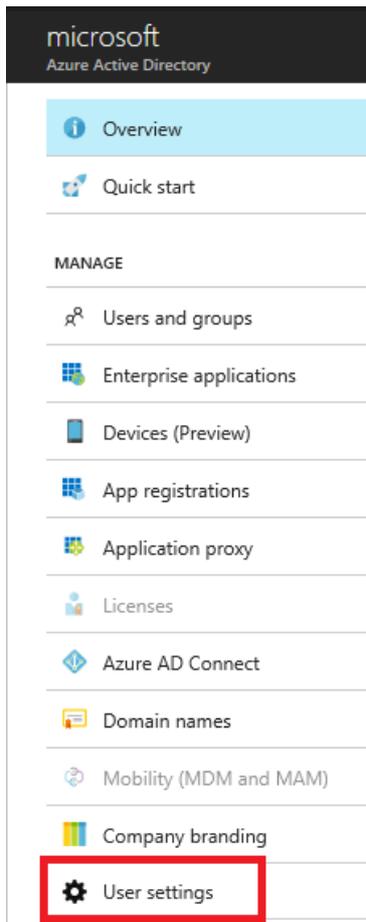
To complete this article, you must have sufficient permissions to register an application with your Azure AD tenant, and assign the application to a role in your Azure subscription. Let's make sure you have the right permissions to perform those steps.

Check Azure Active Directory permissions

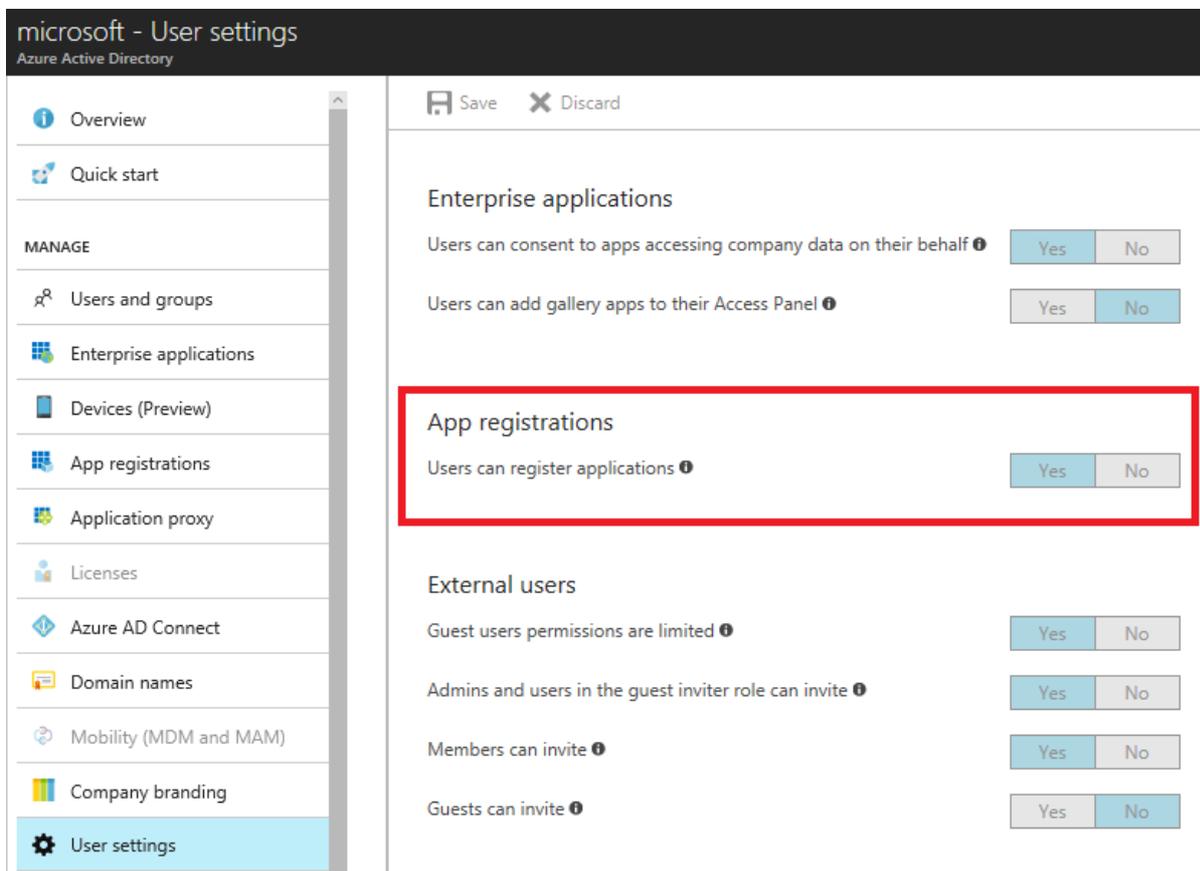
1. Select **Azure Active Directory**.



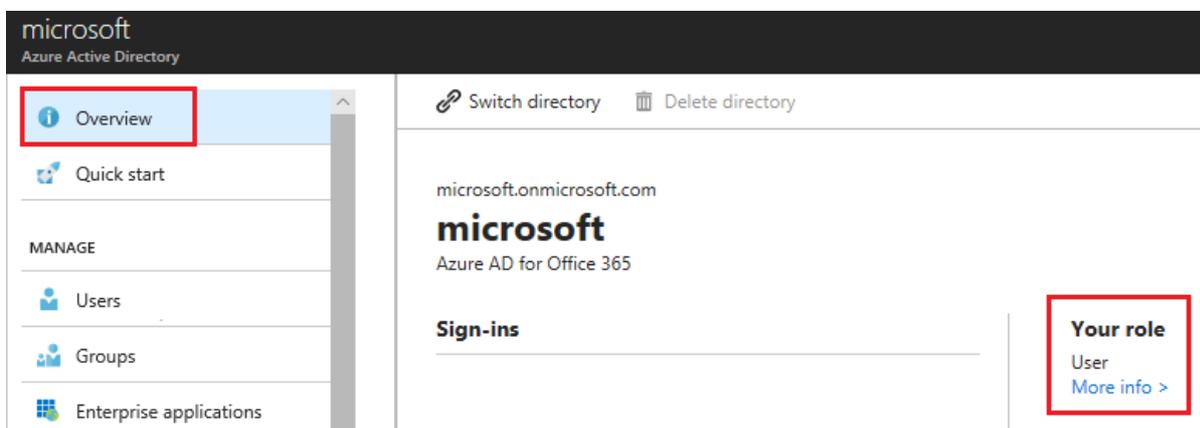
2. In Azure Active Directory, select **User settings**.



3. Check the **App registrations** setting. If set to **Yes**, non-admin users can register AD apps. This setting means any user in the Azure AD tenant can register an app. You can proceed to [Check Azure subscription permissions](#).



4. If the app registrations setting is set to **No**, only [global administrators](#) can register apps. Check whether your account is an admin for the Azure AD tenant. Select **Overview** and look at your user information. If your account is assigned to the User role, but the app registration setting (from the preceding step) is limited to admin users, ask your administrator to either assign you to the global administrator role, or to enable users to register apps.

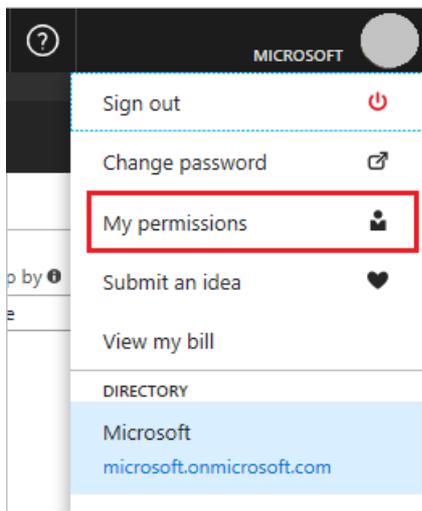


Check Azure subscription permissions

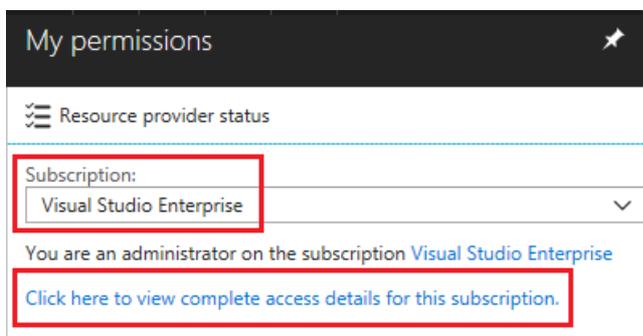
In your Azure subscription, your account must have `Microsoft.Authorization/*/Write` access to assign an AD app to a role. This action is granted through the [Owner](#) role or [User Access Administrator](#) role. If your account is assigned to the **Contributor** role, you do not have adequate permission. You receive an error when attempting to assign the service principal to a role.

To check your subscription permissions:

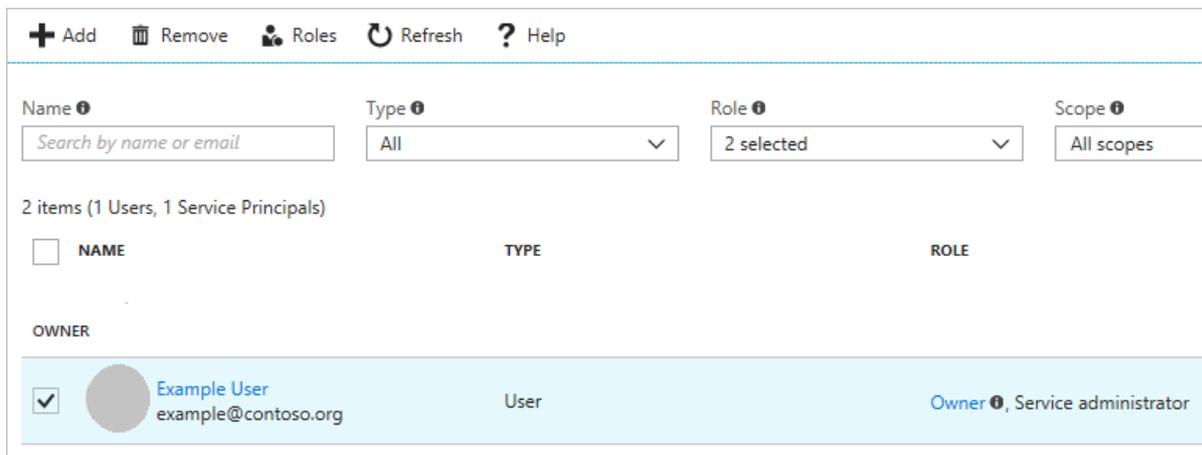
1. Select your account in the upper right corner, and select **My permissions**.



- From the drop-down list, select the subscription. Select **Click here to view complete access details for this subscription.**

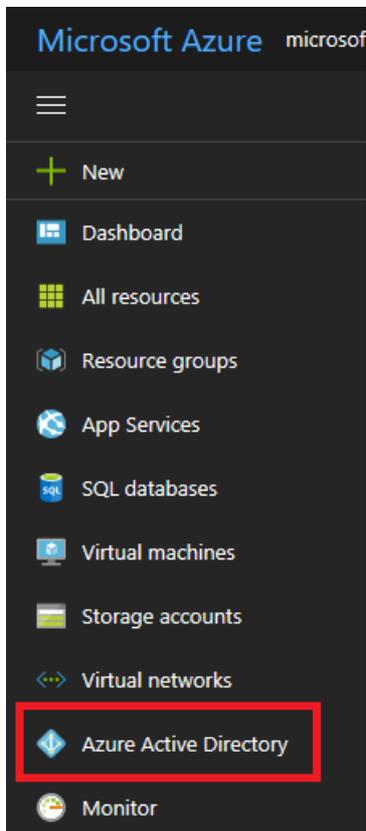


- View your assigned roles, and determine if you have adequate permissions to assign an AD app to a role. If not, ask your subscription administrator to add you to User Access Administrator role. In the following image, the user is assigned to the Owner role, which means that user has adequate permissions.

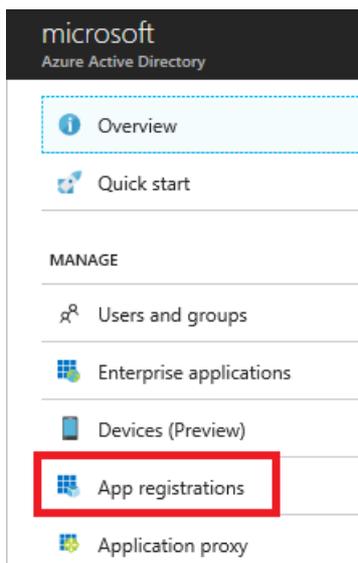


Create an Azure Active Directory application

- Log in to your Azure Account through the [Azure portal](#).
- Select **Azure Active Directory**.



3. Select **App registrations**.



4. Select **New application registration**.



5. Provide a name and URL for the application. Select **Web app / API** for the type of application you want to create. You cannot create credentials for a [Native application](#); therefore, that type does not work for an automated application. After setting the values, select **Create**.

* Name ⓘ
example-app ✓

Application type ⓘ
Web app / API ▾

* Sign-on URL ⓘ
https://contoso.org/exampleapp ✓

Create

You have created your application.

Get application ID and authentication key

When programmatically logging in, you need the ID for your application and an authentication key. To get those values, use the following steps:

1. From **App registrations** in Azure Active Directory, select your application.

+ New application registration Endpoints Troubleshoot

To view and manage your registrations for converged applications, please visit the [Microsoft Application Console](#).

example-app ✓ My apps ▾

DISPLAY NAME	APPLICATION TYPE	APPLICATION ID
EX example-app	Web app / API	9ba61785-e7b9-4526-89d9-84c...

2. Copy the **Application ID** and store it in your application code. Some [sample applications](#) refer to this value as the client ID.

example-app
Registered app

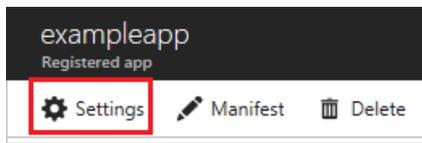
Settings Manifest Delete

Essentials ^

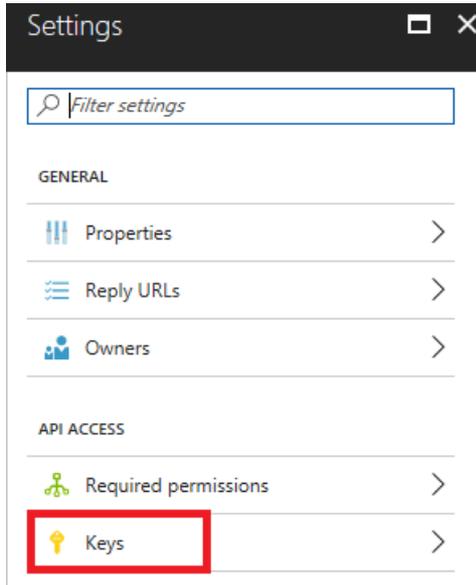
Display name example-app	Application ID 9ba61785-e7b9-4526-89d9-84c41125cf20
Application type Web app / API	Object ID 0cda5df5-d74b-4714-a17b-4439814f2e24
Home page https://contoso.org/exampleapp	Managed application in local directory example-app

All settings →

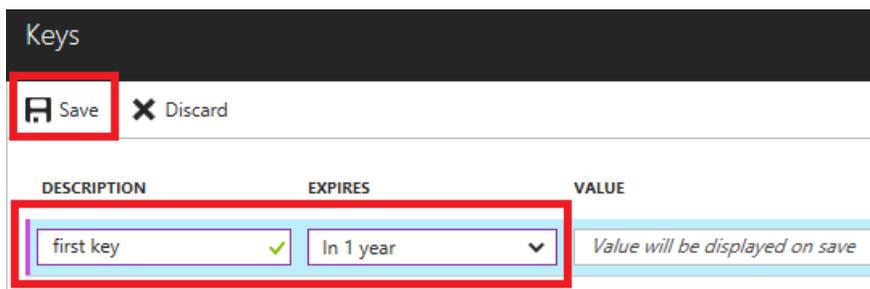
3. To generate an authentication key, select **Settings**.



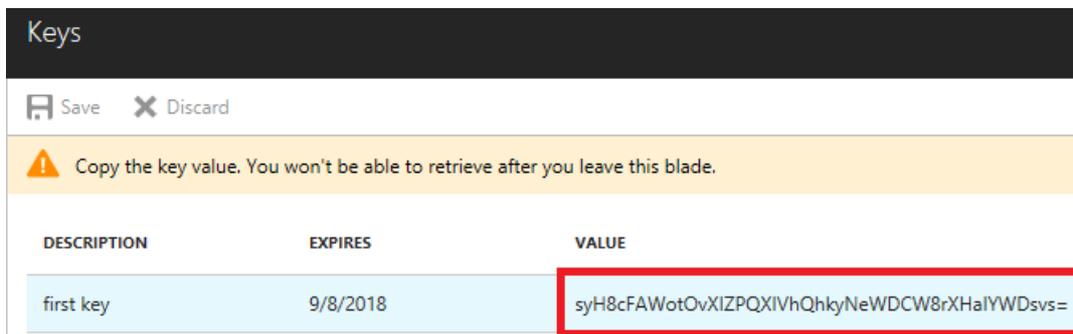
4. To generate an authentication key, select **Keys**.



5. Provide a description of the key, and a duration for the key. When done, select **Save**.



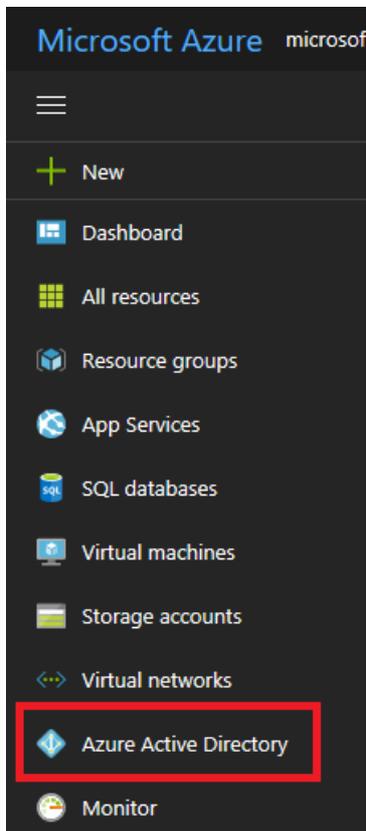
After saving the key, the value of the key is displayed. Copy this value because you are not able to retrieve the key later. You provide the key value with the application ID to log in as the application. Store the key value where your application can retrieve it.



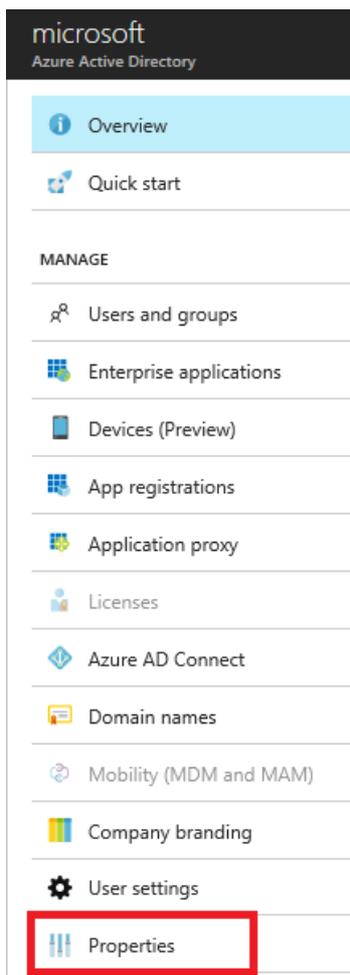
Get tenant ID

When programmatically logging in, you need to pass the tenant ID with your authentication request.

1. Select **Azure Active Directory**.



2. To get the tenant ID, select **Properties** for your Azure AD tenant.



3. Copy the **Directory ID**. This value is your tenant ID.

Save Discard

* Name
Microsoft

Country or region
United States

Location
Asia, United States, Europe datacenters

Notification language
English

Global admin can manage Azure Subscriptions
Yes No

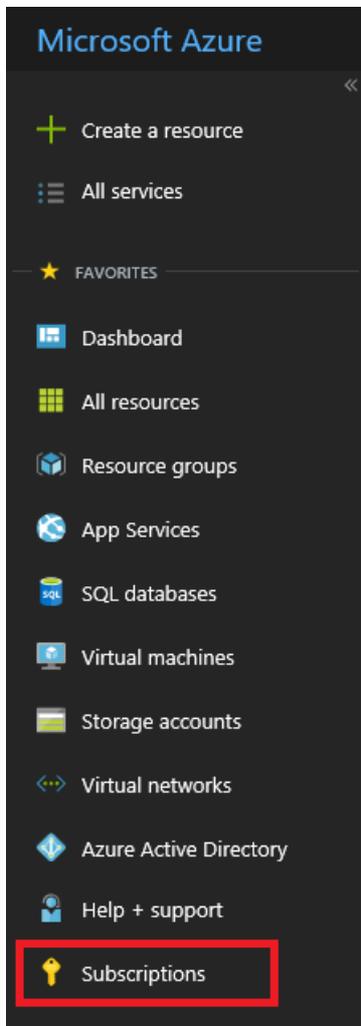
Directory ID
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Assign application to role

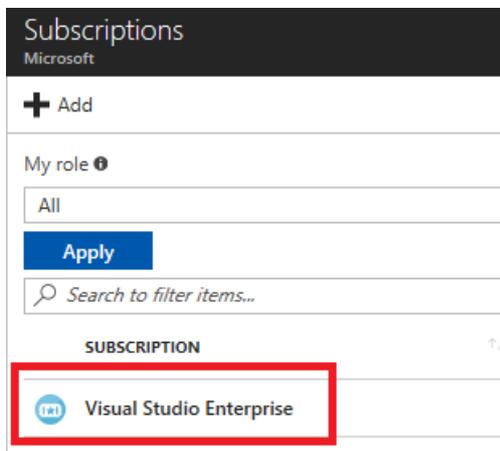
To access resources in your subscription, you must assign the application to a role. Decide which role represents the right permissions for the application. To learn about the available roles, see [RBAC: Built in Roles](#).

You can set the scope at the level of the subscription, resource group, or resource. Permissions are inherited to lower levels of scope. For example, adding an application to the Reader role for a resource group means it can read the resource group and any resources it contains.

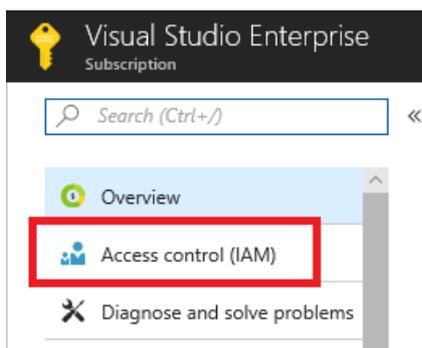
1. Navigate to the level of scope you wish to assign the application to. For example, to assign a role at the subscription scope, select **Subscriptions**. You could instead select a resource group or resource.



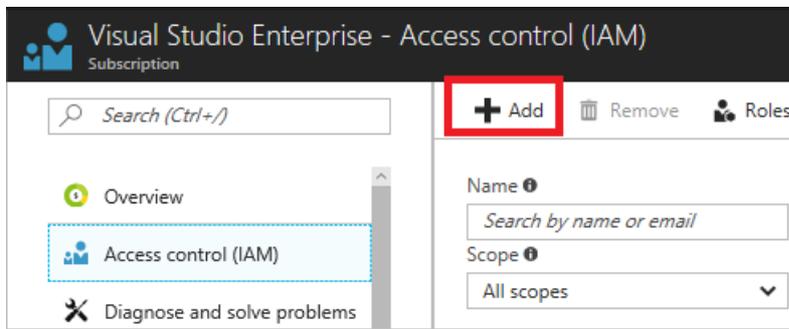
2. Select the particular subscription (resource group or resource) to assign the application to.



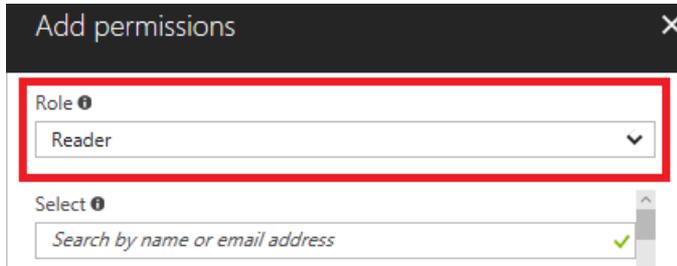
3. Select **Access Control (IAM)**.



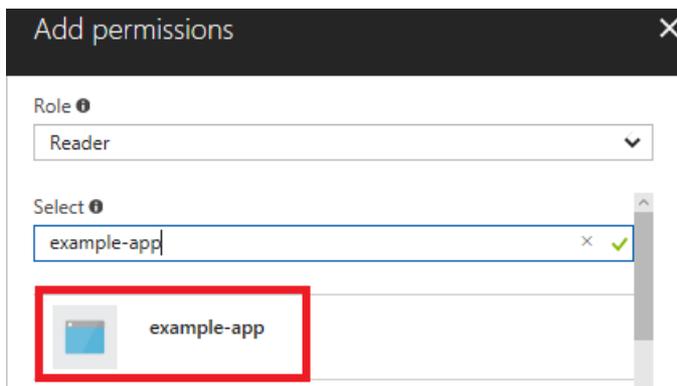
4. Select **Add**.



5. Select the role you wish to assign to the application. The following image shows the **Reader** role.



6. By default, Azure Active Directory applications aren't displayed in the available options. To find your application, you must provide the name of it in the search field. Select it.



7. Select **Save** to finish assigning the role. You see your application in the list of users assigned to a role for that scope.

Next steps

- To set up a multi-tenant application, see [Developer's guide to authorization with the Azure Resource Manager API](#).
- To learn about specifying security policies, see [Azure Role-based Access Control](#).
- For a list of available actions that can be granted or denied to users, see [Azure Resource Manager Resource Provider operations](#).

Use Azure PowerShell to create a service principal with a certificate

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

When you have an app or script that needs to access resources, you can set up an identity for the app and authenticate the app with its own credentials. This identity is known as a service principal. This approach enables you to:

- Assign permissions to the app identity that are different than your own permissions. Typically, these permissions are restricted to exactly what the app needs to do.
- Use a certificate for authentication when executing an unattended script.

IMPORTANT

Instead of creating a service principal, consider using Azure AD Managed Service Identity for your application identity. Azure AD MSI is a public preview feature of Azure Active Directory that simplifies creating an identity for code. If your code runs on a service that supports Azure AD MSI and accesses resources that support Azure Active Directory authentication, Azure AD MSI is a better option for you. To learn more about Azure AD MSI, including which services currently support it, see [Managed Service Identity for Azure resources](#).

This article shows you how to create a service principal that authenticates with a certificate. To set up a service principal with password, see [Create an Azure service principal with Azure PowerShell](#).

You must have the [latest version](#) of PowerShell for this article.

Required permissions

To complete this article, you must have sufficient permissions in both your Azure Active Directory and Azure subscription. Specifically, you must be able to create an app in the Azure Active Directory, and assign the service principal to a role.

The easiest way to check whether your account has adequate permissions is through the portal. See [Check required permission](#).

Create service principal with self-signed certificate

The following example covers a simple scenario. It uses [New-AzureRmADServicePrincipal](#) to create a service principal with a self-signed certificate, and uses [New-AzureRmRoleAssignment](#) to assign the [Contributor](#) role to the service principal. The role assignment is scoped to your currently selected Azure subscription. To select a different subscription, use [Set-AzureRmContext](#).

```

$cert = New-SelfSignedCertificate -CertStoreLocation "cert:\CurrentUser\My" `
  -Subject "CN=exampleappScriptCert" `
  -KeySpec KeyExchange
$keyValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

$sp = New-AzureRMAADServicePrincipal -DisplayName exampleapp `
  -CertValue $keyValue `
  -EndDate $cert.NotAfter `
  -StartDate $cert.NotBefore
Sleep 20
New-AzureRmRoleAssignment -RoleDefinitionName Contributor -ServicePrincipalName $sp.ApplicationId

```

The example sleeps for 20 seconds to allow some time for the new service principal to propagate throughout Azure Active Directory. If your script doesn't wait long enough, you'll see an error stating: "Principal {ID} does not exist in the directory {DIR-ID}." To resolve this error, wait a moment then run the **New-AzureRmRoleAssignment** command again.

You can scope the role assignment to a specific resource group by using the **ResourceGroupName** parameter. You can scope to a specific resource by also using the **ResourceType** and **ResourceName** parameters.

If you **do not have Windows 10 or Windows Server 2016**, you need to download the [Self-signed certificate generator](#) from Microsoft Script Center. Extract its contents and import the cmdlet you need.

```

# Only run if you could not use New-SelfSignedCertificate
Import-Module -Name c:\ExtractedModule\New-SelfSignedCertificateEx.ps1

```

In the script, substitute the following two lines to generate the certificate.

```

New-SelfSignedCertificateEx -StoreLocation CurrentUser `
  -StoreName My `
  -Subject "CN=exampleapp" `
  -KeySpec "Exchange" `
  -FriendlyName "exampleapp"
$cert = Get-ChildItem -path Cert:\CurrentUser\my | where {$PSitem.Subject -eq 'CN=exampleapp' }

```

Provide certificate through automated PowerShell script

Whenever you sign in as a service principal, you need to provide the tenant ID of the directory for your AD app. A tenant is an instance of Azure Active Directory.

```

$TenantId = (Get-AzureRmSubscription -SubscriptionName "Contoso Default").TenantId
$ApplicationId = (Get-AzureRmADApplication -DisplayNameStartWith exampleapp).ApplicationId

$Thumbprint = (Get-ChildItem cert:\CurrentUser\My\ | Where-Object {$_.Subject -match "CN=exampleappScriptCert"
}).Thumbprint
Connect-AzureRmAccount -ServicePrincipal `
  -CertificateThumbprint $Thumbprint `
  -ApplicationId $ApplicationId `
  -TenantId $TenantId

```

Create service principal with certificate from Certificate Authority

The following example uses a certificate issued from a Certificate Authority to create service principal. The assignment is scoped to the specified Azure subscription. It adds the service principal to the [Contributor](#) role. If an error occurs during the role assignment, it retries the assignment.

```

Param (
  [Parameter(Mandatory=$true)]
  [String] $ApplicationDisplayName,

  [Parameter(Mandatory=$true)]
  [String] $SubscriptionId,

  [Parameter(Mandatory=$true)]
  [String] $CertPath,

  [Parameter(Mandatory=$true)]
  [String] $CertPlainPassword
)

Connect-AzureRmAccount
Import-Module AzureRM.Resources
Set-AzureRmContext -Subscription $SubscriptionId

$CertPassword = ConvertTo-SecureString $CertPlainPassword -AsPlainText -Force

$PFXCert = New-Object -TypeName System.Security.Cryptography.X509Certificates.X509Certificate2 -ArgumentList
@($CertPath, $CertPassword)
$KeyValue = [System.Convert]::ToBase64String($PFXCert.GetRawCertData())

$ServicePrincipal = New-AzureRMADServicePrincipal -DisplayName $ApplicationDisplayName
New-AzureRMADSpCredential -ObjectId $ServicePrincipal.Id -CertValue $KeyValue -StartDate $PFXCert.NotBefore -
EndDate $PFXCert.NotAfter
Get-AzureRMADServicePrincipal -ObjectId $ServicePrincipal.Id

$NewRole = $null
$Retries = 0;
While ($NewRole -eq $null -and $Retries -le 6)
{
  # Sleep here for a few seconds to allow the service principal application to become active (should only
  take a couple of seconds normally)
  Sleep 15
  New-AzureRMRoleAssignment -RoleDefinitionName Contributor -ServicePrincipalName
$ServicePrincipal.ApplicationId | Write-Verbose -ErrorAction SilentlyContinue
  $NewRole = Get-AzureRMRoleAssignment -ObjectId $ServicePrincipal.Id -ErrorAction SilentlyContinue
  $Retries++;
}

$NewRole

```

Provide certificate through automated PowerShell script

Whenever you sign in as a service principal, you need to provide the tenant ID of the directory for your AD app. A tenant is an instance of Azure Active Directory.

```

Param (

  [Parameter(Mandatory=$true)]
  [String] $CertPath,

  [Parameter(Mandatory=$true)]
  [String] $CertPlainPassword,

  [Parameter(Mandatory=$true)]
  [String] $ApplicationId,

  [Parameter(Mandatory=$true)]
  [String] $TenantId
)

$CertPassword = ConvertTo-SecureString $CertPlainPassword -AsPlainText -Force
$PFXCert = New-Object `
  -TypeName System.Security.Cryptography.X509Certificates.X509Certificate2 `
  -ArgumentList @($CertPath, $CertPassword)
$Thumbprint = $PFXCert.Thumbprint

Connect-AzureRmAccount -ServicePrincipal `
  -CertificateThumbprint $Thumbprint `
  -ApplicationId $ApplicationId `
  -TenantId $TenantId

```

The application ID and tenant ID aren't sensitive, so you can embed them directly in your script. If you need to retrieve the tenant ID, use:

```
(Get-AzureRmSubscription -SubscriptionName "Contoso Default").TenantId
```

If you need to retrieve the application ID, use:

```
(Get-AzureRmADApplication -DisplayNameStartWith {display-name}).ApplicationId
```

Change credentials

To change the credentials for an AD app, either because of a security compromise or a credential expiration, use the [Remove-AzureRmADAppCredential](#) and [New-AzureRmADAppCredential](#) cmdlets.

To remove all the credentials for an application, use:

```
Get-AzureRmADApplication -DisplayName exampleapp | Remove-AzureRmADAppCredential
```

To add a certificate value, create a self-signed certificate as shown in this article. Then, use:

```
Get-AzureRmADApplication -DisplayName exampleapp | New-AzureRmADAppCredential `
  -CertValue $keyValue `
  -EndDate $cert.NotAfter `
  -StartDate $cert.NotBefore
```

Debug

You may get the following errors when creating a service principal:

- **"Authentication_Unauthorized"** or **"No subscription found in the context."** - You see this error when

your account does not have the [required permissions](#) on the Azure Active Directory to register an app. Typically, you see this error when only admin users in your Azure Active Directory can register apps, and your account is not an admin. Ask your administrator to either assign you to an administrator role, or to enable users to register apps.

- Your account "**does not have authorization to perform action 'Microsoft.Authorization/roleAssignments/write' over scope '/subscriptions/{guid}'.**" - You see this error when your account does not have sufficient permissions to assign a role to an identity. Ask your subscription administrator to add you to User Access Administrator role.

Next steps

- To set up a service principal with password, see [Create an Azure service principal with Azure PowerShell](#).
- For detailed steps on integrating an application into Azure for managing resources, see [Developer's guide to authorization with the Azure Resource Manager API](#).
- For a more detailed explanation of applications and service principals, see [Application Objects and Service Principal Objects](#).
- For more information about Azure Active Directory authentication, see [Authentication Scenarios for Azure AD](#).

Add a service principal to the server administrator role

4/24/2018 • 1 min to read • [Edit Online](#)

To automate unattended PowerShell tasks, a service principal must have **server administrator** privileges on the Analysis Services server being managed. This article describes how to add a service principal to the server administrators role on an Azure AS server.

Before you begin

Before completing this task, you must have a service principal registered in Azure Active Directory.

[Create service principal - Azure portal](#)

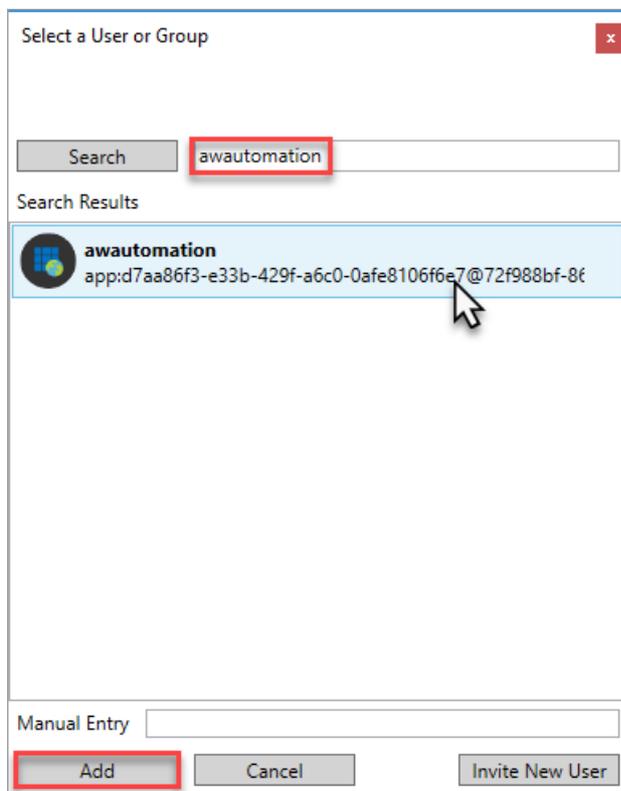
[Create service principal - PowerShell](#)

Required permissions

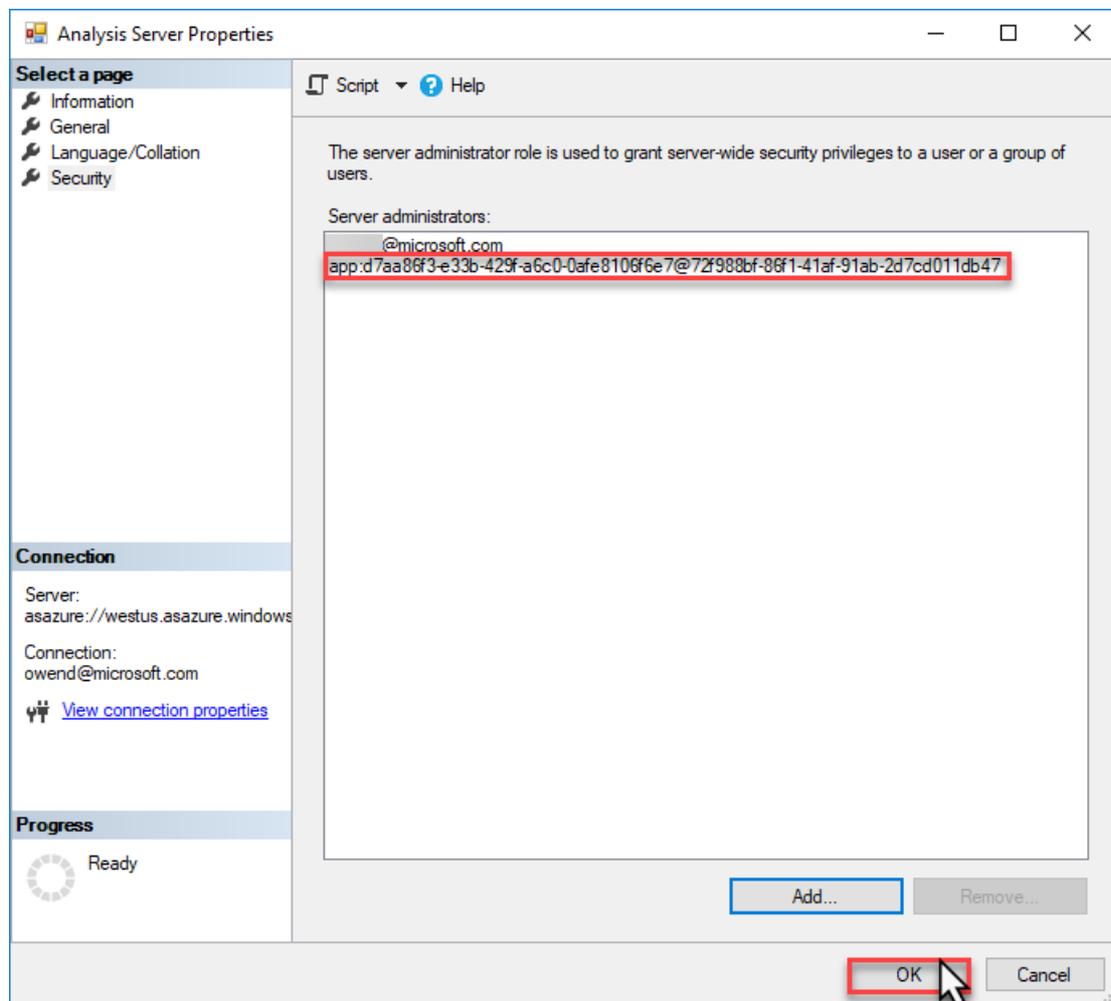
To complete this task, you must have [server administrator](#) permissions on the Azure AS server.

Add service principal to server administrators role

1. In SSMS, connect to your Azure AS server.
2. In **Server Properties** > **Security**, click **Add**.
3. In **Select a User or Group**, search for your registered app by name, select, and then click **Add**.



4. Verify the service principal account ID, and then click **OK**.



NOTE

For server operations using AzureRm cmdlets, service principal running scheduler must also belong to the **Owner** role for the resource in [Azure Role-Based Access Control \(RBAC\)](#).

Related information

- [Download SQL Server PowerShell Module](#)
- [Download SSMS](#)

Alias server names

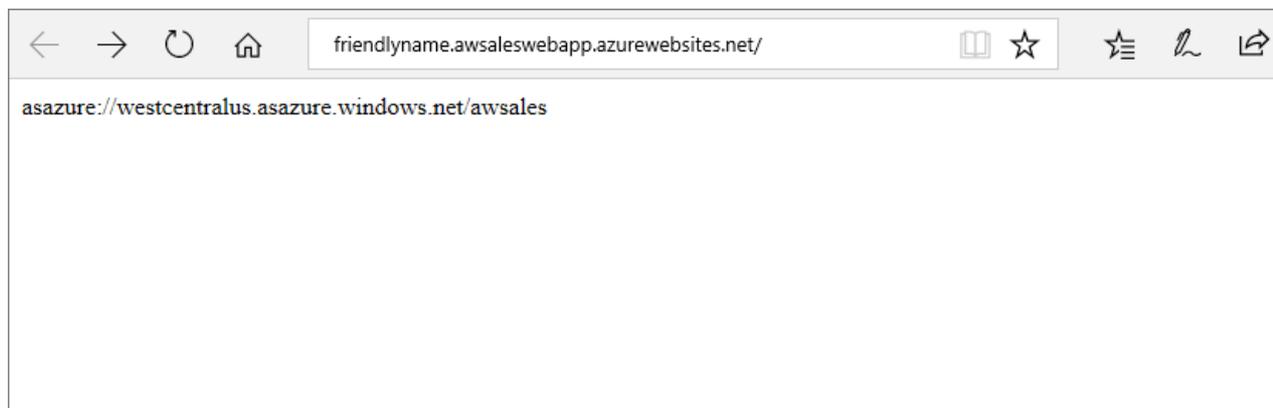
4/18/2018 • 1 min to read • [Edit Online](#)

By using a server name alias, users can connect to your Azure Analysis Services server with a shorter *alias* instead of the server name. When connecting from a client application, the alias is specified as an endpoint using the **link://** protocol format. The endpoint then returns the real server name in order to connect.

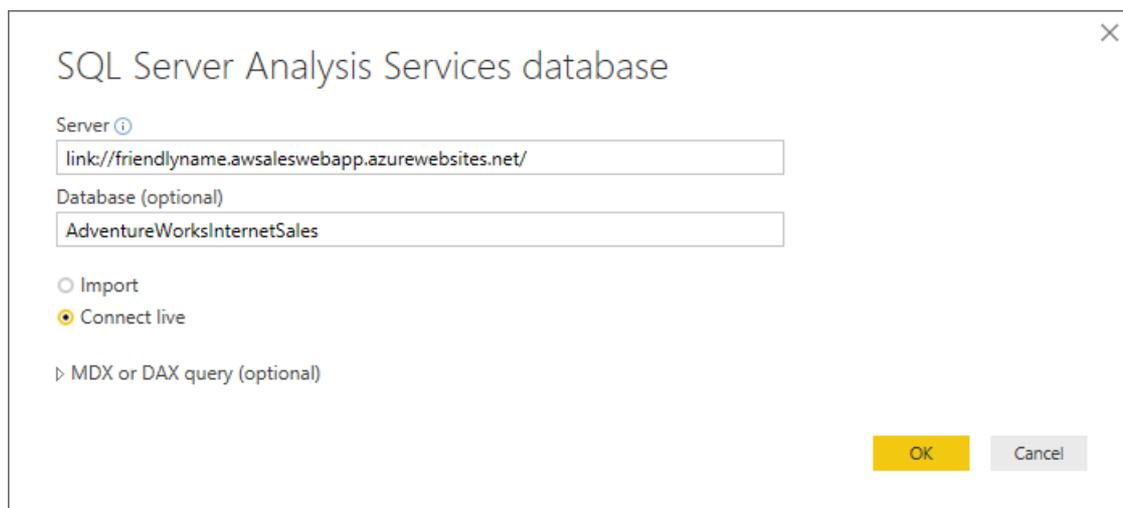
Alias server names are good for:

- Migrating models between servers without affecting users.
- Friendly server names are easier for users to remember.
- Direct users to different servers at different times of the day.
- Direct users in different regions to instances that are geographically closer, like when using Azure Traffic Manager.

Any HTTPS endpoint that returns a valid Azure Analysis Services server name can serve as an alias. The endpoint must support HTTPS over port 443 and the port must not be specified in the URI.



When connecting from a client, the alias server name is entered using **link://** protocol format. For example, in Power BI Desktop:



Create an alias

To create an alias endpoint, you can use any method that returns a valid Azure Analysis Services server name. For example, a reference to a file in Azure Blob Storage containing the real server name, or create and publish an

ASP.NET Web Forms application.

In this example, an ASP.NET Web Forms Application is created in Visual Studio. The master page reference and user control are removed from the Default.aspx page. The contents of Default.aspx are simply the following Page directive:

```
<%@ Page Title="Home Page" Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="FriendlyRedirect._Default" %>
```

The Page_Load event in Default.aspx.cs uses the Response.Write() method to return the Azure Analysis Services server name.

```
protected void Page_Load(object sender, EventArgs e)
{
    this.Response.Write("asazure://<region>.asazure.windows.net/<servername>");
}
```

See also

[Client libraries](#)

[Connect from Power BI Desktop](#)

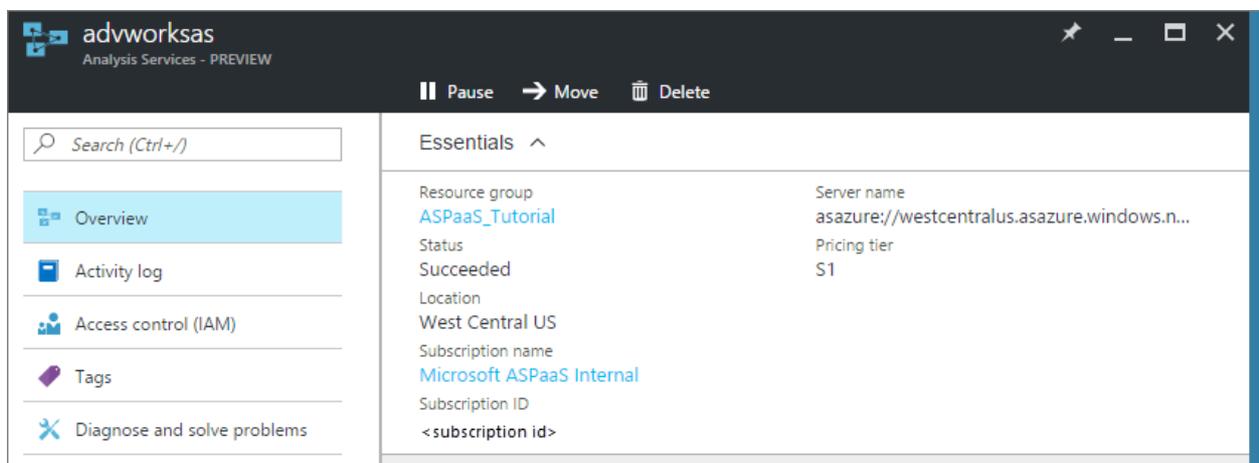
Manage Analysis Services

4/13/2018 • 2 min to read • [Edit Online](#)

Once you've created an Analysis Services server in Azure, there may be some administration and management tasks you need to perform right away or sometime down the road. For example, run processing to the refresh data, control who can access the models on your server, or monitor your server's health. Some management tasks can only be performed in Azure portal, others in SQL Server Management Studio (SSMS), and some tasks can be done in either.

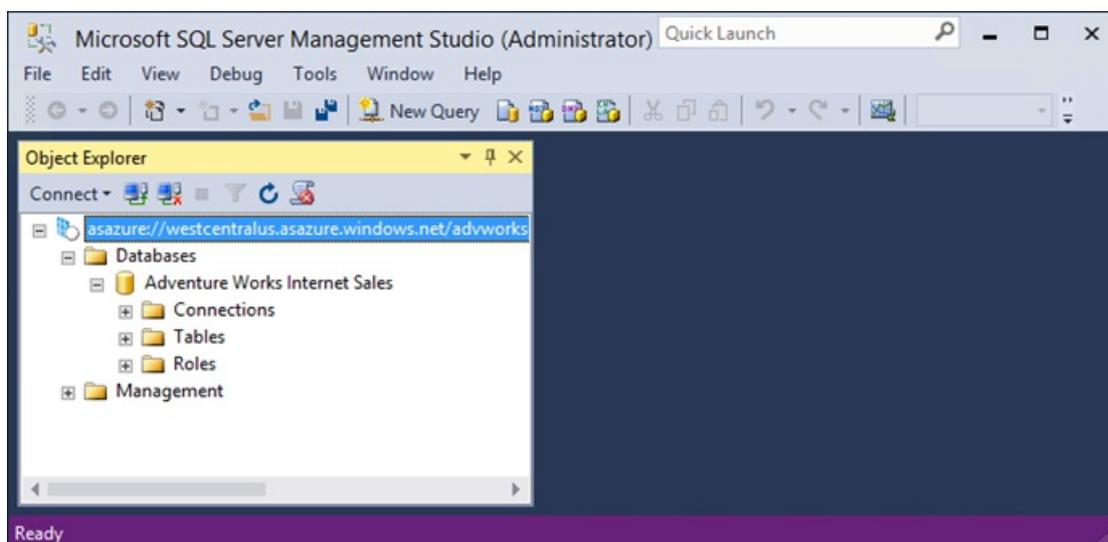
Azure portal

[Azure portal](#) is where you can create and delete servers, monitor server resources, change size, and manage who has access to your servers. If you're having some problems, you can also submit a support request.



SQL Server Management Studio

Connecting to your server in Azure is just like connecting to a server instance in your own organization. From SSMS, you can perform many of the same tasks such as process data or create a processing script, manage roles, and use PowerShell.



Download and install SSMS

To get all the latest features, and the smoothest experience when connecting to your Azure Analysis Services

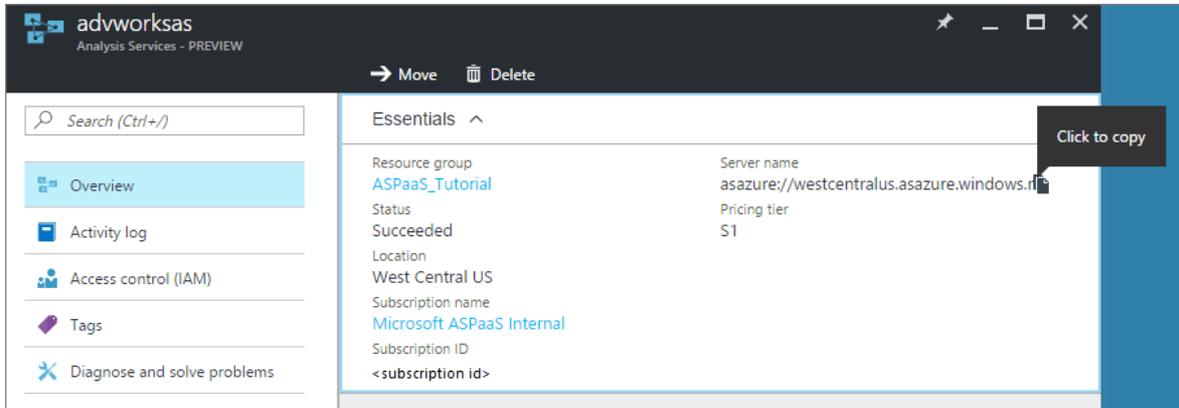
server, be sure you're using the latest version of SSMS.

[Download SQL Server Management Studio.](#)

To connect with SSMS

When using SSMS, before connecting to your server the first time, make sure your username is included in the Analysis Services Admins group. To learn more, see [Server administrators](#) later in this article.

1. Before you connect, you need to get the server name. In **Azure portal** > server > **Overview** > **Server name**, copy the server name.



2. In SSMS > **Object Explorer**, click **Connect** > **Analysis Services**.
3. In the **Connect to Server** dialog box, paste in the server name, then in **Authentication**, choose one of the following authentication types:

NOTE

Authentication type, **Active Directory - Universal with MFA support**, is recommended.

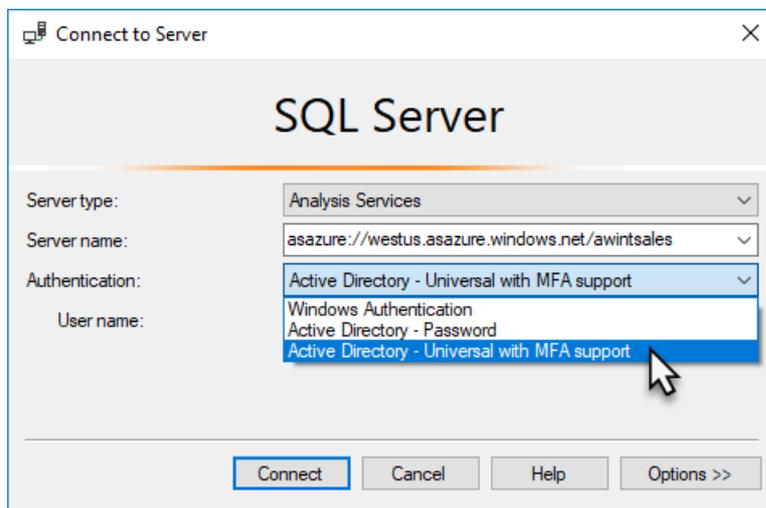
NOTE

If you sign in with a Microsoft Account, Live ID, Yahoo, Gmail, etc., leave the password field blank. You are prompted for a password after clicking Connect.

Windows Authentication to use your Windows domain\username and password credentials.

Active Directory Password Authentication to use an organizational account. For example, when connecting from a non-domain joined computer.

Active Directory - Universal with MFA support to use [non-interactive](#) or [multi-factor authentication](#).



Server administrators and database users

In Azure Analysis Services, there are two types of users, server administrators and database users. Both types of users must be in your Azure Active Directory and must be specified by organizational email address or UPN. To learn more, see [Authentication and user permissions](#).

Troubleshooting connection problems

When connecting using SSMS, if you run into problems, you may need to clear the login cache. Nothing is cached to disc. To clear the cache, close and restart the connect process.

Next steps

If you haven't already deployed a tabular model to your new server, now is a good time. To learn more, see [Deploy to Azure Analysis Services](#).

If you've deployed a model to your server, you're ready to connect to it using a client or browser. To learn more, see [Get data from Azure Analysis Services server](#).

Manage database roles and users

4/13/2018 • 5 min to read • [Edit Online](#)

At the model database level, all users must belong to a role. Roles define users with particular permissions for the model database. Any user or security group added to a role must have an account in an Azure AD tenant in the same subscription as the server.

How you define roles is different depending on the tool you use, but the effect is the same.

Role permissions include:

- **Administrator** - Users have full permissions for the database. Database roles with Administrator permissions are different from server administrators.
- **Process** - Users can connect to and perform process operations on the database, and analyze model database data.
- **Read** - Users can use a client application to connect to and analyze model database data.

When creating a tabular model project, you create roles and add users or groups to those roles by using Role Manager in SSDT. When deployed to a server, you use SSMS, [Analysis Services PowerShell cmdlets](#), or [Tabular Model Scripting Language \(TMSL\)](#) to add or remove roles and user members.

To add or manage roles and users in SSDT

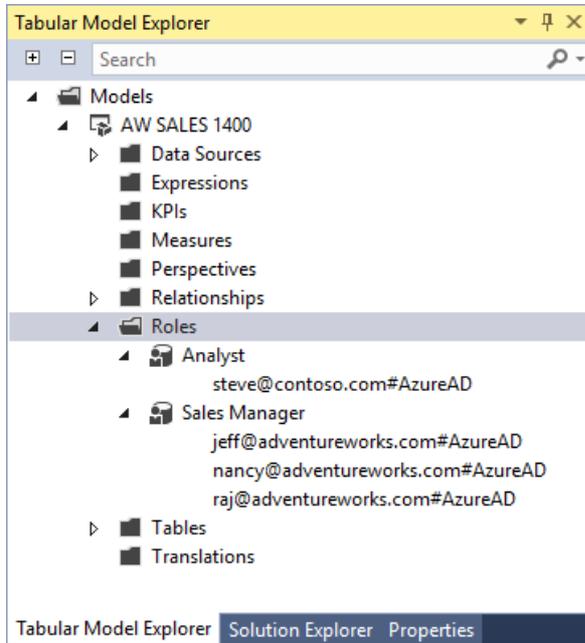
1. In SSDT > **Tabular Model Explorer**, right-click **Roles**.
2. In **Role Manager**, click **New**.
3. Type a name for the role.

By default, the name of the default role is incrementally numbered for each new role. It's recommended you type a name that clearly identifies the member type, for example, Finance Managers or Human Resources Specialists.

4. Select one of the following permissions:

PERMISSION	DESCRIPTION
None	Members cannot modify the model schema and cannot query data.
Read	Members can query data (based on row filters) but cannot modify the model schema.
Read and Process	Members can query data (based on row-level filters) and run Process and Process All operations, but cannot modify the model schema.
Process	Members can run Process and Process All operations. Cannot modify the model schema and cannot query data.
Administrator	Members can modify the model schema and query all data.

- If the role you are creating has Read or Read and Process permission, you can add row filters by using a DAX formula. Click the **Row Filters** tab, then select a table, then click the **DAX Filter** field, and then type a DAX formula.
- Click **Members > Add External**.
- In **Add External Member**, enter users or groups in your tenant Azure AD by email address. After you click OK and close Role Manager, roles and role members appear in Tabular Model Explorer.



- Deploy to your Azure Analysis Services server.

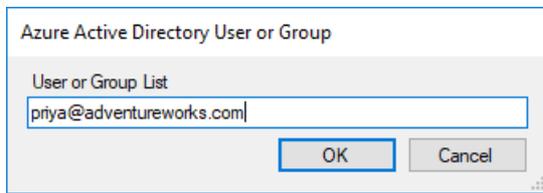
To add or manage roles and users in SSMS

To add roles and users to a deployed model database, you must be connected to the server as a Server administrator or already in a database role with administrator permissions.

- In Object Explorer, right-click **Roles > New Role**.
- In **Create Role**, enter a role name and description.
- Select a permission.

PERMISSION	DESCRIPTION
Full control (Administrator)	Members can modify the model schema, process, and can query all data.
Process database	Members can run Process and Process All operations. Cannot modify the model schema and cannot query data.
Read	Members can query data (based on row filters) but cannot modify the model schema.

- Click **Membership**, then enter a user or group in your tenant Azure AD by email address.



5. If the role you are creating has Read permission, you can add row filters by using a DAX formula. Click **Row Filters**, select a table, and then type a DAX formula in the **DAX Filter** field.

To add roles and users by using a TMSL script

You can run a TMSL script in the XMLA window in SSMS or by using PowerShell. Use the [CreateOrReplace](#) command and the [Roles](#) object.

Sample TMSL script

In this sample, a B2B external user and a group are added to the Analyst role with Read permissions for the SalesBI database. Both the external user and group must be in same tenant Azure AD.

```
{
  "createOrReplace": {
    "object": {
      "database": "SalesBI",
      "role": "Analyst"
    },
    "role": {
      "name": "Users",
      "description": "All allowed users to query the model",
      "modelPermission": "read",
      "members": [
        {
          "memberName": "user1@contoso.com",
          "identityProvider": "AzureAD"
        },
        {
          "memberName": "group1@adventureworks.com",
          "identityProvider": "AzureAD"
        }
      ]
    }
  }
}
```

To add roles and users by using PowerShell

The [SqlServer](#) module provides task-specific database management cmdlets and the general-purpose `Invoke-ASCmd` cmdlet that accepts a Tabular Model Scripting Language (TMSL) query or script. The following cmdlets are used for managing database roles and users.

CMDLET	DESCRIPTION
Add-RoleMember	Add a member to a database role.
Remove-RoleMember	Remove a member from a database role.
Invoke-ASCmd	Execute a TMSL script.

Row filters

Row filters define which rows in a table can be queried by members of a particular role. Row filters are defined for each table in a model by using DAX formulas.

Row filters can be defined only for roles with Read and Read and Process permissions. By default, if a row filter is not defined for a particular table, members can query all rows in the table unless cross-filtering applies from another table.

Row filters require a DAX formula, which must evaluate to a TRUE/FALSE value, to define the rows that can be queried by members of that particular role. Rows not included in the DAX formula cannot be queried. For example, the Customers table with the following row filters expression, `=Customers [Country] = "USA"`, members of the Sales role can only see customers in the USA.

Row filters apply to the specified rows and related rows. When a table has multiple relationships, filters apply security for the relationship that is active. Row filters are intersected with other row filters defined for related tables, for example:

TABLE	DAX EXPRESSION
Region	<code>=Region[Country]="USA"</code>
ProductCategory	<code>=ProductCategory[Name]="Bicycles"</code>
Transactions	<code>=Transactions[Year]=2016</code>

The net effect is members can query rows of data where the customer is in the USA, the product category is bicycles, and the year is 2016. Users cannot query transactions outside of the USA, transactions that are not bicycles, or transactions not in 2016 unless they are a member of another role that grants these permissions.

You can use the filter, `=FALSE()`, to deny access to all rows for an entire table.

Next steps

[Manage server administrators](#)

[Manage Azure Analysis Services with PowerShell](#)

[Tabular Model Scripting Language \(TMSL\) Reference](#)

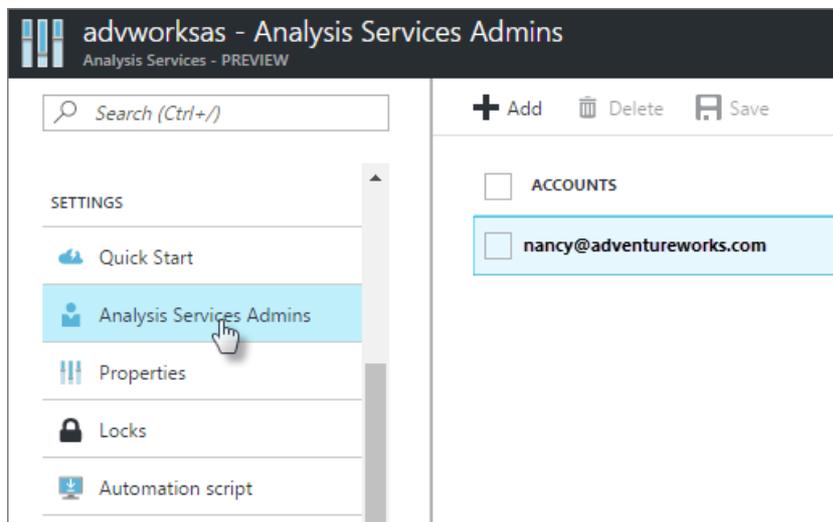
Manage server administrators

4/13/2018 • 1 min to read • [Edit Online](#)

Server administrators must be a valid user or group in the Azure Active Directory (Azure AD) for the tenant in which the server resides. You can use **Analysis Services Admins** for your server in Azure portal, or Server Properties in SSMS to manage server administrators.

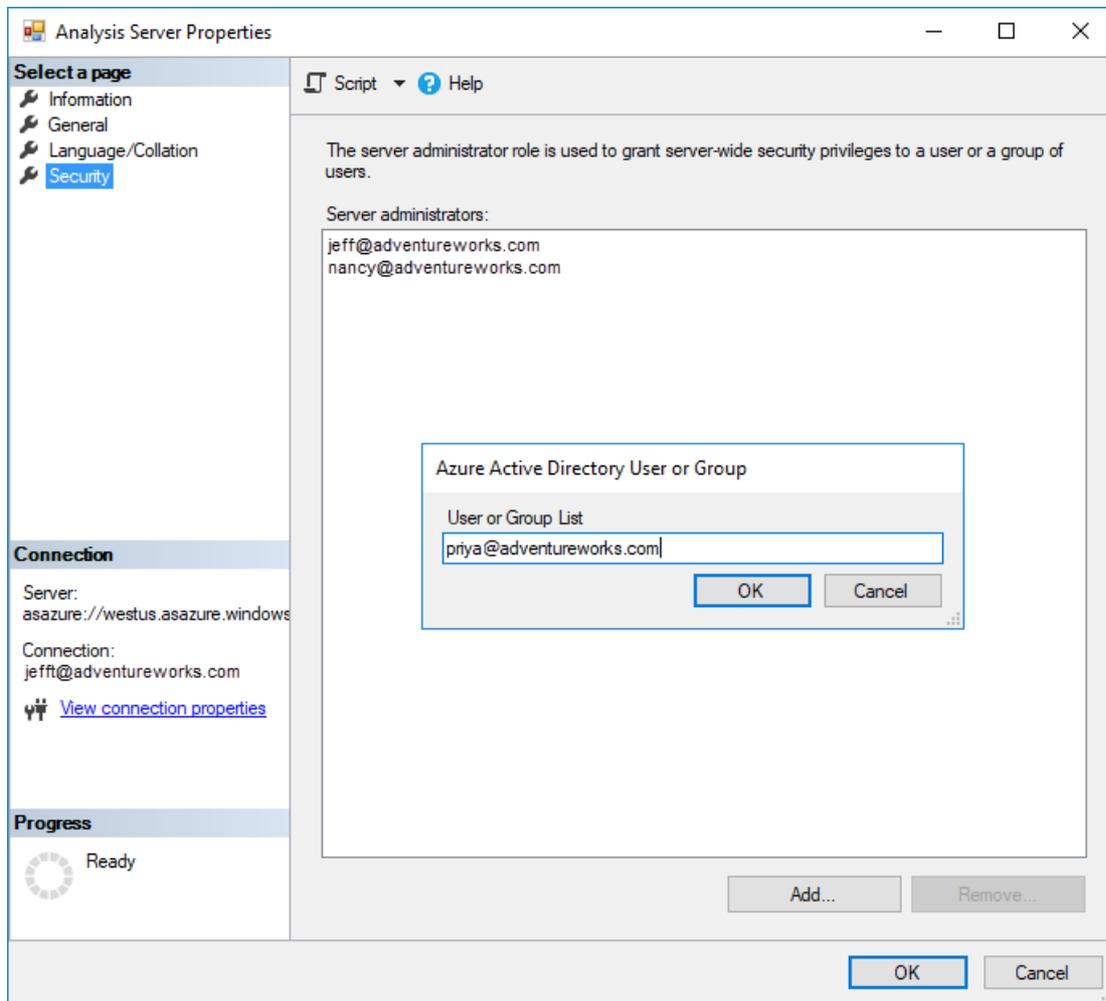
To add server administrators by using Azure portal

1. In the the portal, for your server, click **Analysis Services Admins**.
2. In **<servername> - Analysis Services Admins**, click **Add**.
3. In **Add Server Administrators**, select user accounts from your Azure AD or invite external users by email address.



To add server administrators by using SSMS

1. Right-click the server > **Properties**.
2. In **Analysis Server Properties**, click **Security**.
3. Click **Add**, and then enter the email address for a user or group in your Azure AD.



Next steps

[Authentication and user permissions](#)

[Manage database roles and users](#)

[Role-Based Access Control](#)

Monitor server metrics

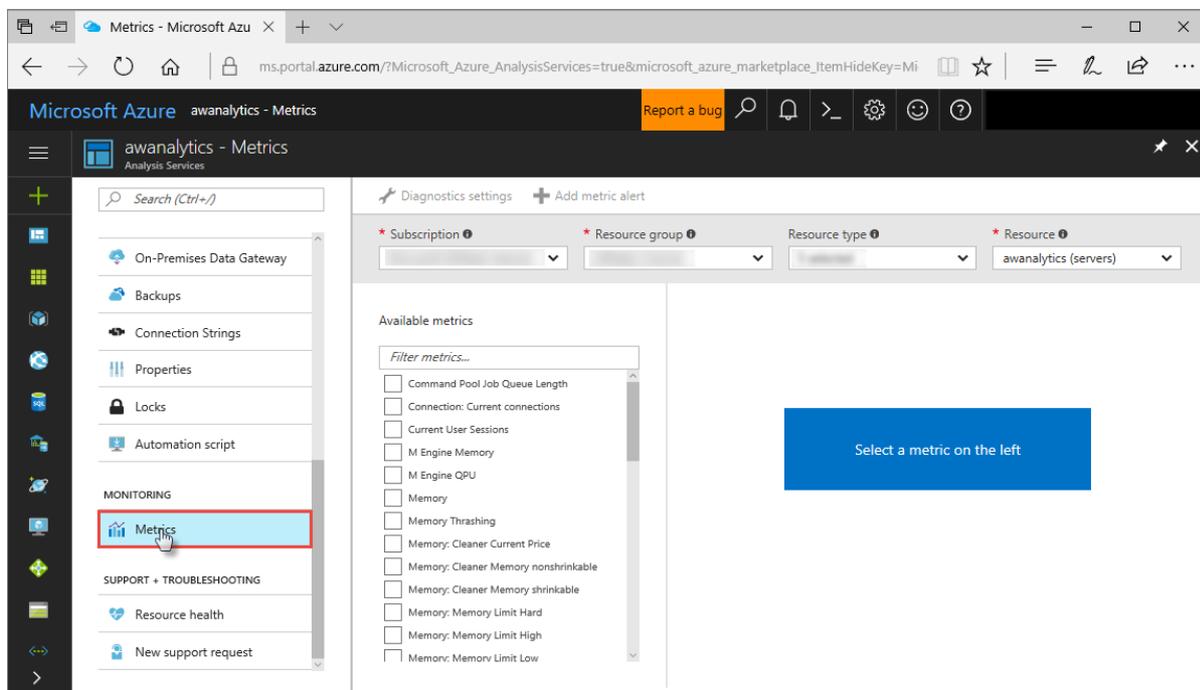
4/13/2018 • 4 min to read • [Edit Online](#)

Analysis Services provides metrics to help you monitor the performance and health of your servers. For example, monitor memory and CPU usage, number of client connections, and query resource consumption. Analysis Services uses the same monitoring framework as most other Azure services. To learn more, see [Metrics in Microsoft Azure](#).

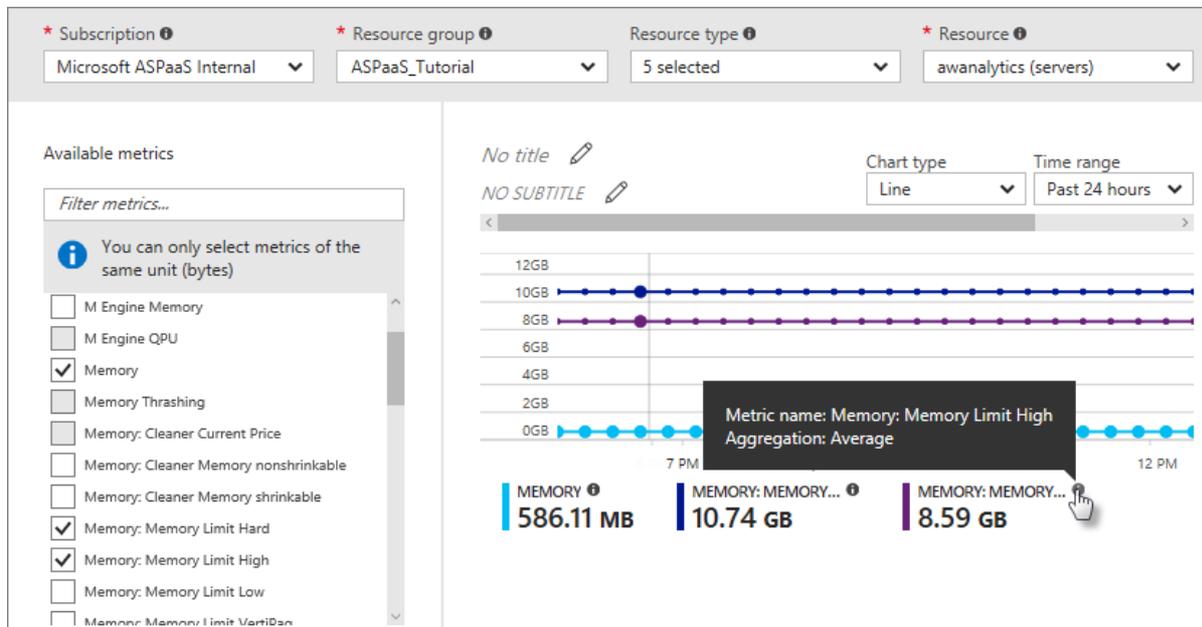
To perform more in-depth diagnostics, track performance, and identify trends across multiple service resources in a resource group or subscription, use [Azure Monitor](#). Azure Monitor (service) may result in a billable service.

To monitor metrics for an Analysis Services server

1. In Azure portal, select **Metrics**.



2. In **Available metrics**, select the metrics to include in your chart.



Server metrics

Use this table to determine which metrics are best for your monitoring scenario. Only metrics of the same unit can be shown on the same chart.

METRIC	METRIC DISPLAY NAME	UNIT	AGGREGATION TYPE	DESCRIPTION
CommandPoolJobQueueLength	Command Pool Job Queue Length	Count	Average	Number of jobs in the queue of the command thread pool.
CurrentConnections	Connection: Current connections	Count	Average	Current number of client connections established.
CurrentUserSessions	Current User Sessions	Count	Average	Current number of user sessions established.
mashup_engine_memory_metric	M Engine Memory	Bytes	Average	Memory usage by mashup engine processes
mashup_engine_qpu_metric	M Engine QPU	Count	Average	QPU usage by mashup engine processes
memory_metric	Memory	Bytes	Average	Memory. Range 0-25 GB for S1, 0-50 GB for S2 and 0-100 GB for S4
memory_thrashing_metric	Memory Thrashing	Percent	Average	Average memory thrashing.
CleanerCurrentPrice	Memory: Cleaner Current Price	Count	Average	Current price of memory, \$/byte/time, normalized to 1000.

METRIC	METRIC DISPLAY NAME	UNIT	AGGREGATION TYPE	DESCRIPTION
CleanerMemoryNonshrinkable	Memory: Cleaner Memory nonshrinkable	Bytes	Average	Amount of memory, in bytes, not subject to purging by the background cleaner.
CleanerMemoryShrinkable	Memory: Cleaner Memory shrinkable	Bytes	Average	Amount of memory, in bytes, subject to purging by the background cleaner.
MemoryLimitHard	Memory: Memory Limit Hard	Bytes	Average	Hard memory limit, from configuration file.
MemoryLimitHigh	Memory: Memory Limit High	Bytes	Average	High memory limit, from configuration file.
MemoryLimitLow	Memory: Memory Limit Low	Bytes	Average	Low memory limit, from configuration file.
MemoryLimitVertiPaq	Memory: Memory Limit VertiPaq	Bytes	Average	In-memory limit, from configuration file.
MemoryUsage	Memory: Memory Usage	Bytes	Average	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 in-memory analytics engine (VertiPaq) in excess of the engine Memory Limit.
Quota	Memory: Quota	Bytes	Average	Current memory quota, in bytes. Memory quota is also known as a memory grant or memory reservation.
QuotaBlocked	Memory: Quota Blocked	Count	Average	Current number of quota requests that are blocked until other memory quotas are freed.

METRIC	METRIC DISPLAY NAME	UNIT	AGGREGATION TYPE	DESCRIPTION
VertiPaqNonpaged	Memory: VertiPaq Nonpaged	Bytes	Average	Bytes of memory locked in the working set for use by the in-memory engine.
VertiPaqPaged	Memory: VertiPaq Paged	Bytes	Average	Bytes of paged memory in use for in-memory data.
ProcessingPoolJobQueueLength	Processing Pool Job Queue Length	Count	Average	Number of non-I/O jobs in the queue of the processing thread pool.
RowsConvertedPerSec	Processing: Rows converted per sec	CountPerSecond	Average	Rate of rows converted during processing.
RowsReadPerSec	Processing: Rows read per sec	CountPerSecond	Average	Rate of rows read from all relational databases.
RowsWrittenPerSec	Processing: Rows written per sec	CountPerSecond	Average	Rate of rows written during processing.
qpu_metric	QPU	Count	Average	QPU. Range 0-100 for S1, 0-200 for S2 and 0-400 for S4
QueryPoolBusyThreads	Query Pool Busy Threads	Count	Average	Number of busy threads in the query thread pool.
SuccessfulConnectionsPerSec	Successful Connections Per Sec	CountPerSecond	Average	Rate of successful connection completions.
CommandPoolBusyThreads	Threads: Command pool busy threads	Count	Average	Number of busy threads in the command thread pool.
CommandPoolIdleThreads	Threads: Command pool idle threads	Count	Average	Number of idle threads in the command thread pool.
LongParsingBusyThreads	Threads: Long parsing busy threads	Count	Average	Number of busy threads in the long parsing thread pool.
LongParsingIdleThreads	Threads: Long parsing idle threads	Count	Average	Number of idle threads in the long parsing thread pool.

METRIC	METRIC DISPLAY NAME	UNIT	AGGREGATION TYPE	DESCRIPTION
LongParsingJobQueueLength	Threads: Long parsing job queue length	Count	Average	Number of jobs in the queue of the long parsing thread pool.
ProcessingPoolIOJobQueueLength	Threads: Processing pool I/O job queue length	Count	Average	Number of I/O jobs in the queue of the processing thread pool.
ProcessingPoolBusyIOJobThreads	Threads: Processing pool busy I/O job threads	Count	Average	Number of threads running I/O jobs in the processing thread pool.
ProcessingPoolBusyNonIOThreads	Threads: Processing pool busy non-I/O threads	Count	Average	Number of threads running non-I/O jobs in the processing thread pool.
ProcessingPoolIdleIOJobThreads	Threads: Processing pool idle I/O job threads	Count	Average	Number of idle threads for I/O jobs in the processing thread pool.
ProcessingPoolIdleNonIOThreads	Threads: Processing pool idle non-I/O threads	Count	Average	Number of idle threads in the processing thread pool dedicated to non-I/O jobs.
QueryPoolIdleThreads	Threads: Query pool idle threads	Count	Average	Number of idle threads for I/O jobs in the processing thread pool.
QueryPoolJobQueueLength	Threads: Query pool job queue length	Count	Average	Number of jobs in the queue of the query thread pool.
ShortParsingBusyThreads	Threads: Short parsing busy threads	Count	Average	Number of busy threads in the short parsing thread pool.
ShortParsingIdleThreads	Threads: Short parsing idle threads	Count	Average	Number of idle threads in the short parsing thread pool.
ShortParsingJobQueueLength	Threads: Short parsing job queue length	Count	Average	Number of jobs in the queue of the short parsing thread pool.
TotalConnectionFailures	Total Connection Failures	Count	Average	Total failed connection attempts.
TotalConnectionRequests	Total Connection Requests	Count	Average	Total connection requests.

Next steps

[Monitoring in Microsoft Azure](#)

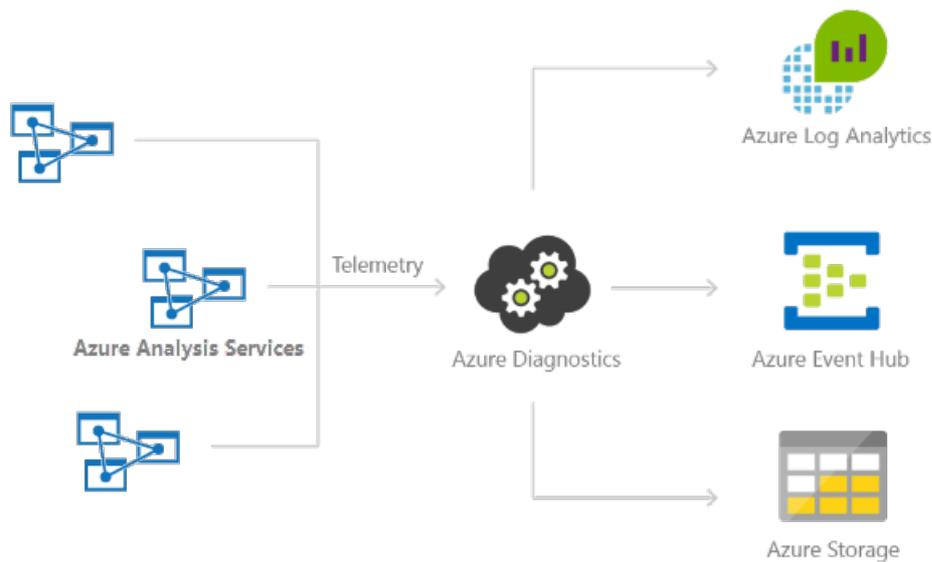
[Metrics in Microsoft Azure](#)

[Metrics in Azure Monitor REST API](#)

Setup diagnostic logging

4/18/2018 • 8 min to read • [Edit Online](#)

An important part of any Analysis Services solution is monitoring how your servers are performing. With [Azure resource diagnostic logs](#), you can monitor and send logs to [Azure Storage](#), stream them to [Azure Event Hubs](#), and export them to [Log Analytics](#), a service of [Azure](#).



What's logged?

You can select **Engine**, **Service**, and **Metrics** categories.

Engine

Selecting **Engine** logs all [xEvents](#). You cannot select individual events.

XEVENT CATEGORIES	EVENT NAME
Security Audit	Audit Login
Security Audit	Audit Logout
Security Audit	Audit Server Starts And Stops
Progress Reports	Progress Report Begin
Progress Reports	Progress Report End
Progress Reports	Progress Report Current
Queries	Query Begin
Queries	Query End

XEVENT CATEGORIES	EVENT NAME
Commands	Command Begin
Commands	Command End
Errors & Warnings	Error
Discover	Discover End
Notification	Notification
Session	Session Initialize
Locks	Deadlock
Query Processing	VertiPaq SE Query Begin
Query Processing	VertiPaq SE Query End
Query Processing	VertiPaq SE Query Cache Match
Query Processing	Direct Query Begin
Query Processing	Direct Query End

Service

OPERATION NAME	OCCURS WHEN
CreateGateway	User configures a gateway on server
ResumeServer	Resume a server
SuspendServer	Pause a server
DeleteServer	Delete a server
RestartServer	User restarts a server through SSMS or PowerShell
GetServerLogFiles	User exports server log through PowerShell
ExportModel	User exports a model in the portal by using Open in Visual Studio

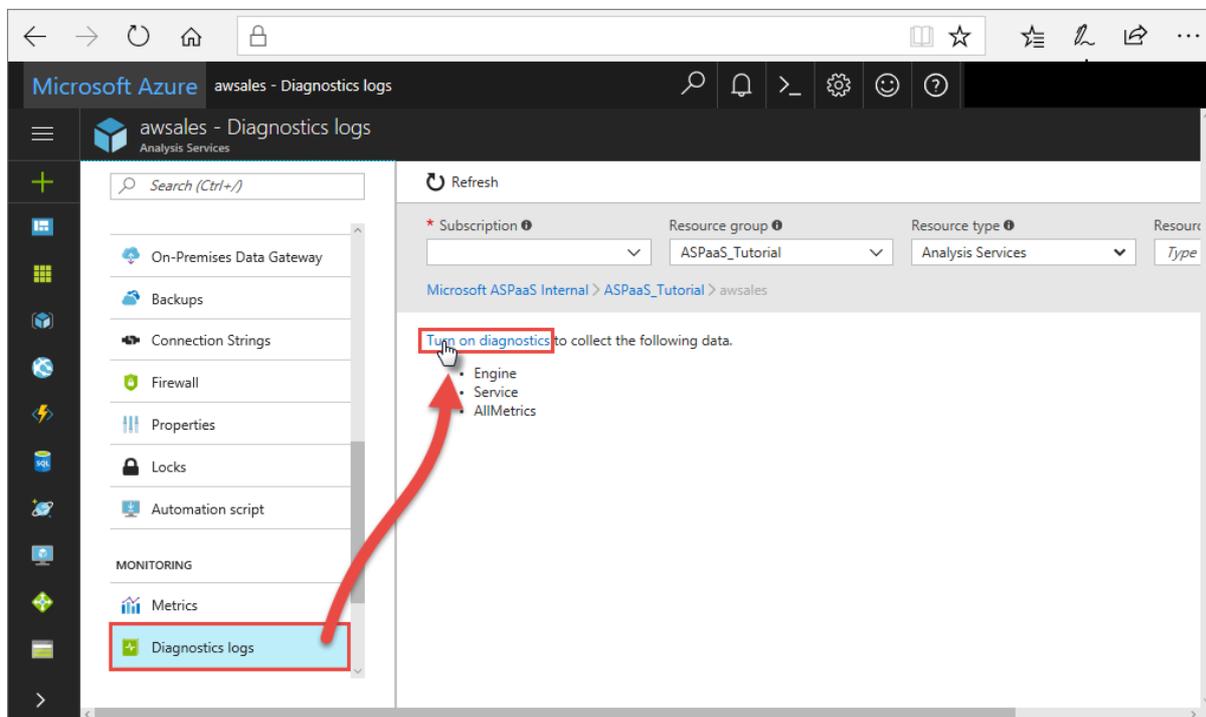
All metrics

The Metrics category logs the same [Server metrics](#) displayed in Metrics.

Setup diagnostics logging

Azure portal

1. In [Azure portal](#) > server, click **Diagnostic logs** in the left navigation, and then click **Turn on diagnostics**.



2. In **Diagnostics settings**, specify the following options:

- **Name.** Enter a name for the logs to create.
- **Archive to a storage account.** To use this option, you need an existing storage account to connect to. See [Create a storage account](#). Follow the instructions to create a Resource Manager, general-purpose account, then select your storage account by returning to this page in the portal. It may take a few minutes for newly created storage accounts to appear in the drop-down menu.
- **Stream to an event hub.** To use this option, you need an existing Event Hub namespace and event hub to connect to. To learn more, see [Create an Event Hubs namespace and an event hub using the Azure portal](#). Then return to this page in the portal to select the Event Hub namespace and policy name.
- **Send to Log Analytics.** To use this option, either use an existing workspace or create a new Log Analytics workspace by following the steps to [create a new workspace](#) in the portal. For more information on viewing your logs in Log Analytics, see [View logs in Log Analytics](#).
- **Engine.** Select this option to log xEvents. If you're archiving to a storage account, you can select the retention period for the diagnostic logs. Logs are autodeleted after the retention period expires.
- **Service.** Select this option to log service level events. If you are archiving to a storage account, you can select the retention period for the diagnostic logs. Logs are autodeleted after the retention period expires.
- **Metrics.** Select this option to store verbose data in [Metrics](#). If you are archiving to a storage account, you can select the retention period for the diagnostic logs. Logs are autodeleted after the retention period expires.

3. Click **Save**.

If you receive an error that says "Failed to update diagnostics for <workspace name>. The subscription <subscription id> is not registered to use microsoft.insights." follow the [Troubleshoot Azure Diagnostics](#) instructions to register the account, then retry this procedure.

If you want to change how your diagnostic logs are saved at any point in the future, you can return to this page to modify settings.

PowerShell

Here are the basic commands to get you going. If you want step-by-step help on setting up logging to a storage account by using PowerShell, see the tutorial later in this article.

To enable metrics and diagnostics logging by using PowerShell, use the following commands:

- To enable storage of diagnostics logs in a storage account, use this command:

```
Set-AzureRmDiagnosticSetting -ResourceId [your resource id] -StorageAccountId [your storage account id]
-Enabled $true
```

The storage account ID is the resource ID for the storage account where you want to send the logs.

- To enable streaming of diagnostics logs to an event hub, use this command:

```
Set-AzureRmDiagnosticSetting -ResourceId [your resource id] -ServiceBusRuleId [your service bus rule id]
-Enabled $true
```

The Azure Service Bus rule ID is a string with this format:

```
{service bus resource ID}/authorizationrules/{key name}
```

- To enable sending diagnostics logs to a Log Analytics workspace, use this command:

```
Set-AzureRmDiagnosticSetting -ResourceId [your resource id] -WorkspaceId [resource id of the log
analytics workspace] -Enabled $true
```

- You can obtain the resource ID of your Log Analytics workspace by using the following command:

```
(Get-AzureRmOperationalInsightsWorkspace).ResourceId
```

You can combine these parameters to enable multiple output options.

REST API

Learn how to [change diagnostics settings by using the Azure Monitor REST API](#).

Resource Manager template

Learn how to [enable diagnostics settings at resource creation by using a Resource Manager template](#).

Manage your logs

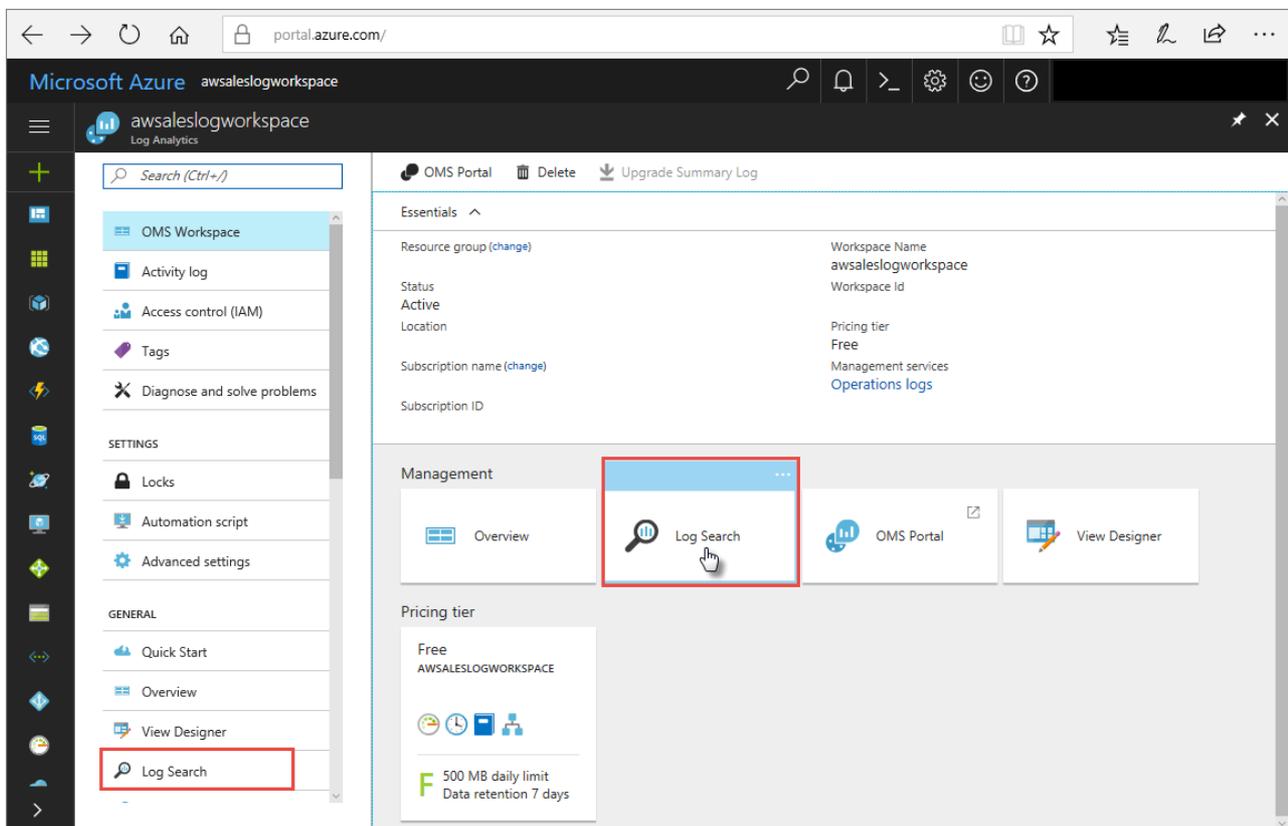
Logs are typically available within a couple hours of setting up logging. It's up to you to manage your logs in your storage account:

- Use standard Azure access control methods to secure your logs by restricting who can access them.
- Delete logs that you no longer want to keep in your storage account.
- Be sure to set a retention period for so old logs are deleted from your storage account.

View logs in Log Analytics

Metrics and server events are integrated with xEvents in Log Analytics for side-by-side analysis. Log Analytics can also be configured to receive events from other Azure services providing a holistic view of diagnostic logging data across your architecture.

To view your diagnostic data in Log Analytics, open the Log Search page from the left menu or the Management area, as shown below.



Now that you've enabled data collection, in **Log Search**, click **All collected data**.

In **Type**, click **AzureDiagnostics**, and then click **Apply**. AzureDiagnostics includes Engine and Service events. Notice a Log Analytics query is created on-the-fly. The `EventClass_s` field contains xEvent names, which may look familiar if you've used xEvents for on-premises logging.

Click **EventClass_s** or one of the event names and Log Analytics continues constructing a query. Be sure to save your queries to reuse later.

Be sure to see Log Analytics, which provides a website with enhanced query, dashboarding, and alerting capabilities on collected data.

Queries

There are hundreds of queries you can use. Here are a few to get you started. To learn more about using the new Log Search query language, see [Understanding log searches in Log Analytics](#).

- Query return queries submitted to Azure Analysis Services that took over five minutes (300,000 milliseconds) to complete.

```
search * | where ( Type == "AzureDiagnostics" ) | where ( EventClass_s == "QUERY_END" ) | where toint(Duration_s) > 300000
```

- Identify scale out replicas.

```
search * | summarize count() by ServerName_s
```

When using scale-out, you can identify read-only replicas because the `ServerName_s` field values have the replica instance number appended to the name. The resource field contains the Azure resource name, which matches the server name that the users see. The `IsQueryScaleoutReadonlyInstance_s` field equals true for replicas.

TIP

Have a great Log Analytics query you want to share? If you have a GitHub account, you can add it to this article. Just click **Edit** at the top-right of this page.

Tutorial - Turn on logging by using PowerShell

In this quick tutorial, you create a storage account in the same subscription and resource group as your Analysis Service server. You then use `Set-AzureRmDiagnosticSetting` to turn on diagnostics logging, sending output to the new storage account.

Prerequisites

To complete this tutorial, you must have the following resources:

- An existing Azure Analysis Services server. For instructions on creating a server resource, see [Create a server in Azure portal](#), or [Create an Azure Analysis Services server by using PowerShell](#).

Connect to your subscriptions

Start an Azure PowerShell session and sign in to your Azure account with the following command:

```
Connect-AzureRmAccount
```

In the pop-up browser window, enter your Azure account user name and password. Azure PowerShell gets all the subscriptions that are associated with this account and by default, uses the first one.

If you have multiple subscriptions, you might have to specify a specific one that was used to create your Azure Key Vault. Type the following to see the subscriptions for your account:

```
Get-AzureRmSubscription
```

Then, to specify the subscription that's associated with the Azure Analysis Services account you are logging, type:

```
Set-AzureRmContext -SubscriptionId <subscription ID>
```

NOTE

If you have multiple subscriptions associated with your account, it is important to specify the subscription.

Create a new storage account for your logs

You can use an existing storage account for your logs, provided it's in the same subscription as your server. For this tutorial you create a new storage account dedicated to Analysis Services logs. To make it easy, you're storing the storage account details in a variable named **sa**.

You also use the same resource group as the one that contains your Analysis Services server. Substitute values for `awsales_resgroup`, `awsaleslogs`, and `West Central US` with your own values:

```
$sa = New-AzureRmStorageAccount -ResourceGroupName awsales_resgroup `
-Name awsaleslogs -Type Standard_LRS -Location 'West Central US'
```

Identify the server account for your logs

Set the account name to a variable named **account**, where `ResourceName` is the name of the account.

```
$account = Get-AzureRmResource -ResourceGroupName awsales_resgroup `
-ResourceName awsales -ResourceType "Microsoft.AnalysisServices/servers"
```

Enable logging

To enable logging, use the Set-AzureRmDiagnosticSetting cmdlet together with the variables for the new storage account, server account, and the category. Run the following command, setting the **-Enabled** flag to **\$true**:

```
Set-AzureRmDiagnosticSetting -ResourceId $account.ResourceId -StorageAccountId $sa.Id -Enabled $true -
Categories Engine
```

The output should look something like this:

```
StorageAccountId      :
/subscriptions/a23279b5-xxxx-xxxx-xxxx-
47b7c6d423ea/resourceGroups/awsales_resgroup/providers/Microsoft.Storage/storageAccounts/awsaleslogs
ServiceBusRuleId      :
EventHubAuthorizationRuleId :
Metrics
  TimeGrain           : PT1M
  Enabled              : False
  RetentionPolicy
  Enabled              : False
  Days                 : 0

Logs
  Category            : Engine
  Enabled              : True
  RetentionPolicy
  Enabled              : False
  Days                 : 0

  Category            : Service
  Enabled              : False
  RetentionPolicy
  Enabled              : False
  Days                 : 0

WorkspaceId           :
Id                    : /subscriptions/a23279b5-xxxx-xxxx-xxxx-
47b7c6d423ea/resourceGroups/awsales_resgroup/providers/microsoft.analysis Servic
es/servers/awsales/providers/microsoft.insights/diagnosticSettings/service
Name                  : service
Type                  :
Location              :
Tags                  :
```

This confirms that logging is now enabled for the server, saving information to the storage account.

You can also set retention policy for your logs so older logs are automatically deleted. For example, set retention policy using **-RetentionEnabled** flag to **\$true**, and set **-RetentionInDays** parameter to **90**. Logs older than 90 days are automatically deleted.

```
Set-AzureRmDiagnosticSetting -ResourceId $account.ResourceId `
-StorageAccountId $sa.Id -Enabled $true -Categories Engine `
-RetentionEnabled $true -RetentionInDays 90
```

Next steps

Learn more about [Azure resource diagnostic logging](#).

See [Set-AzureRmDiagnosticSetting](#) in PowerShell help.

Use gateway for data sources on an Azure Virtual Network (VNet)

4/23/2018 • 1 min to read • [Edit Online](#)

This article describes the **AlwaysUseGateway** server property for use when data sources are on an [Azure Virtual Network \(VNet\)](#).

Server access to VNet data sources

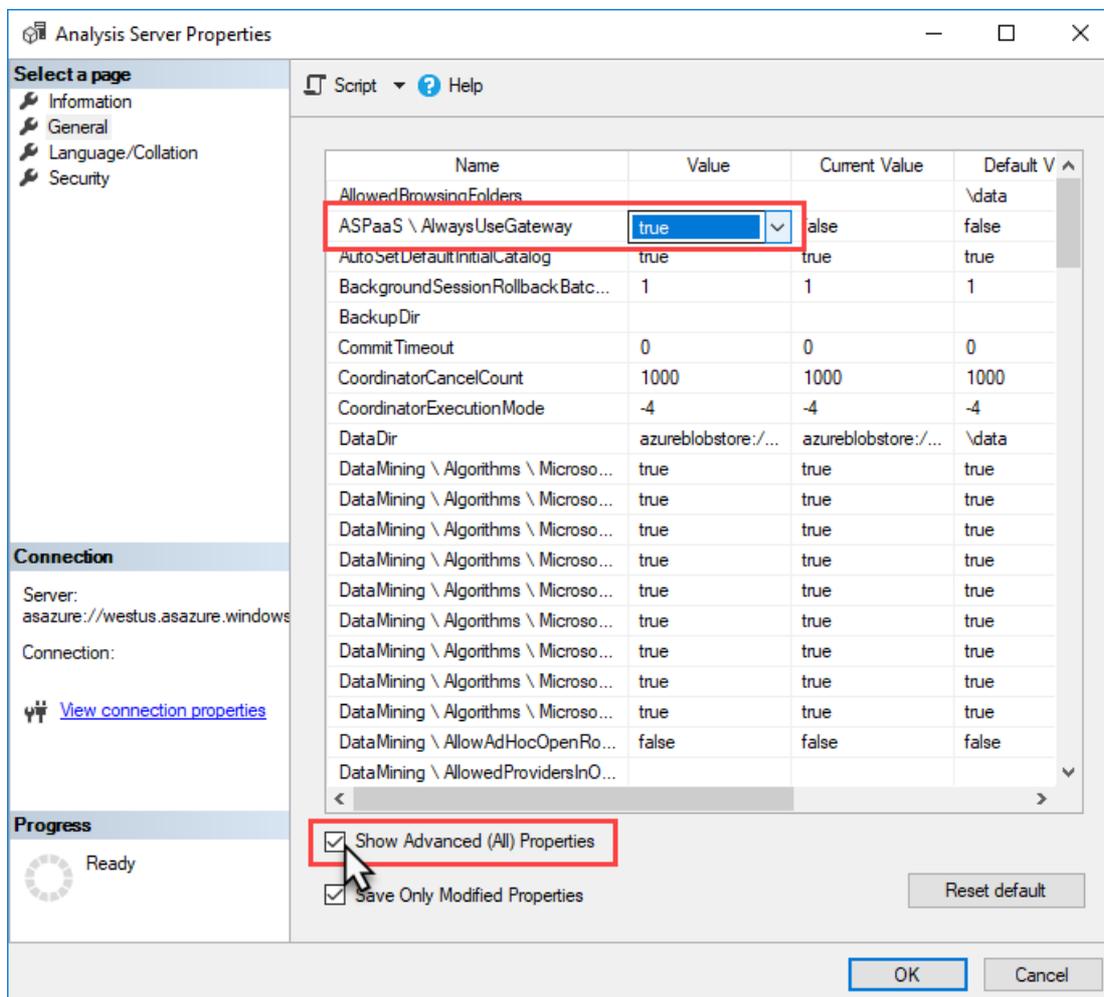
If your data sources are accessed through a VNet, your Azure Analysis Services server must connect to those data sources as if they are on-premises, in your own environment. You can configure the **AlwaysUseGateway** server property to specify the server to access all datasource data through an [On-premises gateway](#).

NOTE

This property is effective only when an [On-premises data gateway](#) is installed and configured. The gateway can be on the VNet.

Configure AlwaysUseGateway property

1. In SSMS > server > **Properties** > **General**, select **Show Advanced (All) Properties**.
2. In the **ASPaaS\AlwaysUseGateway**, select **true**.



See also

[Connecting to on-premises data sources](#)

[Install and configure an on-premises data gateway](#)

[Azure Virtual Network \(VNET\)](#)

Client libraries for connecting to Azure Analysis Services

4/13/2018 • 3 min to read • [Edit Online](#)

Client libraries are necessary for client applications and tools to connect to Analysis Services servers.

Download the latest client libraries (Windows Installer)

DOWNLOAD	PRODUCT VERSION
MSOLAP (amd64)	15.0.1.208
MSOLAP (x86)	15.0.1.208
AMO	15.0.2
ADOMD	15.0.2

AMO and ADOMD (NuGet packages)

Analysis Services Management Objects (AMO) and ADOMD client libraries are available as installable packages from [NuGet.org](#). It's recommended you migrate to NuGet references instead of using Windows Installer.

PACKAGE	PRODUCT VERSION
AMO	15.0.2.0
ADOMD	15.0.2.0

NuGet package assemblies AssemblyVersion follow semantic versioning: MAJOR.MINOR.PATCH. NuGet references load the expected version even if there is a different version in the GAC (resulting from MSI install). PATCH is incremented for each release. AMO and ADOMD versions are kept in-sync.

Understanding client libraries

Analysis Services utilize three client libraries, also known as data providers. 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. Typically, all three are installed at the same time. **Azure Analysis Services requires the latest versions of all three libraries.**

Microsoft client applications like Power BI Desktop and Excel install all three client libraries and update them when new versions are available. Depending on the version or frequency of updates, some client libraries may not be the latest versions required by Azure Analysis Services. The same applies to custom applications or other interfaces such as AsCmd, TOM, ADOMD.NET. These applications require manually or programmatically installing the libraries. The client libraries for manual installation are included in SQL Server feature packs as distributable packages. However, these client libraries are tied to the SQL Server version and may not be the latest.

Client libraries for client connections are different from data providers required to connect from an Azure

Analysis Services server to a data source. To learn more about datasource connections, see [Datasource connections](#).

Client library types

Analysis Services OLE DB Provider (MSOLAP)

Analysis Services OLE DB Provider (MSOLAP) is the native client library for Analysis Services database connections. It's used indirectly by both ADOMD.NET and AMO, delegating connection requests to the data provider. You can also call the OLE DB Provider directly from application code.

The Analysis Services OLE DB Provider is installed automatically by most tools and client applications used to access Analysis Services databases. It must be installed on computers used to access Analysis Services data.

OLE DB providers are often specified in connection strings. An Analysis Services connection string uses a different nomenclature to refer to the OLE DB Provider: MSOLAP.<version>.dll.

AMO

AMO is a managed client library used for server administration and data definition. It's installed and used by tools and client applications. For example, SQL Server Management Studio (SSMS) uses AMO to connect to Analysis Services. A connection using AMO is typically minimal, consisting of `"data source=\\<servername>"`. After a connection is established, you use the API to work with database collections and major objects. Both SSDT and SSMS use AMO to connect to an Analysis Services instance.

ADOMD

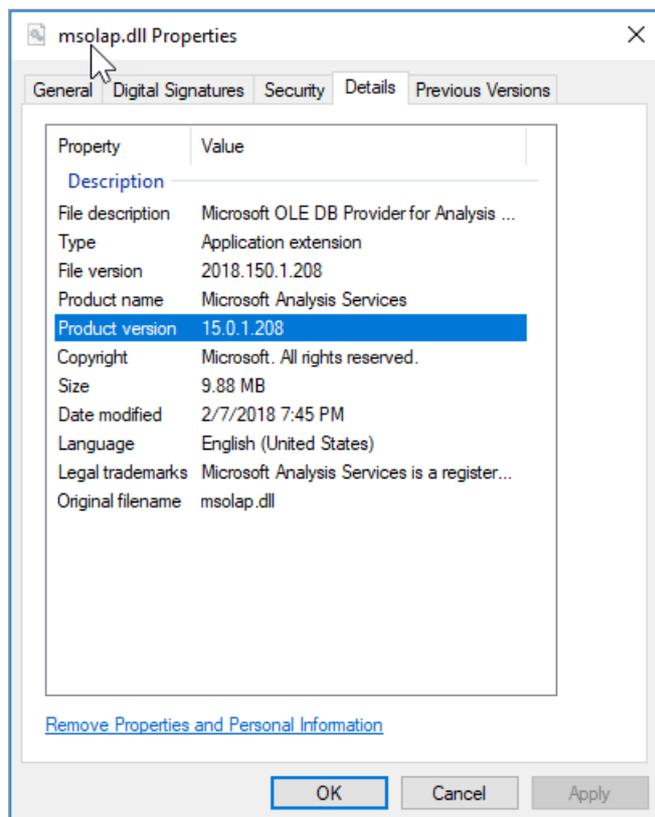
ADOMD.NET is a managed data client library used for querying Analysis Services data. It's installed and used by tools and client applications.

When connecting to a database, the connection string properties for all three libraries are similar. Almost any connection string you define for ADOMD.NET by using [Microsoft.AnalysisServices.AdomdClient.AdomdConnectionString](#) also works for AMO and the Analysis Services OLE DB Provider (MSOLAP). To learn more, see [Connection string properties \(Analysis Services\)](#).

How to determine client library version

OLEDB (MSOLAP)

1. Go to `C:\Program Files\Microsoft Analysis Services\AS OLEDB. If you have more than one folder, choose the higher number.
2. Right-click **msolap.dll** > **Properties** > **Details**. If the filename is msolap140.dll, it's older than latest version and should be upgraded.



AMO

1. Go to `C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.AnalysisServices\`. If you have more than one folder, choose the higher number.
2. Right-click **Microsoft.AnalysisServices** > **Properties** > **Details**.

ADOMD

1. Go to `C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.AnalysisServices.AdomdClient\`. If you have more than one folder, choose the higher number.
2. Right-click **Microsoft.AnalysisServices.AdomdClient** > **Properties** > **Details**.

Next steps

[Connect with Excel](#)

[Connect with Power BI](#)

Connect with Excel

4/13/2018 • 1 min to read • [Edit Online](#)

Once you've created a server in Azure, and deployed a tabular model to it, you're ready to connect and begin exploring data.

Connect in Excel

Connecting to a server in Excel is supported by using Get Data in Excel 2016. Connecting by using the Import Table Wizard in Power Pivot is not supported.

To connect in Excel 2016

1. In Excel 2016, on the **Data** ribbon, click **Get External Data > From Other Sources > From Analysis Services**.
2. In the Data Connection Wizard, in **Server name**, enter the server name including protocol and URI. Then, in **Logon credentials**, select **Use the following User Name and Password**, and then type the organizational user name, for example nancy@adventureworks.com, and password.

NOTE

If you sign in with a Microsoft Account, Live ID, Yahoo, Gmail, etc., or you are required to sign in with multi-factor authentication, leave the password field blank. You are prompted for a password after clicking Next.

Data Connection Wizard

Connect to Database Server

Enter the information required to connect to the database server.

1. Server name: asazure://westcentralus.asazure.windows.net/adv

2. Log on credentials

Use Windows Authentication

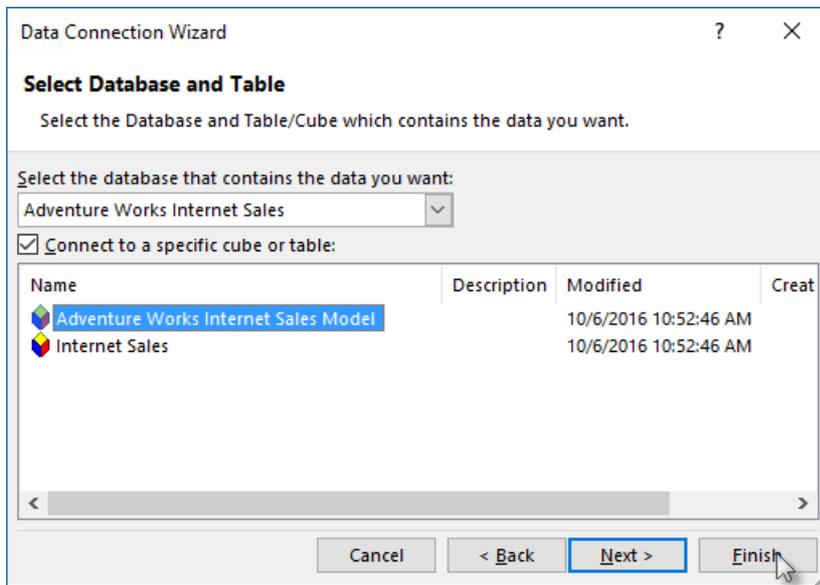
Use the following User Name and Password

User Name: nancy@adventureworks.com

Password: ●●●●●●●●●●

Cancel < Back Next > Finish

3. In **Select Database and Table**, select the database and model or perspective, and then click **Finish**.



See also

[Client libraries](#)

[Manage your server](#)

Connect with Power BI

4/13/2018 • 1 min to read • [Edit Online](#)

Once you've created a server in Azure, and deployed a tabular model to it, users in your organization are ready to connect and begin exploring data.

TIP

Be sure to use the latest version of [Power BI Desktop](#).

Connect in Power BI Desktop

1. In Power BI Desktop, click **Get Data** > **Azure** > **Azure Analysis Services database**.
2. In **Server**, enter the server name. Be sure to include the full URL; for example, `asazure://westcentralus.asazure.windows.net/advworks`.
3. In **Database**, if you know the name of the tabular model database or perspective you want to connect to, paste it here. Otherwise, you can leave this field blank and select a database or perspective later.
4. Select a connection option and then press **Connect**.

Both **Connect live** and **Import** options are supported. However, we recommended you use live connections because Import mode does have some limitations; most notably, server performance might be impacted during import. Also, if the model is to be refreshed in the Power BI service, the **Allow access from Power BI** setting applies only when choosing **Connect live**.

5. If prompted, enter your login credentials.
6. In **Navigator**, expand the server, then select the model or perspective you want to connect to, and then click **Connect**. Click a model or perspective to show all objects for that view.

The model opens in Power BI Desktop with a blank report in Report view. The Fields list displays all non-hidden model objects. Connection status is displayed in the lower-right corner.

Connect in Power BI (service)

1. Create a Power BI Desktop file that has a live connection to your model on your server.
2. In [Power BI](#), click **Get Data** > **Files**, then locate and select your .pbix file.

See also

[Connect to Azure Analysis Services](#)

[Client libraries](#)

Create an Office Data Connection file

4/13/2018 • 1 min to read • [Edit Online](#)

Information in this article describes how you can create an Office Data Connection file to connect to an Azure Analysis Services server from Excel 2016 version number 16.0.7369.2117 or earlier, or Excel 2013. An updated [MSOLAP.7 provider](#) is also required.

1. Copy the sample connection file below and paste into a text editor.
2. In `odc:ConnectionString`, change the following properties:
 - In `Data Source=asazure://<region>.asazure.windows.net/<servername>;` change `<region>` to the region of your Analysis Services server and `<servername>` to the name of your server.
 - In `Initial Catalog=<database>;` change `<database>` to the name of your database.
3. In `<odc:CommandText>Model</odc:CommandText>` change `Model` to the name of your model or perspective.
4. Save the file with an `.odc` extension to the `C:\Users\username\Documents\My Data Sources` folder.
5. Right-click the file, and then click **Open in Excel**. Or in Excel, on the **Data** ribbon, click **Existing Connections**, select your file, and then click **Open**.

Sample connection file

```
<html xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns="http://www.w3.org/TR/REC-html40">

<head>
<meta http-equiv=Content-Type content="text/x-ms-odc; charset=utf-8">
<meta name=ProgId content=ODC.Cube>
<meta name=SourceType content=OLEDB>
<meta name=Catalog content="Database">
<meta name=Table content=Model>
<title>AzureAnalysisServicesConnection</title>
<xml id=docprops><o:DocumentProperties
  xmlns:o="urn:schemas-microsoft-com:office:office"
  xmlns="http://www.w3.org/TR/REC-html40">
  <o:Name>SampleAzureAnalysisServices</o:Name>
</o:DocumentProperties>
</xml><xml id=msodc><odc:OfficeDataConnection
  xmlns:odc="urn:schemas-microsoft-com:office:odc"
  xmlns="http://www.w3.org/TR/REC-html40">
  <odc:Connection odc:Type="OLEDB">
  <odc:ConnectionString>Provider=MSOLAP.7;Data
Source=asazure://<region>.asazure.windows.net/<servername>;Initial Catalog=<database>;</odc:ConnectionString>
  <odc:CommandType>Cube</odc:CommandType>
  <odc:CommandText>Model</odc:CommandText>
  </odc:Connection>
</odc:OfficeDataConnection>
</xml>
<style>
<!--
  .ODCDataSource
  {
  behavior: url(dataconn.htc);
  }
-->
</style>
```

```

</head>

<body onload='init()' scroll=no leftmargin=0 topmargin=0 rightmargin=0 style='border: 0px'>
<table style='border: solid 1px threedface; height: 100%; width: 100%' cellpadding=0 cellspacing=0
width='100%'>
  <tr>
    <td id=tdName style='font-family:arial; font-size:medium; padding: 3px; background-color: threedface'>
      &nbsp;
    </td>
    <td id=tdTableDropdown style='padding: 3px; background-color: threedface; vertical-align: top; padding-
bottom: 3px'>

      &nbsp;
    </td>
  </tr>
  <tr>
    <td id=tdDesc colspan='2' style='border-bottom: 1px threedshadow solid; font-family: Arial; font-size:
1pt; padding: 2px; background-color: threedface'>

      &nbsp;
    </td>
  </tr>
  <tr>
    <td colspan='2' style='height: 100%; padding-bottom: 4px; border-top: 1px threedhighlight solid;'>
      <div id='pt' style='height: 100%' class='ODCDataSource'></div>
    </td>
  </tr>
</table>

<script language='javascript'>

function init() {
  var sName, sDescription;
  var i, j;

  try {
    sName = unescape(location.href)

    i = sName.lastIndexOf(".")
    if (i>=0) { sName = sName.substring(1, i); }

    i = sName.lastIndexOf("/")
    if (i>=0) { sName = sName.substring(i+1, sName.length); }

    document.title = sName;
    document.getElementById("tdName").innerText = sName;

    sDescription = document.getElementById("docprops").innerHTML;

    i = sDescription.indexOf("escription>")
    if (i>=0) { j = sDescription.indexOf("escription>", i + 11); }

    if (i>=0 && j >= 0) {
      j = sDescription.lastIndexOf("</", j);

      if (j>=0) {
        sDescription = sDescription.substring(i+11, j);
        if (sDescription != "") {
          document.getElementById("tdDesc").style.fontSize="x-small";
          document.getElementById("tdDesc").innerHTML = sDescription;
        }
      }
    }
  }
  catch(e) {

  }
}

```

```
</script>
```

```
</body>
```

```
</html>
```

Tutorial: Add a sample model

4/13/2018 • 1 min to read • [Edit Online](#)

In this tutorial, you add a sample Adventure Works model to your server. The sample model is a completed version of the Adventure Works Internet Sales (1200) data modeling tutorial. A sample model is useful for testing model management, connecting with tools and client applications, and querying model data.

Before you begin

To complete this tutorial, you need:

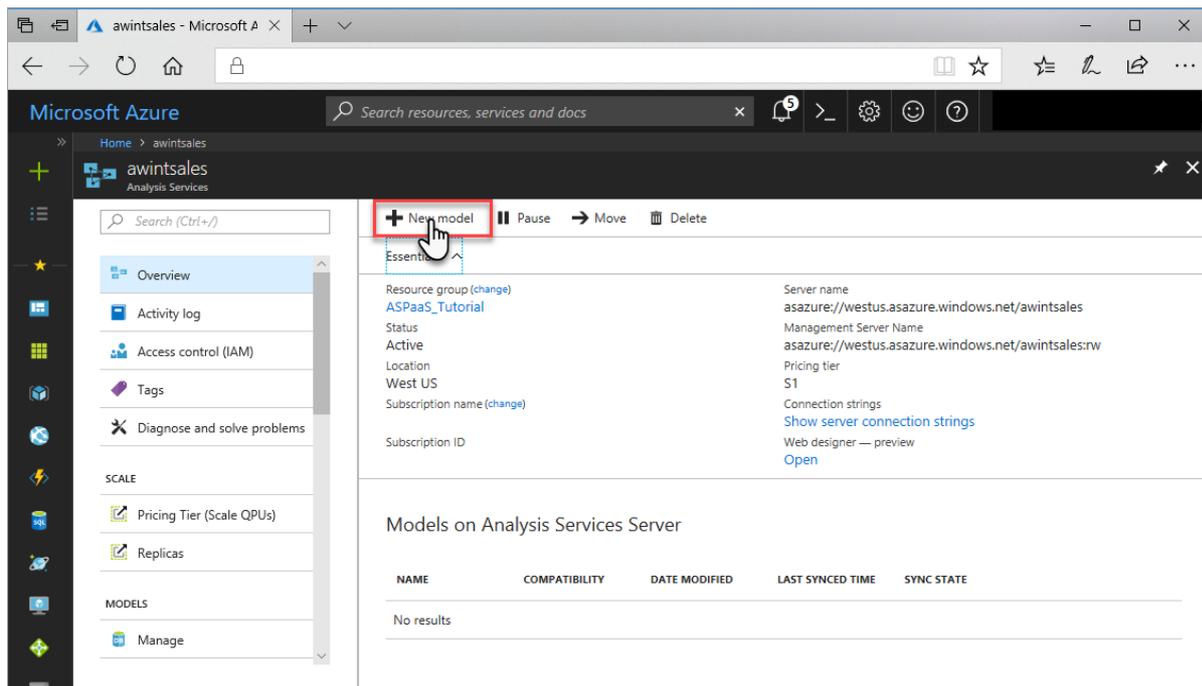
- An Azure Analysis Services server
- Server administrator permissions

Sign in to the Azure portal

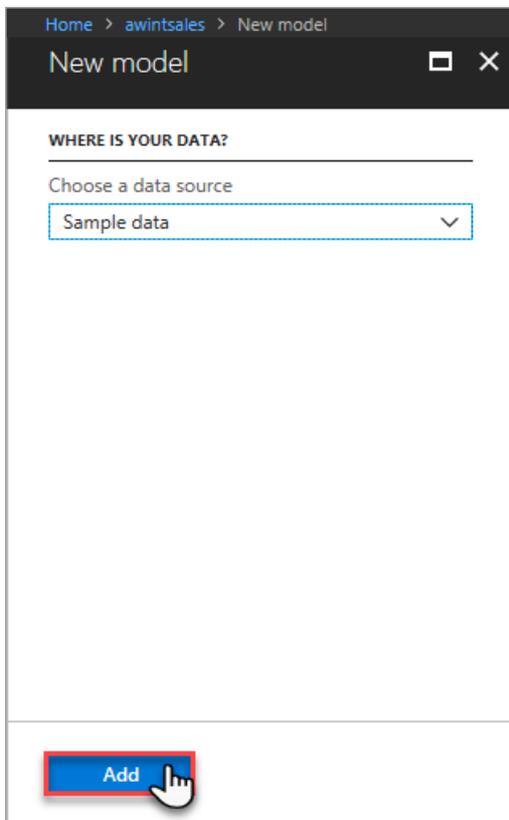
Sign in to the [Azure portal](#).

Create a sample model

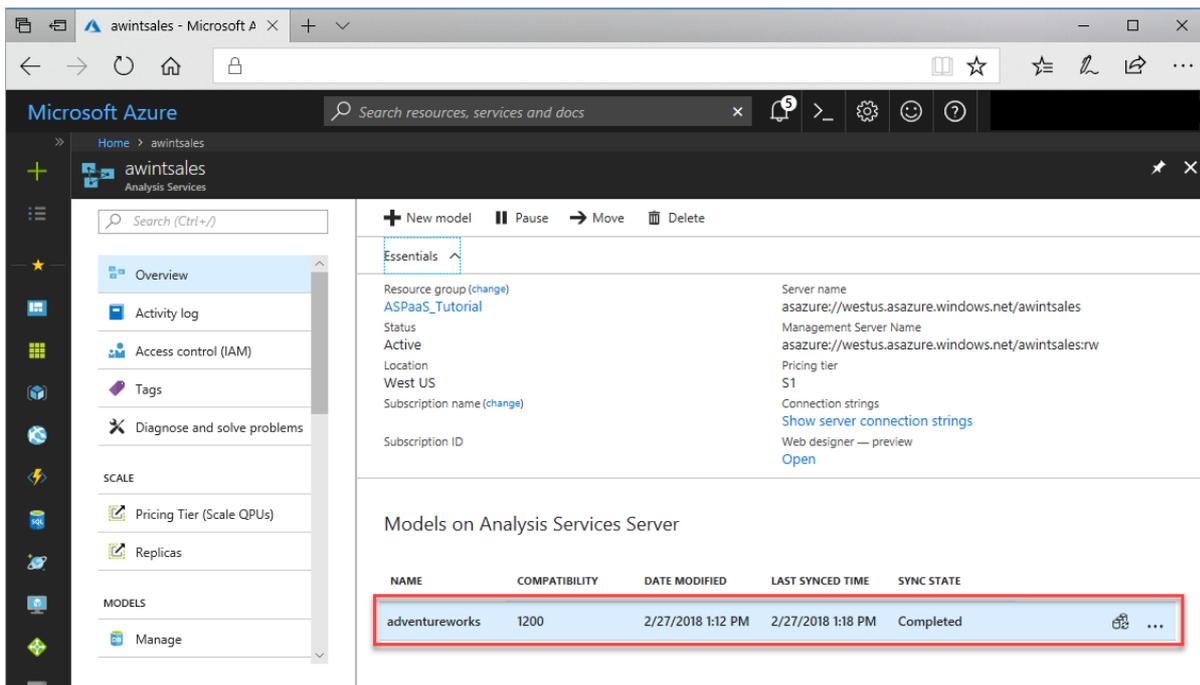
1. In server **Overview**, click **New model**.



2. In **New model** > **Choose a datasource**, verify **Sample data** is selected, and then click **Add**.



3. In **Overview**, verify the `adventureworks` sample is created.



Clean up resources

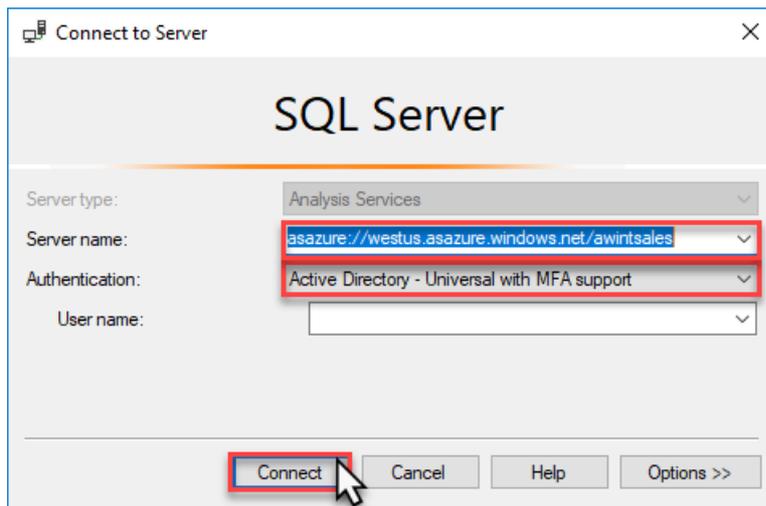
Your sample model is using cache memory resources. If you are not using your sample model for testing, you should remove it from your server.

NOTE

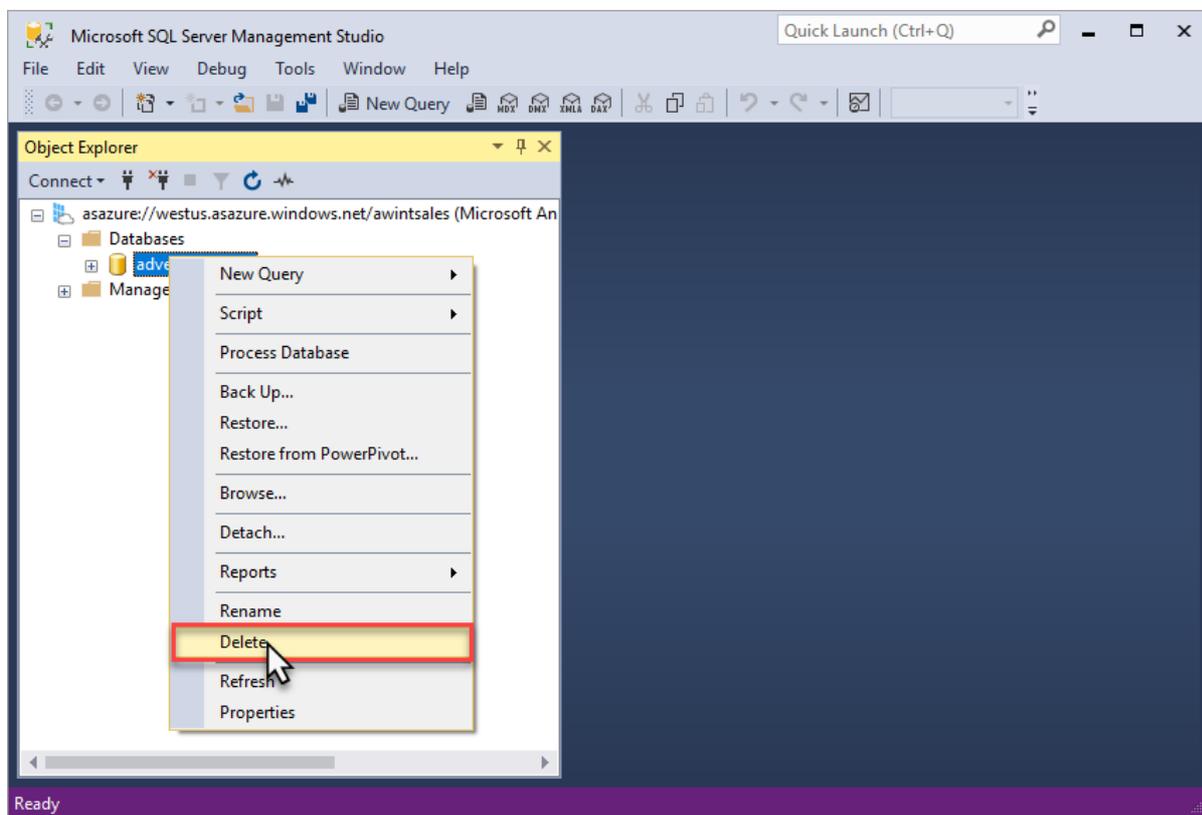
These steps describe how to delete a model from a server by using SSMS. You can also delete a model by using the preview Web designer feature.

1. In SSMS > **Object Explorer**, click **Connect** > **Analysis Services**.

2. In **Connect to Server**, paste in the server name, then in **Authentication**, choose **Active Directory - Universal with MFA support**, enter your username, and then click **Connect**.



3. In **Object Explorer**, right-click the `adventureworks` sample database, and then click **Delete**.



Next steps

[Connect in Power BI Desktop](#)

[Manage database roles and users](#)

Azure Analysis Services - Adventure Works tutorial

4/13/2018 • 3 min to read • [Edit Online](#)

This tutorial provides lessons on how to create and deploy a tabular model at the 1400 compatibility level by using Visual Studio with [SQL Server Data Tools \(SSDT\)](#).

If you're new to Analysis Services and tabular modeling, completing this tutorial is the quickest way to learn how to create and deploy a basic tabular model by using Visual Studio. Once you have the prerequisites in-place, it should take between two to three hours to complete.

What you learn

- How to create a new tabular model project at the **1400 compatibility level** in Visual Studio with SSDT.
- How to import data from a relational database into a tabular model project workspace database.
- How to create and manage relationships between tables in the model.
- How to create calculated columns, measures, and Key Performance Indicators that help users analyze critical business metrics.
- How to create and manage perspectives and hierarchies that help users more easily browse model data by providing business and application-specific viewpoints.
- How to create partitions that divide table data into smaller logical parts that can be processed independent from other partitions.
- How to secure model objects and data by creating roles with user members.
- How to deploy a tabular model to an **Azure Analysis Services** server or **SQL Server 2017 Analysis Services** server by using SSDT.

Prerequisites

To complete this tutorial, you need:

- An Azure Analysis Services server. Sign up for a free [Azure Analysis Services trial](#) and [create a server](#).
- An [Azure SQL Data Warehouse](#) with the **sample AdventureWorksDW database**, or a SQL Server Data Warehouse with the [Adventure Works sample database](#).

Important: If you install the sample database to an on-premises SQL Server Data Warehouse, and deploy your model to an Azure Analysis Services server, an [On-premises data gateway](#) is required.

- The latest version of [SQL Server Data Tools \(SSDT\)](#) for Visual Studio.
- The latest version of [SQL Server Management Studio \(SSMS\)](#).
- A client application such as [Power BI Desktop](#) or Excel.

Scenario

This tutorial is based on Adventure Works Cycles, a fictitious company. Adventure Works is a large, multinational manufacturing company that produces and distributes bicycles, parts, and accessories to commercial markets in North America, Europe, and Asia. The company employs 500 workers. Additionally, Adventure Works employs

several regional sales teams throughout its market base. Your project is to create a tabular model for sales and marketing users to analyze Internet sales data in the AdventureWorksDW database.

To complete the tutorial, you must complete various lessons. In each lesson, there are tasks. Completing each task in order is necessary for completing the lesson. While in a particular lesson there may be several tasks that accomplish a similar outcome, but how you complete each task is slightly different. This method shows there is often more than one way to complete a task, and to challenge you by using skills you've learned in previous lessons and tasks.

The purpose of the lessons is to guide you through authoring a basic tabular model by using many of the features included in SSDT. Because each lesson builds upon the previous lesson, you should complete the lessons in order.

This tutorial does not provide lessons about managing a server in Azure portal, managing a server or database by using SSMS, or using a client application to browse model data.

Lessons

This tutorial includes the following lessons:

LESSON	ESTIMATED TIME TO COMPLETE
1 - Create a new tabular model project	10 minutes
2 - Get data	10 minutes
3 - Mark as Date Table	3 minutes
4 - Create relationships	10 minutes
5 - Create calculated columns	15 minutes
6 - Create measures	30 minutes
7 - Create Key Performance Indicators (KPI)	15 minutes
8 - Create perspectives	5 minutes
9 - Create hierarchies	20 minutes
10 - Create partitions	15 minutes
11 - Create roles	15 minutes
12 - Analyze in Excel	5 minutes
13 - Deploy	5 minutes

Supplemental lessons

These lessons are not required to complete the tutorial, but can be helpful in better understanding advanced tabular model authoring features.

LESSON	ESTIMATED TIME TO COMPLETE
Detail Rows	10 minutes
Dynamic security	30 minutes
Ragged hierarchies	20 minutes

Next steps

To get started, see [Lesson 1: Create a New Tabular Model Project](#).

Create a tabular model project

4/13/2018 • 3 min to read • [Edit Online](#)

In this lesson, you use Visual Studio with SQL Server Data Tools (SSDT) to create a new tabular model project at the 1400 compatibility level. Once your new project is created, you can begin adding data and authoring your model. This lesson also gives you a brief introduction to the tabular model authoring environment in Visual Studio.

Estimated time to complete this lesson: **10 minutes**

Prerequisites

This topic is the first lesson in a tabular model authoring tutorial. To complete this lesson, there are several prerequisites you need to have in-place. To learn more, see [Azure Analysis Services - Adventure Works tutorial](#).

Create a new tabular model project

To create a new tabular model project

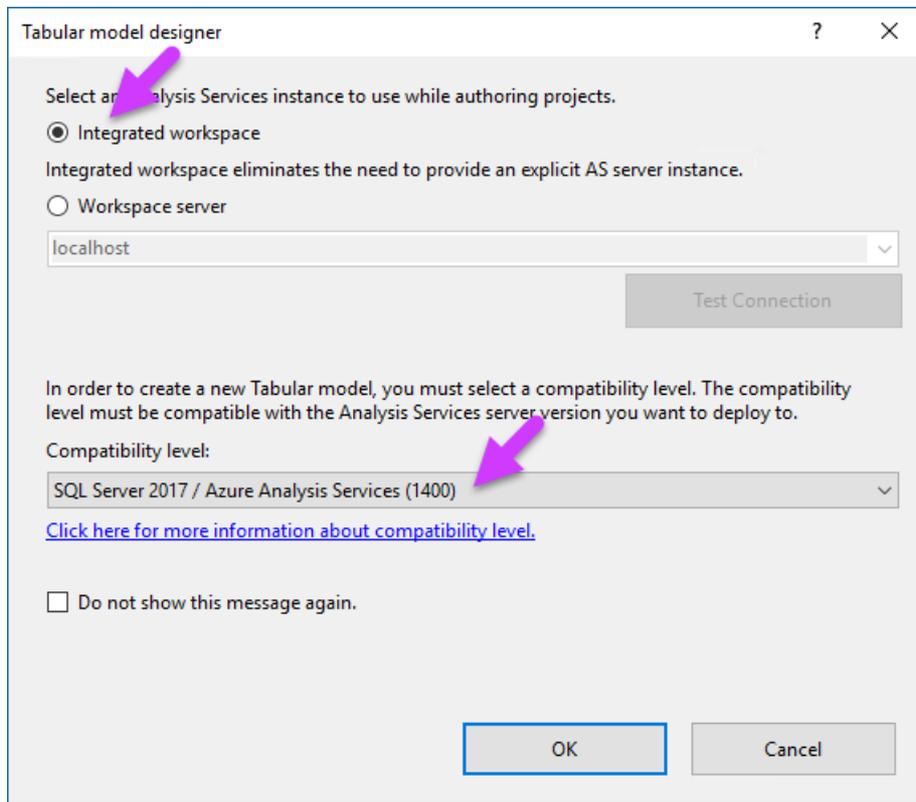
1. In Visual Studio, on the **File** menu, click **New > Project**.
2. In the **New Project** dialog box, expand **Installed > Business Intelligence > Analysis Services**, and then click **Analysis Services Tabular Project**.
3. In **Name**, type **AW Internet Sales**, and then specify a location for the project files.

By default, **Solution Name** is the same as the project name; however, you can type a different solution name.

4. Click **OK**.
5. In the **Tabular model designer** dialog box, select **Integrated workspace**.

The workspace hosts a tabular model database with the same name as the project during model authoring. Integrated workspace means Visual Studio uses a built-in instance, eliminating the need to install a separate Analysis Services server instance just for model authoring.

6. In **Compatibility level**, select **SQL Server 2017 / Azure Analysis Services (1400)**.

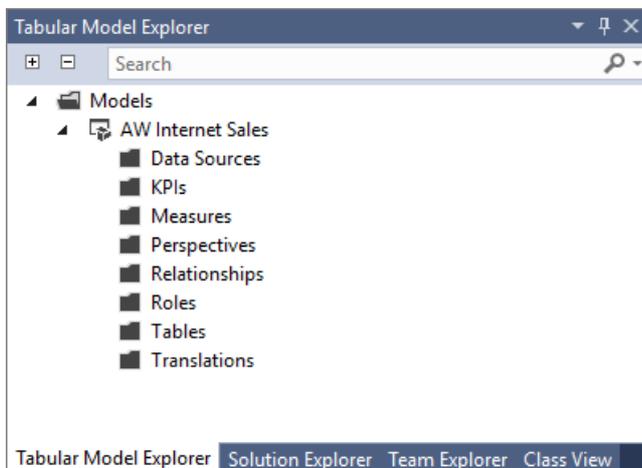


If you don't see SQL Server 2017 / Azure Analysis Services (1400) in the Compatibility level listbox, you're not using the latest version of SQL Server Data Tools. To get the latest version, see [Install SQL Server Data tools](#).

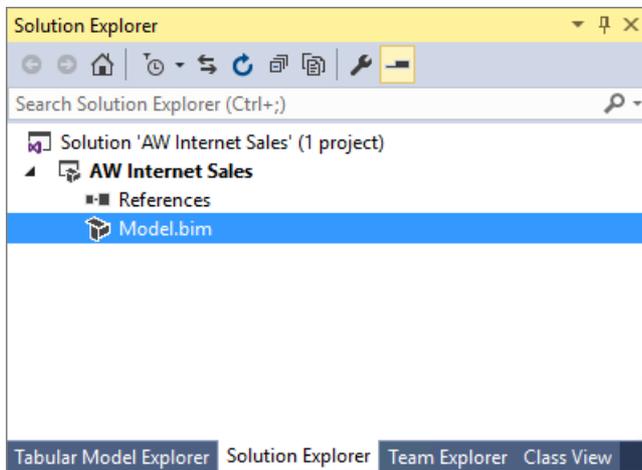
Understanding the SSDT tabular model authoring environment

Now that you've created a new tabular model project, let's take a moment to explore the tabular model authoring environment in Visual Studio.

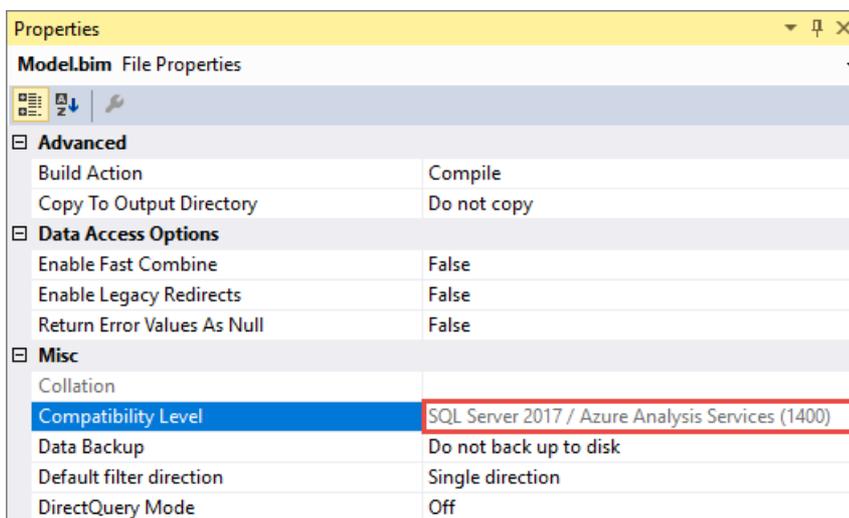
After your project is created, it opens in Visual Studio. On the right side, in **Tabular Model Explorer**, you see a tree view of the objects in your model. Since you haven't yet imported data, the folders are empty. You can right-click an object folder to perform actions, similar to the menu bar. As you step through this tutorial, you use the Tabular Model Explorer to navigate different objects in your model project.



Click the **Solution Explorer** tab. Here, you see your **Model.bim** file. If you don't see the designer window to the left (the empty window with the Model.bim tab), in **Solution Explorer**, under **AW Internet Sales Project**, double-click the **Model.bim** file. The Model.bim file contains the metadata for your model project.



Click **Model.bim**. In the **Properties** window, you see the model properties, most important of which is the **DirectQuery Mode** property. This property specifies if the model is deployed in In-Memory mode (Off) or DirectQuery mode (On). For this tutorial, you author and deploy your model in In-Memory mode.



When you create a model project, certain model properties are set automatically according to the Data Modeling settings that can be specified in the **Tools** menu > **Options** dialog box. Data Backup, Workspace Retention, and Workspace Server properties specify how and where the workspace database (your model authoring database) is backed up, retained in-memory, and built. You can change these settings later if necessary, but for now, leave these properties as they are.

In **Solution Explorer**, right-click **AW Internet Sales** (project), and then click **Properties**. The **AW Internet Sales Property Pages** dialog box appears. You set some of these properties later when you deploy your model.

When you installed SSDT, several new menu items were added to the Visual Studio environment. Click the **Model** menu. From here, you can import data, refresh workspace data, browse your model in Excel, create perspectives and roles, select the model view, and set calculation options. Click the **Table** menu. From here, you can create and manage relationships, specify date table settings, create partitions, and edit table properties. If you click the **Column** menu, you can add and delete columns in a table, freeze columns, and specify sort order. SSDT also adds some buttons to the bar. Most useful is the AutoSum feature to create a standard aggregation measure for a selected column. Other toolbar buttons provide quick access to frequently used features and commands.

Explore some of the dialogs and locations for various features specific to authoring tabular models. While some items are not yet active, you can get a good idea of the tabular model authoring environment.

What's next?

[Lesson 2: Get data.](#)

Get data

4/13/2018 • 3 min to read • [Edit Online](#)

In this lesson, you use Get Data in SSDT to connect to the Adventure Works sample database, select data, preview and filter, and then import into your model workspace.

By using Get Data, you can import data from a wide variety of sources: Azure SQL Database, Oracle, Sybase, OData Feed, Teradata, files and more. Data can also be queried using a Power Query M formula expression.

NOTE

Tasks and images in this tutorial show connecting to an AdventureWorksDW2014 database on an on-premises server. In some cases, an Adventure Works database on Azure may be different.

Estimated time to complete this lesson: **10 minutes**

Prerequisites

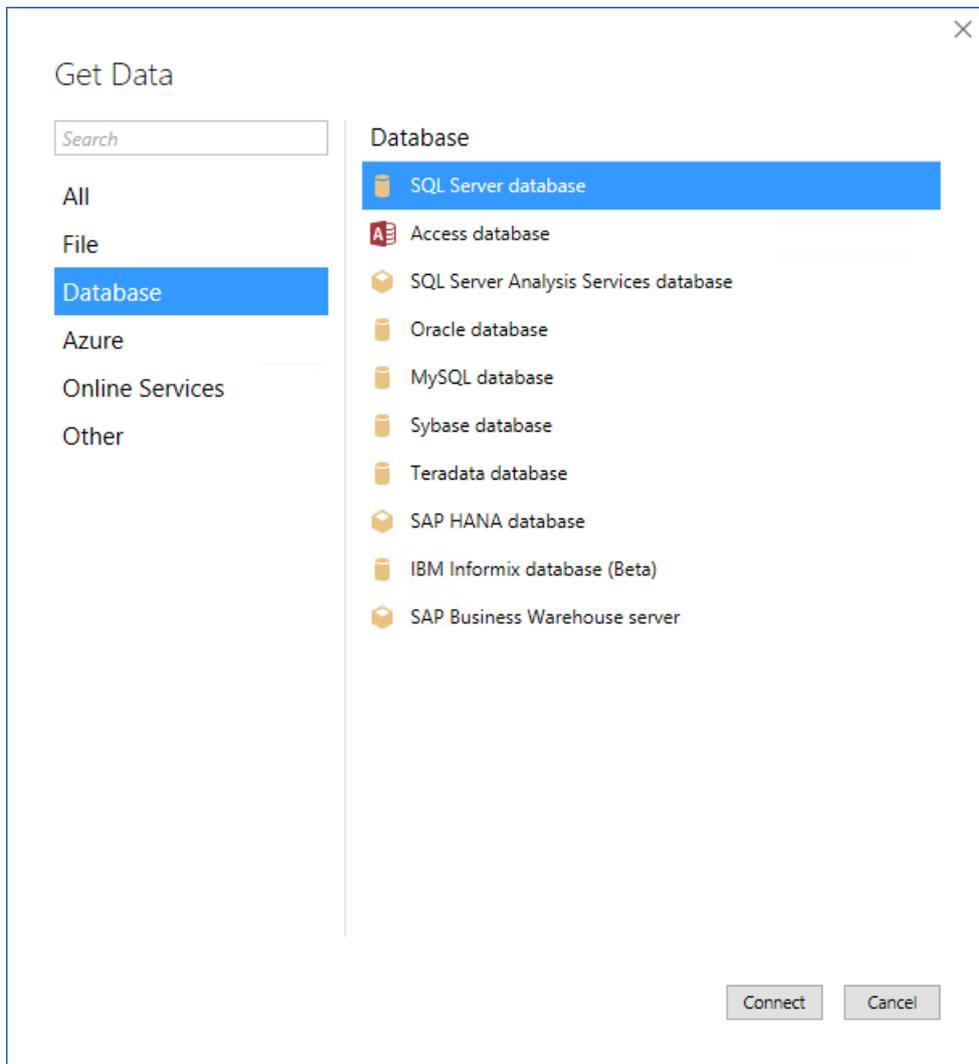
This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 1: Create a new tabular model project](#).

Create a connection

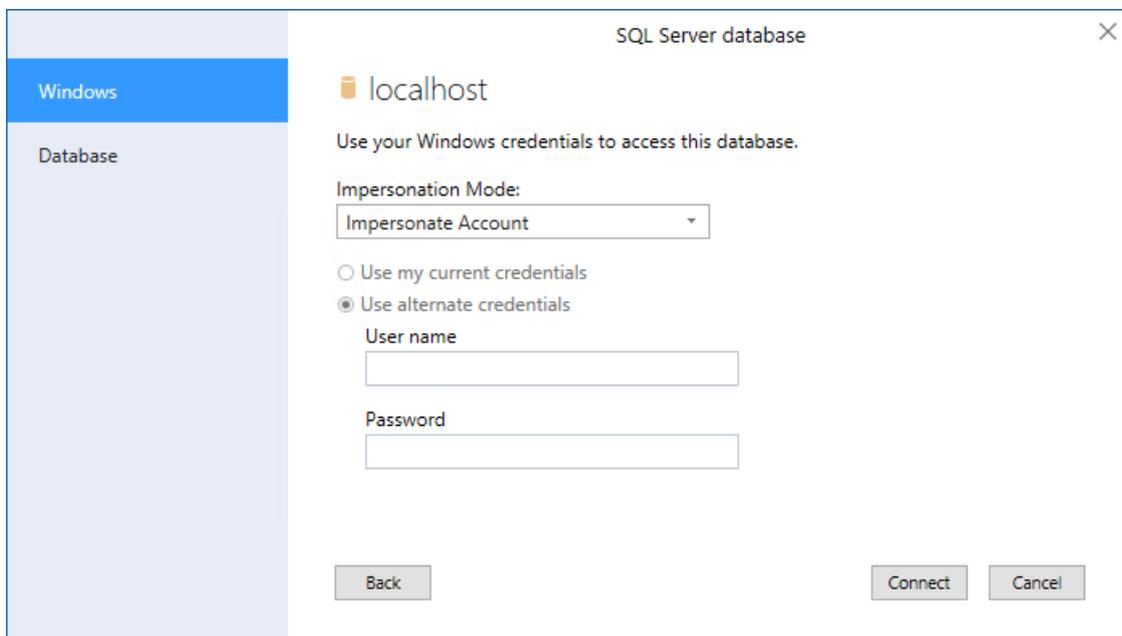
To create a connection to the AdventureWorksDW2014 database

1. In Tabular Model Explorer, right-click **Data Sources** > **Import from Data Source**.

This launches Get Data, which guides you through connecting to a data source. If you don't see Tabular Model Explorer, in **Solution Explorer**, double-click **Model.bim** to open the model in the designer.



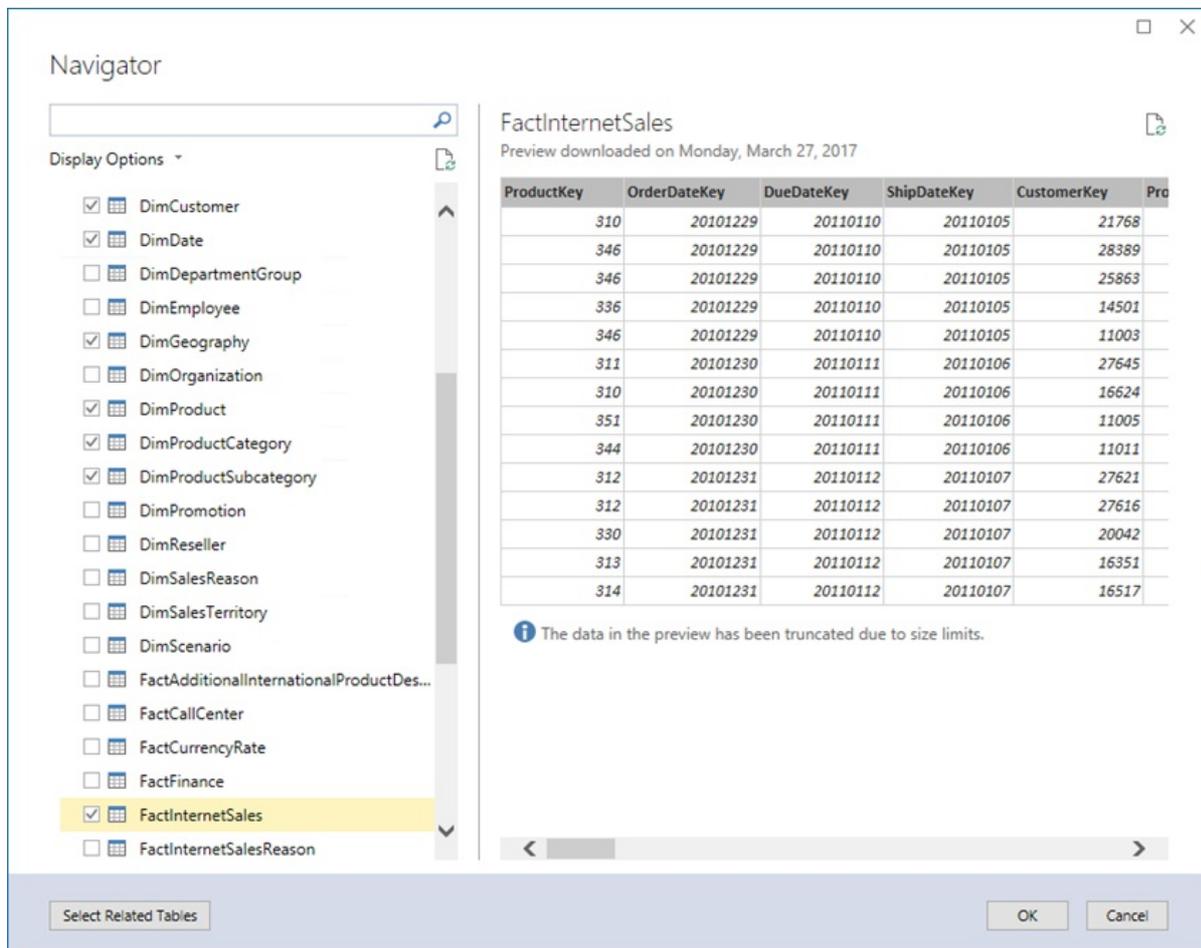
2. In Get Data, click **Database** > **SQL Server Database** > **Connect**.
3. In the **SQL Server Database** dialog, in **Server**, type the name of the server where you installed the AdventureWorksDW2014 database, and then click **Connect**.
4. When prompted to enter credentials, you need to specify the credentials Analysis Services uses to connect to the data source when importing and processing data. In **Impersonation Mode**, select **Impersonate Account**, then enter credentials, and then click **Connect**. It's recommended you use an account where the password doesn't expire.



NOTE

Using a Windows user account and password provides the most secure method of connecting to a data source.

- In Navigator, select the **AdventureWorksDW2014** database, and then click **OK**. This creates the connection to the database.
- In Navigator, select the check box for the following tables: **DimCustomer**, **DimDate**, **DimGeography**, **DimProduct**, **DimProductCategory**, **DimProductSubcategory**, and **FactInternetSales**.



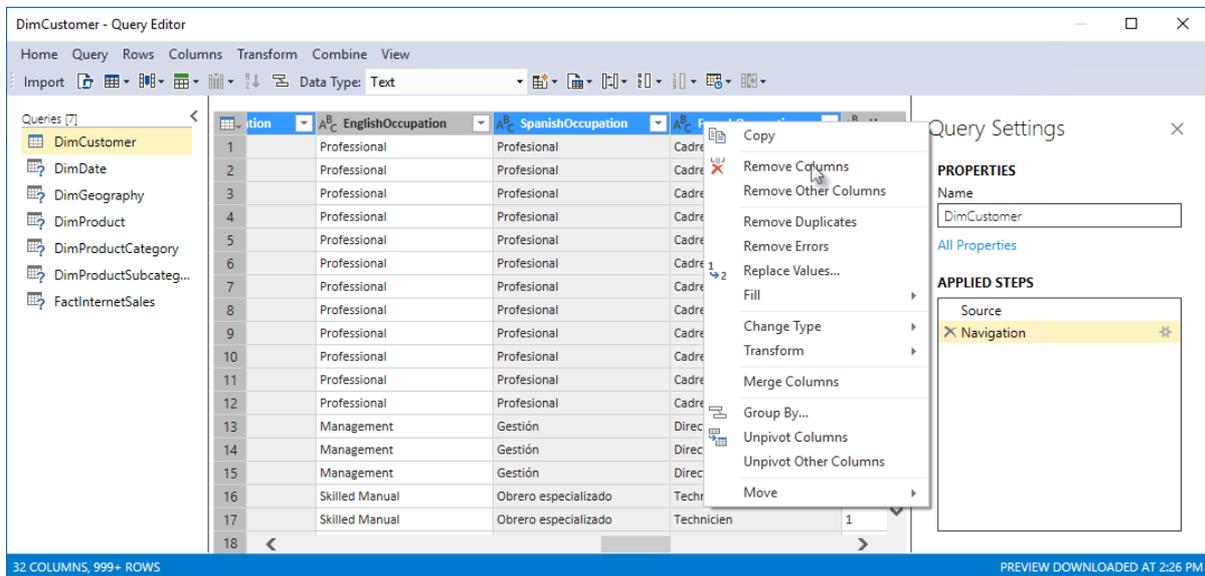
After you click OK, Query Editor opens. In the next section, you select only the data you want to import.

Filter the table data

Tables in the AdventureWorksDW2014 sample database have data that isn't necessary to include in your model. When possible, you want to filter out unnecessary data to save in-memory space used by the model. You filter out some of the columns from tables so they're not imported into the workspace database, or the model database after it has been deployed.

To filter the table data before importing

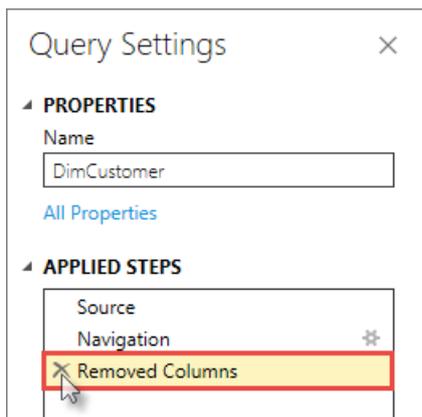
- In Query Editor, select the **DimCustomer** table. A view of the DimCustomer table at the datasource (your AdventureWorksDW2014 sample database) appears.
- Multi-select (Ctrl + click) **SpanishEducation**, **FrenchEducation**, **SpanishOccupation**, **FrenchOccupation**, then right-click, and then click **Remove Columns**.



Since the values for these columns are not relevant to Internet sales analysis, there is no need to import these columns. Eliminating unnecessary columns makes your model smaller and more efficient.

TIP

If you make a mistake, you can backup by deleting a step in **APPLIED STEPS**.



3. Filter the remaining tables by removing the following columns in each table:

DimDate

COLUMN
DateKey
SpanishDayNameOfWeek
FrenchDayNameOfWeek
SpanishMonthName
FrenchMonthName

DimGeography

COLUMN
SpanishCountryRegionName
FrenchCountryRegionName
IpAddressLocator

DimProduct

COLUMN
SpanishProductName
FrenchProductName
FrenchDescription
ChineseDescription
ArabicDescription
HebrewDescription
ThaiDescription
GermanDescription
JapaneseDescription
TurkishDescription

DimProductCategory

COLUMN
SpanishProductCategoryName
FrenchProductCategoryName

DimProductSubcategory

COLUMN
SpanishProductSubcategoryName
FrenchProductSubcategoryName

FactInternetSales

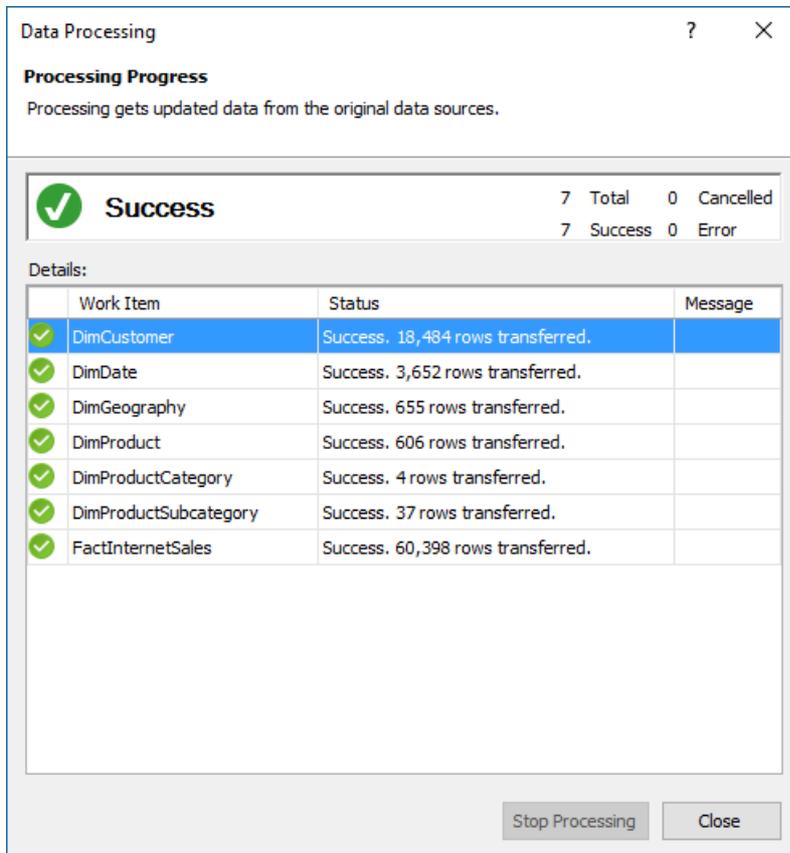
No columns removed.

Import the selected tables and column data

Now that you've previewed and filtered out unnecessary data, you can import the rest of the data you do want. The wizard imports the table data along with any relationships between tables. New tables and columns are created in the model and data that you filtered out is not be imported.

To import the selected tables and column data

1. Review your selections. If everything looks okay, click **Import**. The Data Processing dialog shows the status of data being imported from your datasource into your workspace database.



2. Click **Close**.

Save your model project

It's important to frequently save your model project.

To save the model project

- Click **File > Save All**.

What's next?

[Lesson 3: Mark as Date Table.](#)

Mark as Date Table

4/13/2018 • 1 min to read • [Edit Online](#)

In Lesson 2: Get data, you imported a dimension table named DimDate. While in your model this table is named DimDate, it can also be known as a *Date table*, in that it contains date and time data.

Whenever you use DAX time-intelligence functions, like when you create measures later, you must specify properties which include a *Date table* and a unique identifier *Date column* in that table.

In this lesson, you mark the DimDate table as the *Date table* and the Date column (in the Date table) as the *Date column* (unique identifier).

Before you mark the date table and date column, it's a good time to do a little housekeeping to make your model easier to understand. Notice in the DimDate table a column named **FullDateAlternateKey**. This column contains one row for every day in each calendar year included in the table. You use this column a lot in measure formulas and in reports. But, FullDateAlternateKey isn't really a good identifier for this column. You rename it to **Date**, making it easier to identify and include in formulas. Whenever possible, it's a good idea to rename objects like tables and columns to make them easier to identify in SSDT and client reporting applications like Power BI and Excel.

Estimated time to complete this lesson: **Three minutes**

Prerequisites

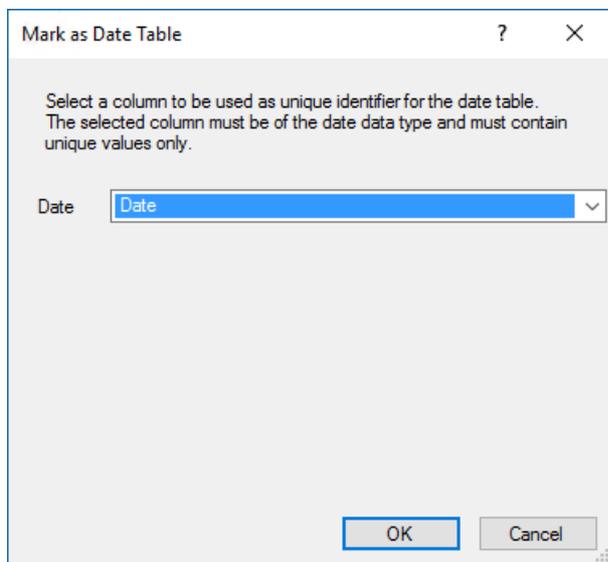
This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 2: Get data](#).

To rename the FullDateAlternateKey column

1. In the model designer, click the **DimDate** table.
2. Double-click the header for the **FullDateAlternateKey** column, and then rename it to **Date**.

To set Mark as Date Table

1. Select the **Date** column, and then in the **Properties** window, under **Data Type**, make sure **Date** is selected.
2. Click the **Table** menu, then click **Date**, and then click **Mark as Date Table**.
3. In the **Mark as Date Table** dialog box, in the **Date** listbox, select the **Date** column as the unique identifier. It's usually selected by default. Click **OK**.



What's next?

[Lesson 4: Create relationships.](#)

Create relationships

4/13/2018 • 4 min to read • [Edit Online](#)

In this lesson, you verify the relationships that were created automatically when you imported data and add new relationships between different tables. A relationship is a connection between two tables that establishes how the data in those tables should be correlated. For example, the DimProduct table and the DimProductSubcategory table have a relationship based on the fact that each product belongs to a subcategory. To learn more, see [Relationships](#).

Estimated time to complete this lesson: **10 minutes**

Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 3: Mark as Date Table](#).

Review existing relationships and add new relationships

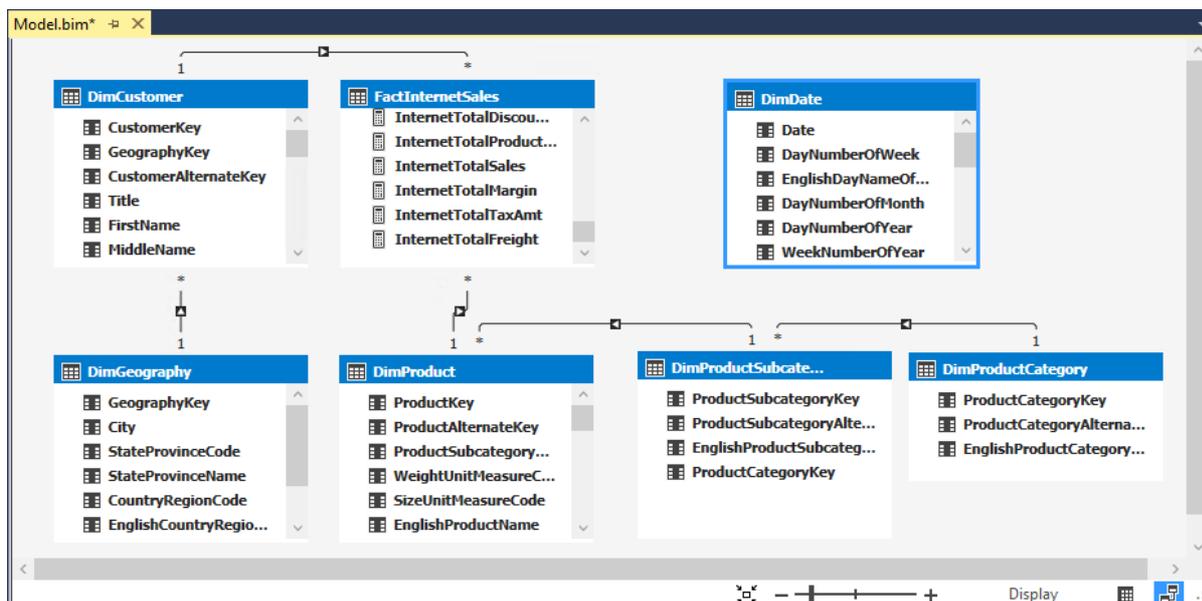
When you imported data by using Get Data, you got seven tables from the AdventureWorksDW2014 database. Generally, when you import data from a relational source, existing relationships are automatically imported together with the data. In order for Get Data to automatically create relationships in the data model, there must be relationships between tables at the data source.

Before you proceed with authoring your model, you should verify those relationships between tables were created properly. For this tutorial, you also add three new relationships.

To review existing relationships

1. Click the **Model** menu > **Model View** > **Diagram View**.

The model designer now appears in Diagram View, a graphical format displaying all the tables you imported with lines between them. The lines between tables indicate the relationships that were automatically created when you imported the data.



NOTE

If you don't see any relationships between tables, it likely means there are no relationships between those tables at the datasource.

Include as many of the tables as possible by using minimap controls in the lower-right corner of the model designer. You can also click and drag tables to different locations, bringing tables closer together, or putting them in a particular order. Moving tables does not affect the relationships between the tables. To view all the columns in a particular table, click and drag on a table edge to expand or make it smaller.

- Click the solid line between the **DimCustomer** table and the **DimGeography** table. The solid line between these two tables shows this relationship is active, that is, it is used by default when calculating DAX formulas.

Notice the **GeographyKey** column in the **DimCustomer** table and the **GeographyKey** column in the **DimGeography** table now both each appear within a box. These columns are used in the relationship. The relationship's properties now also appear in the **Properties** window.

TIP

In addition to using the model designer in diagram view, you can also use the Manage Relationships dialog box to show the relationships between all tables in a table format. In Tabular Model Explorer, right-click **Relationships** > **Manage Relationships**.

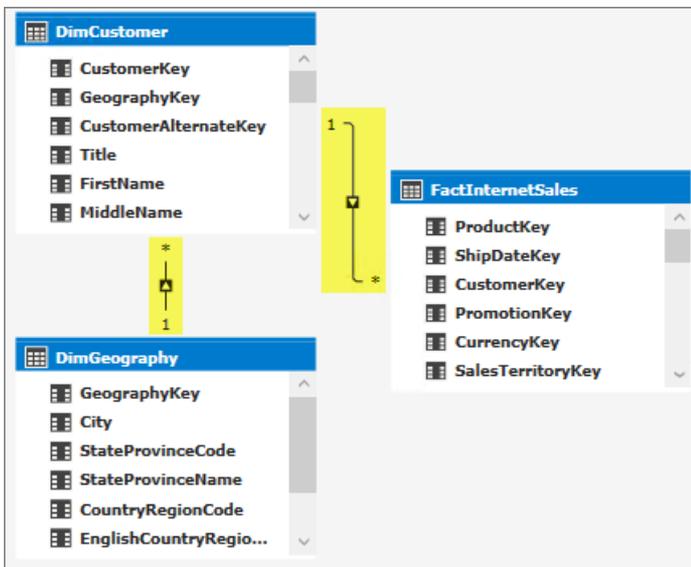
- Verify the following relationships were created when each of the tables were imported from the AdventureWorksDW database:

ACTIVE	TABLE	RELATED LOOKUP TABLE
Yes	DimCustomer [GeographyKey]	DimGeography [GeographyKey]
Yes	DimProduct [ProductSubcategoryKey]	DimProductSubcategory [ProductSubcategoryKey]
Yes	DimProductSubcategory [ProductCategoryKey]	DimProductCategory [ProductCategoryKey]
Yes	FactInternetSales [CustomerKey]	DimCustomer [CustomerKey]
Yes	FactInternetSales [ProductKey]	DimProduct [ProductKey]

If any of the relationships are missing, verify your model includes the following tables: DimCustomer, DimDate, DimGeography, DimProduct, DimProductCategory, DimProductSubcategory, and FactInternetSales. If tables from the same datasource connection are imported at separate times, any relationships between those tables are not be created and must be created manually. If no relationships appear, it means there are no relationships at the datasource. You can create them manually in the data model.

Take a closer look

In Diagram View, notice an arrow, an asterisk, and a number on the lines that show the relationship between tables.



The arrow shows the filter direction. The asterisk shows this table is the many side in the relationship's cardinality, and the one shows this table is the one side of the relationship. If you need to edit a relationship; for example, change the relationship's filter direction or cardinality, double-click the relationship line to open the Edit Relationship dialog.

The 'Edit Relationship' dialog box is shown with the following settings:

- Table 1:** DimCustomer
- Table 2:** DimGeography
- Cardinality:** Many to One (*:1)
- Filter Direction:** << To DimCustomer
- Active:**
- Apply the Filter Direction when using Row Level Security:**

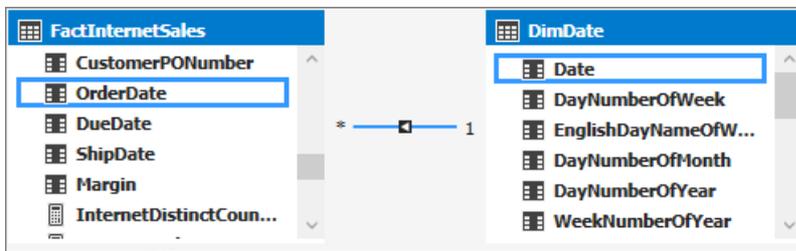
These features are meant for advanced data modeling and are outside the scope of this tutorial. To learn more, see [Bi-directional cross filters for tabular models in Analysis Services](#).

In some cases, you may need to create additional relationships between tables in your model to support certain business logic. For this tutorial, you need to create three additional relationships between the FactInternetSales table and the DimDate table.

To add new relationships between tables

1. In the model designer, in the **FactInternetSales** table, click, and hold on the **OrderDate** column, then drag the cursor to the **Date** column in the **DimDate** table, and then release.

A solid line appears showing you have created an active relationship between the **OrderDate** column in the **Internet Sales** table, and the **Date** column in the **Date** table.



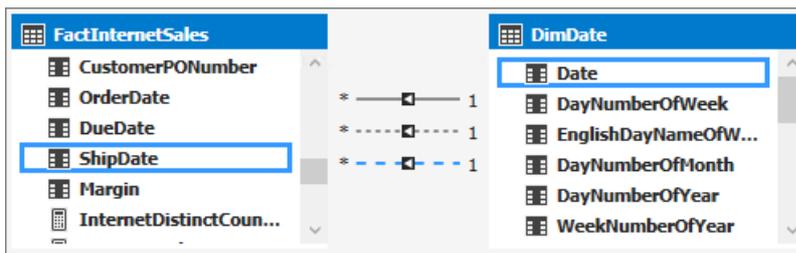
NOTE

When creating relationships, the cardinality and filter direction between the primary table and the related lookup table is automatically selected.

- In the **FactInternetSales** table, click and hold on the **DueDate** column, then drag the cursor to the **Date** column in the **DimDate** table, and then release.

A dotted line appears showing you have created an inactive relationship between the **DueDate** column in the **FactInternetSales** table, and the **Date** column in the **DimDate** table. You can have multiple relationships between tables, but only one relationship can be active at a time. Inactive relationships can be made active to perform special aggregations in custom DAX expressions.

- Finally, create one more relationship. In the **FactInternetSales** table, click and hold on the **ShipDate** column, then drag the cursor to the **Date** column in the **DimDate** table, and then release.



What's next?

[Lesson 5: Create calculated columns.](#)

Create calculated columns

4/13/2018 • 3 min to read • [Edit Online](#)

In this lesson, you create data in your model by adding calculated columns. You can create calculated columns (as custom columns) when using Get Data, by using the Query Editor, or later in the model designer like you do here. To learn more, see [Calculated columns](#).

You create five new calculated columns in three different tables. The steps are slightly different for each task showing there are several ways to create columns, rename them, and place them in various locations in a table.

This lesson is also where you first use Data Analysis Expressions (DAX). DAX is a special language for creating highly customizable formula expressions for tabular models. In this tutorial, you use DAX to create calculated columns, measures, and role filters. To learn more, see [DAX in tabular models](#).

Estimated time to complete this lesson: **15 minutes**

Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 4: Create relationships](#).

Create calculated columns

Create a MonthCalendar calculated column in the DimDate table

1. Click the **Model** menu > **Model View** > **Data View**.

Calculated columns can only be created by using the model designer in Data View.

2. In the model designer, click the **DimDate** table (tab).
3. Right-click the **CalendarQuarter** column header, and then click **Insert Column**.

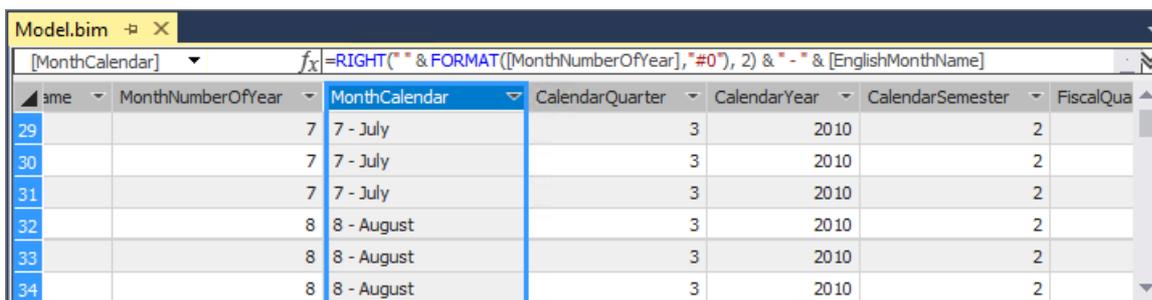
A new column named **Calculated Column 1** is inserted to the left of the **Calendar Quarter** column.

4. In the formula bar above the table, type the following DAX formula: AutoComplete helps you type the fully qualified names of columns and tables, and lists the functions that are available.

```
=RIGHT(" " & FORMAT([MonthNumberOfYear],"#0"), 2) & " - " & [EnglishMonthName]
```

Values are then populated for all the rows in the calculated column. If you scroll down through the table, you see rows can have different values for this column, based on the data in each row.

5. Rename this column to **MonthCalendar**.



name	MonthNumberOfYear	MonthCalendar	CalendarQuarter	CalendarYear	CalendarSemester	FiscalQua
29	7	7 - July	3	2010	2	
30	7	7 - July	3	2010	2	
31	7	7 - July	3	2010	2	
32	8	8 - August	3	2010	2	
33	8	8 - August	3	2010	2	
34	8	8 - August	3	2010	2	

The MonthCalendar calculated column provides a sortable name for Month.

Create a DayOfWeek calculated column in the DimDate table

1. With the **DimDate** table still active, click the **Column** menu, and then click **Add Column**.
2. In the formula bar, type the following formula:

```
=RIGHT(" " & FORMAT([DayNumberOfWeek],"#0"), 2) & " - " & [EnglishDayNameOfWeek]
```

When you've finished building the formula, press ENTER. The new column is added to the far right of the table.

3. Rename the column to **DayOfWeek**.
4. Click the column heading, and then drag the column between the **EnglishDayNameOfWeek** column and the **DayNumberOfMonth** column.

TIP

Moving columns in your table makes it easier to navigate.

The DayOfWeek calculated column provides a sortable name for the day of week.

Create a ProductSubcategoryName calculated column in the DimProduct table

1. In the **DimProduct** table, scroll to the far right of the table. Notice the right-most column is named **Add Column** (italicized), click the column heading.
2. In the formula bar, type the following formula:

```
=RELATED('DimProductSubcategory'[EnglishProductSubcategoryName])
```

3. Rename the column to **ProductSubcategoryName**.

The ProductSubcategoryName calculated column is used to create a hierarchy in the DimProduct table, which includes data from the EnglishProductSubcategoryName column in the DimProductSubcategory table. Hierarchies cannot span more than one table. You create hierarchies later in Lesson 9.

Create a ProductCategoryName calculated column in the DimProduct table

1. With the **DimProduct** table still active, click the **Column** menu, and then click **Add Column**.
2. In the formula bar, type the following formula:

```
=RELATED('DimProductCategory'[EnglishProductCategoryName])
```

3. Rename the column to **ProductCategoryName**.

The ProductCategoryName calculated column is used to create a hierarchy in the DimProduct table, which includes data from the EnglishProductCategoryName column in the DimProductCategory table. Hierarchies cannot span more than one table.

Create a Margin calculated column in the FactInternetSales table

1. In the model designer, select the **FactInternetSales** table.
2. Create a new calculated column between the **SalesAmount** column and the **TaxAmt** column.
3. In the formula bar, type the following formula:

=[SalesAmount]-[TotalProductCost]

4. Rename the column to **Margin**.

	ProductCost	SalesAmount	Margin	TaxAmt	Freight	CarrierTrackingNumber	CustomerPONum
25553	\$41.57	\$53.99	\$12.42	\$4.32	\$1.35		
25554	\$2.97	\$7.95	\$4.98	\$0.64	\$0.20		
25555	\$20.57	\$54.99	\$34.42	\$4.40	\$1.37		
25556	\$9.16	\$24.49	\$15.33	\$1.96	\$0.61		
25557	\$44.88	\$120.00	\$75.12	\$9.60	\$3.00		
25558	\$1.87	\$4.99	\$3.12	\$0.40	\$0.12		
25559	\$1.49	\$3.99	\$2.50	\$0.32	\$0.10		

The Margin calculated column is used to analyze profit margins for each sale.

What's next?

[Lesson 6: Create measures.](#)

Create measures

4/13/2018 • 3 min to read • [Edit Online](#)

In this lesson, you create measures to be included in your model. Similar to the calculated columns you created, a measure is a calculation created by using a DAX formula. However, unlike calculated columns, measures are evaluated based on a user selected *filter*. For example, a particular column or slicer added to the Row Labels field in a PivotTable. A value for each cell in the filter is then calculated by the applied measure. Measures are powerful, flexible calculations that you want to include in almost all tabular models to perform dynamic calculations on numerical data. To learn more, see [Measures](#).

To create measures, you use the *Measure Grid*. By default, each table has an empty measure grid; however, you typically do not create measures for every table. The measure grid appears below a table in the model designer when in Data View. To hide or show the measure grid for a table, click the **Table** menu, and then click **Show Measure Grid**.

You can create a measure by clicking an empty cell in the measure grid, and then typing a DAX formula in the formula bar. When you click ENTER to complete the formula, the measure then appears in the cell. You can also create measures using a standard aggregation function by clicking a column, and then clicking the AutoSum button (Σ) on the toolbar. Measures created using the AutoSum feature appear in the measure grid cell directly beneath the column, but can be moved.

In this lesson, you create measures by both entering a DAX formula in the formula bar, and by using the AutoSum feature.

Estimated time to complete this lesson: **30 minutes**

Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 5: Create calculated columns](#).

Create measures

To create a DaysCurrentQuarterToDate measure in the DimDate table

1. In the model designer, click the **DimDate** table.
2. In the measure grid, click the top-left empty cell.
3. In the formula bar, type the following formula:

```
DaysCurrentQuarterToDate:=COUNTROWS( DATESQTD( 'DimDate'[Date]))
```

Notice the top-left cell now contains a measure name, **DaysCurrentQuarterToDate**, followed by the result, **92**. The result is not relevant at this point because no user filter has been applied.

Date	Day...	EnglishDayNameOfWeek	DayOfWeek	DayNumberOfMon
7/1/2010 12:00:00 AM	5	Thursday	5 - Thursday	
7/2/2010 12:00:00 AM	6	Friday	6 - Friday	
7/3/2010 12:00:00 AM	7	Saturday	7 - Saturday	
7/4/2010 12:00:00 AM	1	Sunday	1 - Sunday	
7/5/2010 12:00:00 AM	2	Monday	2 - Monday	
7/6/2010 12:00:00 AM	3	Tuesday	3 - Tuesday	

DaysCurrentQuarterToDate: 92

Unlike calculated columns, with measure formulas you can type the measure name, followed by a colon, followed by the formula expression.

To create a DaysInCurrentQuarter measure in the DimDate table

1. With the **DimDate** table still active in the model designer, in the measure grid, click the empty cell below the measure you created.
2. In the formula bar, type the following formula:

```
DaysInCurrentQuarter:=COUNTROWS( DATESBETWEEN( 'DimDate'[Date], STARTOFQUARTER(
LASTDATE('DimDate'[Date])), ENDOFQUARTER('DimDate'[Date])))
```

When creating a comparison ratio between one incomplete period and the previous period. The formula must calculate the proportion of the period that has elapsed and compare it to the same proportion in the previous period. In this case, [DaysCurrentQuarterToDate]/[DaysInCurrentQuarter] gives the proportion elapsed in the current period.

To create an InternetDistinctCountSalesOrder measure in the FactInternetSales table

1. Click the **FactInternetSales** table.
2. Click the **SalesOrderNumber** column heading.
3. On the toolbar, click the down-arrow next to the AutoSum (Σ) button, and then select **DistinctCount**.

The AutoSum feature automatically creates a measure for the selected column using the DistinctCount standard aggregation formula.

AgencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	RevisionNumber	OrderNumber
1	100	4 SO51900		1	1
2	100	4 SO51948		1	1
3	100	4 SO52043		1	1
4	100	4 SO52045		1	1
5	100	4 SO52094		1	1
6	100	4 SO52175		1	1

Distinct Count SalesOrderNumber: 27659

4. In the measure grid, click the new measure, and then in the **Properties** window, in **Measure Name**, rename the measure to **InternetDistinctCountSalesOrder**.

To create additional measures in the FactInternetSales table

1. By using the AutoSum feature, create and name the following measures:

COLUMN	MEASURE NAME	AUTOSUM (Σ)	FORMULA
SalesOrderLineNumber	InternetOrderLinesCount	Count	=COUNTA([SalesOrderLine Number])

COLUMN	MEASURE NAME	AUTOSUM (Σ)	FORMULA
OrderQuantity	InternetTotalUnits	Sum	=SUM([OrderQuantity])
DiscountAmount	InternetTotalDiscountAmount	Sum	=SUM([DiscountAmount])
TotalProductCost	InternetTotalProductCost	Sum	=SUM([TotalProductCost])
SalesAmount	InternetTotalSales	Sum	=SUM([SalesAmount])
Margin	InternetTotalMargin	Sum	=SUM([Margin])
TaxAmt	InternetTotalTaxAmt	Sum	=SUM([TaxAmt])
Freight	InternetTotalFreight	Sum	=SUM([Freight])

2. By clicking an empty cell in the measure grid, and by using the formula bar, create, the following custom measures in order:

```
InternetPreviousQuarterMargin:=CALCULATE([InternetTotalMargin],PREVIOUSQUARTER('DimDate'[Date]))
```

```
InternetCurrentQuarterMargin:=TOTALQTD([InternetTotalMargin],'DimDate'[Date])
```

```
InternetPreviousQuarterMarginProportionToQTD:=[InternetPreviousQuarterMargin]*
([DaysCurrentQuarterToDate]/[DaysInCurrentQuarter])
```

```
InternetPreviousQuarterSales:=CALCULATE([InternetTotalSales],PREVIOUSQUARTER('DimDate'[Date]))
```

```
InternetCurrentQuarterSales:=TOTALQTD([InternetTotalSales],'DimDate'[Date])
```

```
InternetPreviousQuarterSalesProportionToQTD:=[InternetPreviousQuarterSales]*
([DaysCurrentQuarterToDate]/[DaysInCurrentQuarter])
```

Measures created for the FactInternetSales table can be used to analyze critical financial data such as sales, costs, and profit margin for items defined by the user selected filter.

What's next?

[Lesson 7: Create Key Performance Indicators.](#)

Create Key Performance Indicators

4/13/2018 • 1 min to read • [Edit Online](#)

In this lesson, you create Key Performance Indicators (KPIs). KPIs are used to gauge performance of a value defined by a *Base* measure, against a *Target* value also defined by a measure, or by an absolute value. In reporting client applications, KPIs can provide business professionals a quick and easy way to understand a summary of business success or to identify trends. To learn more, see [KPIs](#)

Estimated time to complete this lesson: **15 minutes**

Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 6: Create measures](#).

Create Key Performance Indicators

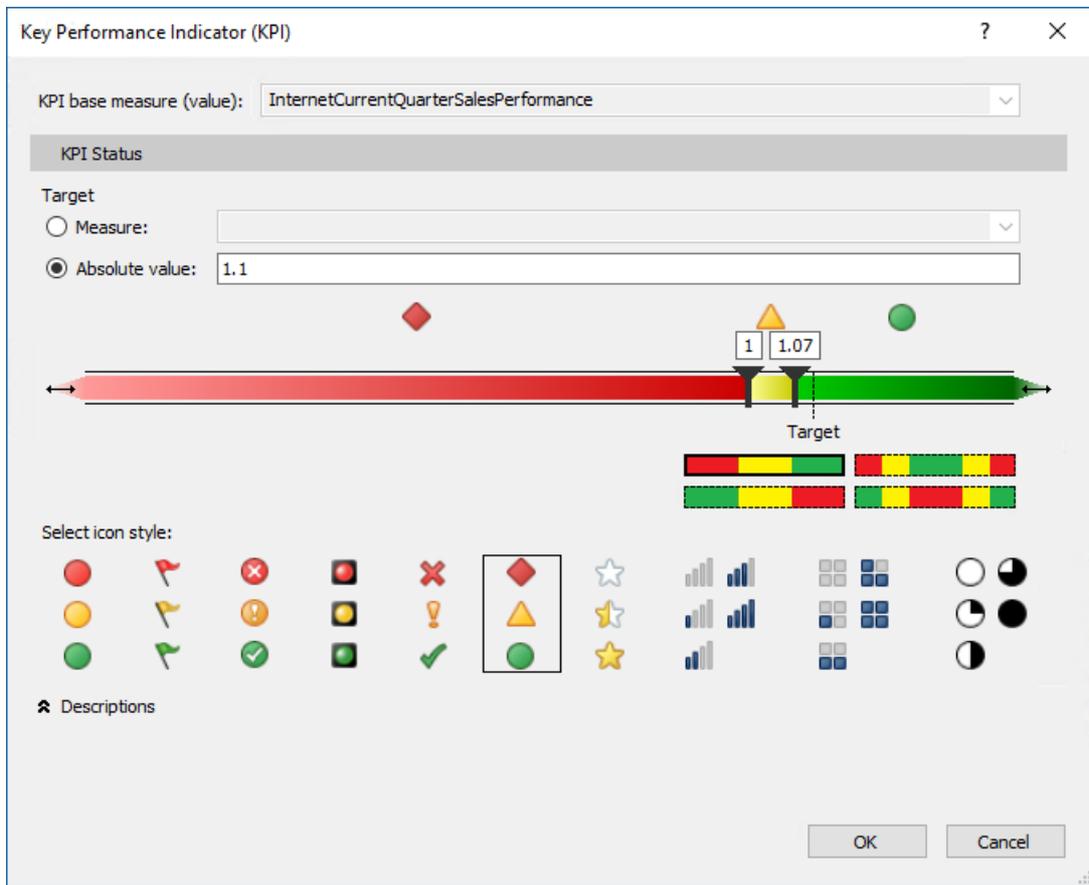
To create an **InternetCurrentQuarterSalesPerformance** KPI

1. In the model designer, click the **FactInternetSales** table.
2. In the measure grid, click an empty cell.
3. In the formula bar, above the table, type the following formula:

```
InternetCurrentQuarterSalesPerformance  
:=DIVIDE([InternetCurrentQuarterSales]/[InternetPreviousQuarterSalesProportionToQTD],BLANK())
```

This measure serves as the Base measure for the KPI.

4. In the measure grid, right-click **InternetCurrentQuarterSalesPerformance** > **Create KPI**.
5. In the Key Performance Indicator (KPI) dialog box, in **Target** select **Absolute Value**, and then type **1.1**.
6. In the left (low) slider field, type **1**, and then in the right (high) slider field, type **1.07**.
7. In **Select Icon Style**, select the diamond (red), triangle (yellow), circle (green) icon type.



TIP

Notice the expandable **Descriptions** label below the available icon styles. Use descriptions for the various KPI elements to make them more identifiable in client applications.

- Click **OK** to complete the KPI.

In the measure grid, notice the icon next to the **InternetCurrentQuarterSalesPerformance** measure. This icon indicates that this measure serves as a Base value for a KPI.

To create an **InternetCurrentQuarterMarginPerformance** KPI

- In the measure grid for the **FactInternetSales** table, click an empty cell.
- In the formula bar, above the table, type the following formula:

```
InternetCurrentQuarterMarginPerformance :=IF([InternetPreviousQuarterMarginProportionToQTD]<>0,
([InternetCurrentQuarterMargin]-
[InternetPreviousQuarterMarginProportionToQTD])/[InternetPreviousQuarterMarginProportionToQTD],BLANK())
```

- Right-click **InternetCurrentQuarterMarginPerformance** > **Create KPI**.
- In the Key Performance Indicator (KPI) dialog box, in **Target** select **Absolute Value**, and then type **1.25**.
- In the left (low) slider field, slide until the field displays **0.8**, and then slide the right (high) slider field, until the field displays **1.03**.
- In **Select Icon Style**, select the diamond (red), triangle (yellow), circle (green) icon type, and then click **OK**.

What's next?

[Lesson 8: Create perspectives.](#)

Create perspectives

4/13/2018 • 1 min to read • [Edit Online](#)

In this lesson, you create an Internet Sales perspective. A perspective defines a viewable subset of a model that provides focused, business-specific, or application-specific viewpoints. When a user connects to a model by using a perspective, they see only those model objects (tables, columns, measures, hierarchies, and KPIs) as fields defined in that perspective. To learn more, see [Perspectives](#).

The Internet Sales perspective you create in this lesson excludes the DimCustomer table object. When you create a perspective that excludes certain objects from view, that object still exists in the model. However, it is not visible in a reporting client field list. Calculated columns and measures either included in a perspective or not can still calculate from object data that is excluded.

The purpose of this lesson is to describe how to create perspectives and become familiar with the tabular model authoring tools. If you later expand this model to include additional tables, you can create additional perspectives to define different viewpoints of the model, for example, Inventory and Sales.

Estimated time to complete this lesson: **Five minutes**

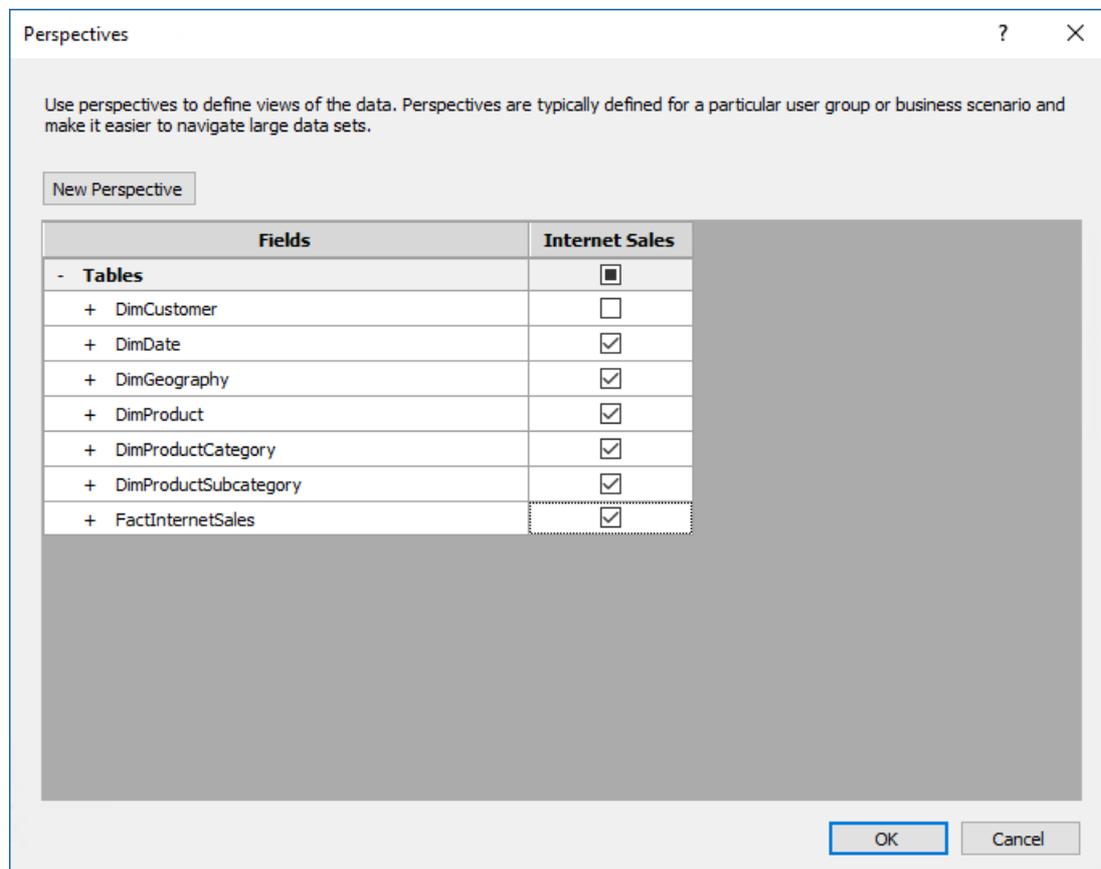
Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 7: Create Key Performance Indicators](#).

Create perspectives

To create an Internet Sales perspective

1. Click the **Model** menu > **Perspectives** > **Create and Manage**.
2. In the **Perspectives** dialog box, click **New Perspective**.
3. Double-click the **New Perspective** column heading, and then rename **Internet Sales**.
4. Select the all the tables *except* **DimCustomer**.



In a later lesson, you use the Analyze in Excel feature to test this perspective. The Excel PivotTable Fields List includes each table except the DimCustomer table.

What's next?

[Lesson 9: Create hierarchies.](#)

Create hierarchies

4/13/2018 • 1 min to read • [Edit Online](#)

In this lesson, you create hierarchies. Hierarchies are groups of columns arranged in levels. For example, a Geography hierarchy might have sublevels for Country, State, County, and City. Hierarchies can appear separate from other columns in a reporting client application field list, making them easier for client users to navigate and include in a report. To learn more, see [Hierarchies](#)

To create hierarchies, use the model designer in *Diagram View*. Creating and managing hierarchies is not supported in Data View.

Estimated time to complete this lesson: **20 minutes**

Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 8: Create perspectives](#).

Create hierarchies

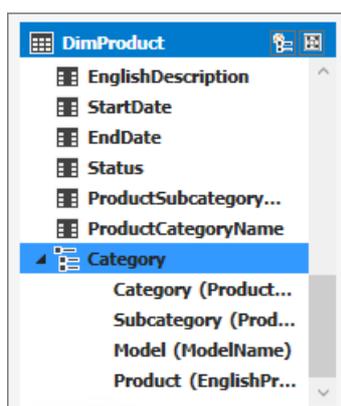
To create a **Category** hierarchy in the **DimProduct** table

1. In the model designer (diagram view), right-click the **DimProduct** table > **Create Hierarchy**. A new hierarchy appears at the bottom of the table window. Rename the hierarchy **Category**.
2. Click and drag the **ProductCategoryName** column to the new **Category** hierarchy.
3. In the **Category** hierarchy, right-click **ProductCategoryName** > **Rename**, and then type **Category**.

NOTE

Renaming a column in a hierarchy does not rename that column in the table. A column in a hierarchy is just a representation of the column in the table.

4. Click and drag the **ProductSubcategoryName** column to the **Category** hierarchy. Rename it **Subcategory**.
5. Right-click the **ModelName** column > **Add to hierarchy**, and then select **Category**. Rename it **Model**.
6. Finally, add **EnglishProductName** to the **Category** hierarchy. Rename it **Product**.



To create hierarchies in the **DimDate** table

1. In the **DimDate** table, create a hierarchy named **Calendar**.
2. Add the following columns in-order:
 - CalendarYear
 - CalendarSemester
 - CalendarQuarter
 - MonthCalendar
 - DayNumberOfMonth
3. In the **DimDate** table, create a **Fiscal** hierarchy. Include the following columns in-order:
 - FiscalYear
 - FiscalSemester
 - FiscalQuarter
 - MonthCalendar
 - DayNumberOfMonth
4. Finally, in the **DimDate** table, create a **ProductionCalendar** hierarchy. Include the following columns in-order:
 - CalendarYear
 - WeekNumberOfYear
 - DayNumberOfWeek

What's next?

[Lesson 10: Create partitions.](#)

Create partitions

4/13/2018 • 3 min to read • [Edit Online](#)

In this lesson, you create partitions to divide the FactInternetSales table into smaller logical parts that can be processed (refreshed) independent of other partitions. By default, every table you include in your model has one partition, which includes all the table's columns and rows. For the FactInternetSales table, we want to divide the data by year; one partition for each of the table's five years. Each partition can then be processed independently. To learn more, see [Partitions](#).

Estimated time to complete this lesson: **15 minutes**

Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 9: Create Hierarchies](#).

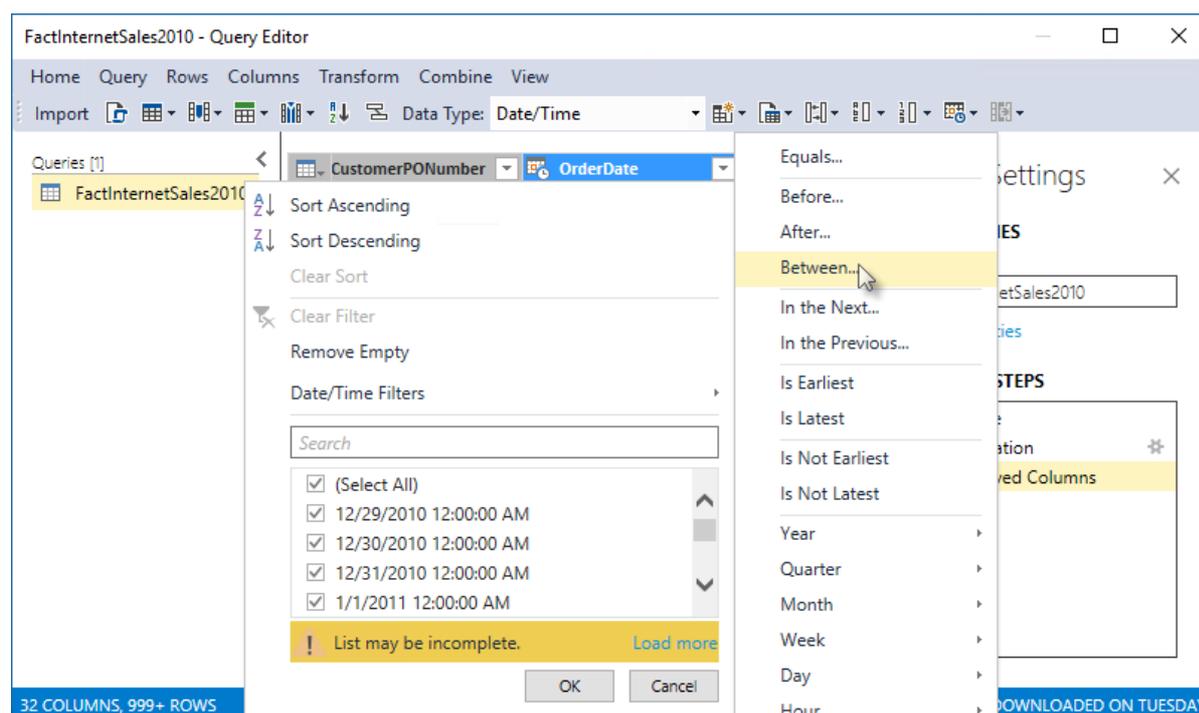
Create partitions

To create partitions in the FactInternetSales table

1. In Tabular Model Explorer, expand **Tables**, and then right-click **FactInternetSales** > **Partitions**.
2. In Partition Manager, click **Copy**, and then change the name to **FactInternetSales2010**.

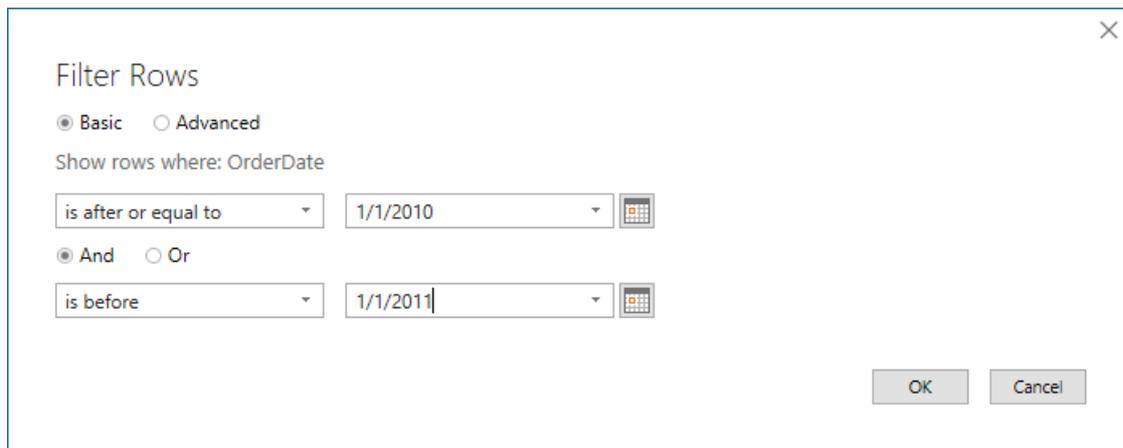
Because you want the partition to include only those rows within a certain period, for the year 2010, you must modify the query expression.

3. Click **Design** to open Query Editor, and then click the **FactInternetSales2010** query.
4. In preview, click the down arrow in the **OrderDate** column heading, and then click **Date/Time Filters** > **Between**.



5. In the Filter Rows dialog box, in **Show rows where: OrderDate**, leave **is after or equal to**, and then in the date field, enter **1/1/2010**. Leave the **And** operator selected, then select **is before**, then in the date field,

enter **1/1/2011**, and then click **OK**.



Notice in Query Editor, in APPLIED STEPS, you see another step named Filtered Rows. This filter is to select only order dates from 2010.

6. Click **Import**.

In Partition Manager, notice the query expression now has an additional Filtered Rows clause.

```
let
  Source = #"SQL/owend_dt1;AdventureWorksDW2014",
  dbo_FactInternetSales = Source{[Schema="dbo",Item="FactInternetSales"]}[Data],
  #Filtered Rows = Table.SelectRows(dbo_FactInternetSales, each [OrderDate] >= #datetime(2010, 1, 1, 0, 0, 0) and
  [OrderDate] < #datetime(2011, 1, 1, 0, 0, 0))
in
  #Filtered Rows
```

This statement specifies this partition should include only the data in those rows where the OrderDate is in the 2010 calendar year as specified in the filtered rows clause.

To create a partition for the 2011 year

1. In the partitions list, click the **FactInternetSales2010** partition you created, and then click **Copy**. Change the partition name to **FactInternetSales2011**.

You do not need to use Query Editor to create a new filtered rows clause. Because you created a copy of the query for 2010, all you need to do is make a slight change in the query for 2011.

2. In **Query Expression**, in-order for this partition to include only those rows for the 2011 year, replace the years in the Filtered Rows clause with **2011** and **2012**, respectively, like:

```
let
  Source = #"SQL/localhost;AdventureWorksDW2014",
  dbo_FactInternetSales = Source{[Schema="dbo",Item="FactInternetSales"]}[Data],
  #Removed Columns = Table.RemoveColumns(dbo_FactInternetSales, {"OrderDateKey", "DueDateKey",
  "ShipDateKey"}),
  #Filtered Rows = Table.SelectRows(#Removed Columns, each [OrderDate] >= #datetime(2011, 1, 1, 0,
  0, 0) and [OrderDate] < #datetime(2012, 1, 1, 0, 0, 0))
in
  #Filtered Rows
```

To create partitions for 2012, 2013, and 2014.

- Follow the previous steps, creating partitions for 2012, 2013, and 2014, changing the years in the Filtered Rows clause to include only rows for that year.

Delete the FactInternetSales partition

Now that you have partitions for each year, you can delete the FactInternetSales partition; preventing overlap when choosing Process all when processing partitions.

To delete the FactInternetSales partition

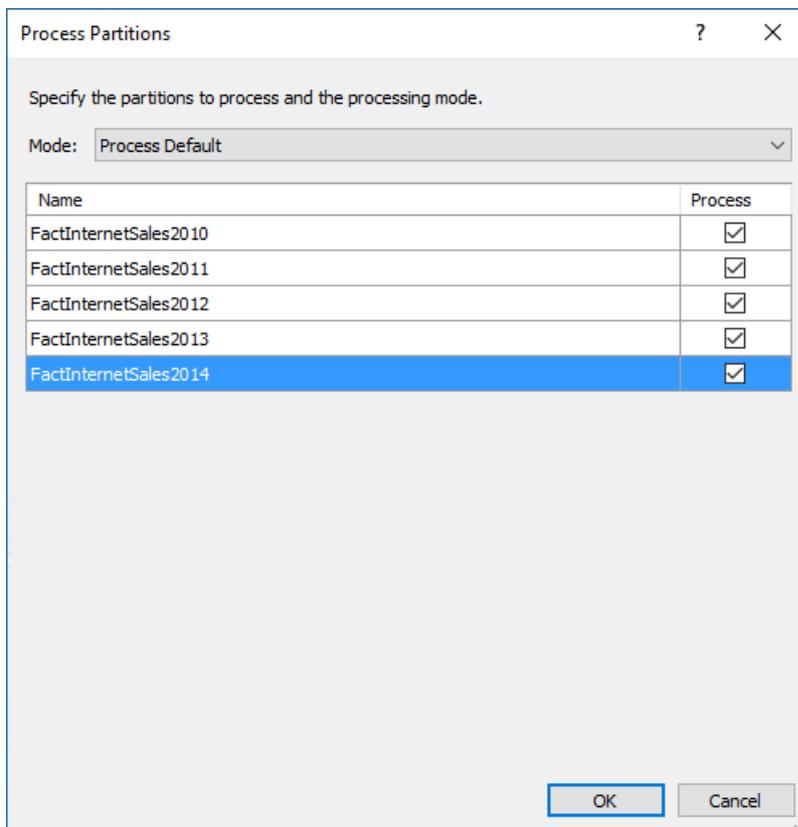
- Click the FactInternetSales partition, and then click **Delete**.

Process partitions

In Partition Manager, notice the **Last Processed** column for each of the new partitions you created shows these partitions have never been processed. When you create partitions, you should run a Process Partitions or Process Table operation to refresh the data in those partitions.

To process the FactInternetSales partitions

1. Click **OK** to close Partition Manager.
2. Click the **FactInternetSales** table, then click the **Model** menu > **Process** > **Process Partitions**.
3. In the Process Partitions dialog box, verify **Mode** is set to **Process Default**.
4. Select the checkbox in the **Process** column for each of the five partitions you created, and then click **OK**.



If you're prompted for Impersonation credentials, enter the Windows user name and password you specified in Lesson 2.

The **Data Processing** dialog box appears and displays process details for each partition. Notice that a different number of rows for each partition are transferred. Each partition includes only those rows for the year specified in the WHERE clause in the SQL Statement. When processing is finished, go ahead and close the Data Processing dialog box.

Data Processing ? X

Processing Progress
Processing gets updated data from the original data sources.

 **Success** 5 Total 0 Cancelled
5 Success 0 Error

Details:

Work Item	Status	Message
 FactInternetSales2010	Success. 14 rows transferred.	
 FactInternetSales2011	Success. 2,216 rows transferred.	
 FactInternetSales2012	Success. 3,397 rows transferred.	
 FactInternetSales2013	Success. 52,801 rows transferred.	
 FactInternetSales2014	Success. 1,970 rows transferred.	

Stop Processing Close

What's next?

Go to the next lesson: [Lesson 11: Create Roles](#).

Create roles

4/13/2018 • 2 min to read • [Edit Online](#)

In this lesson, you create roles. Roles provide model database object and data security by limiting access to only those users that are role members. Each role is defined with a single permission: None, Read, Read and Process, Process, or Administrator. Roles can be defined during model authoring by using Role Manager. After a model has been deployed, you can manage roles by using SQL Server Management Studio (SSMS). To learn more, see [Roles](#).

NOTE

Creating roles is not necessary to complete this tutorial. By default, the account you are currently logged in with has Administrator privileges on the model. However, for other users in your organization to browse by using a reporting client, you must create at least one role with Read permissions and add those users as members.

You create three roles:

- **Sales Manager** – This role can include users in your organization for which you want to have Read permission to all model objects and data.
- **Sales Analyst US** – This role can include users in your organization for which you want only to be able to browse data related to sales in the United States. For this role, you use a DAX formula to define a *Row Filter*, which restricts members to browse data only for the United States.
- **Administrator** – This role can include users for which you want to have Administrator permission, which allows unlimited access and permissions to perform administrative tasks on the model database.

Because Windows user and group accounts in your organization are unique, you can add accounts from your particular organization to members. However, for this tutorial, you can also leave the members blank. You test the effect of each role later in Lesson 12: Analyze in Excel.

Estimated time to complete this lesson: **15 minutes**

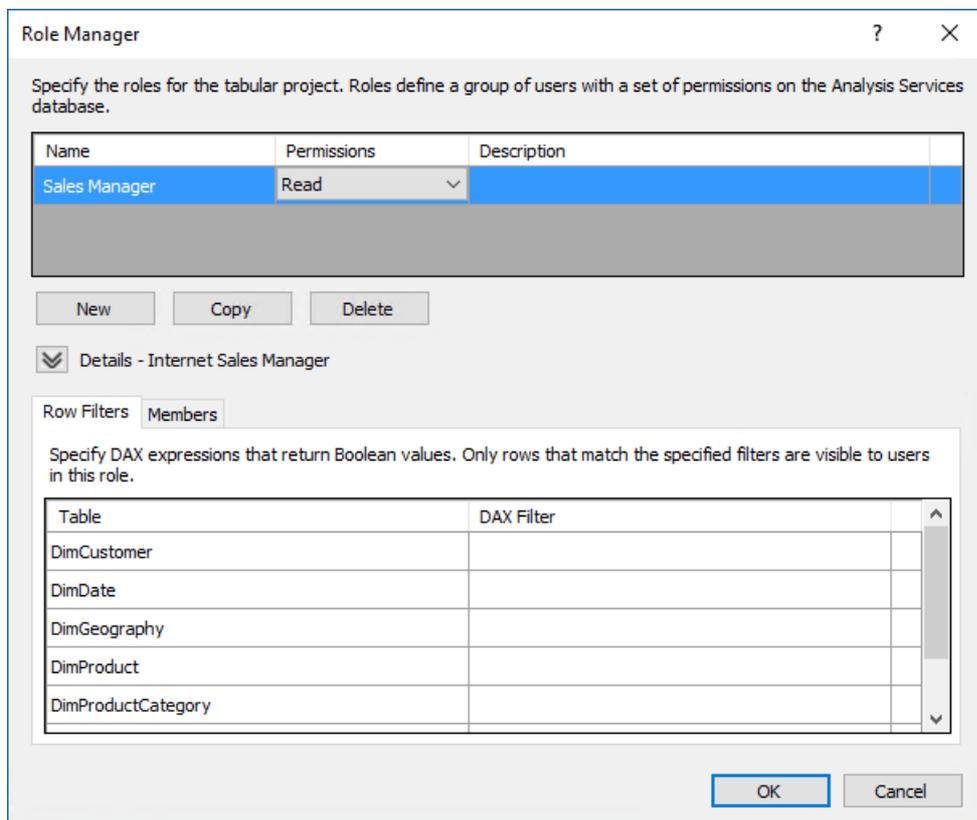
Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 10: Create partitions](#).

Create roles

To create a Sales Manager user role

1. In Tabular Model Explorer, right-click **Roles** > **Roles**.
2. In Role Manager, click **New**.
3. Click the new role, and then in the **Name** column, rename the role to **Sales Manager**.
4. In the **Permissions** column, click the dropdown list, and then select the **Read** permission.



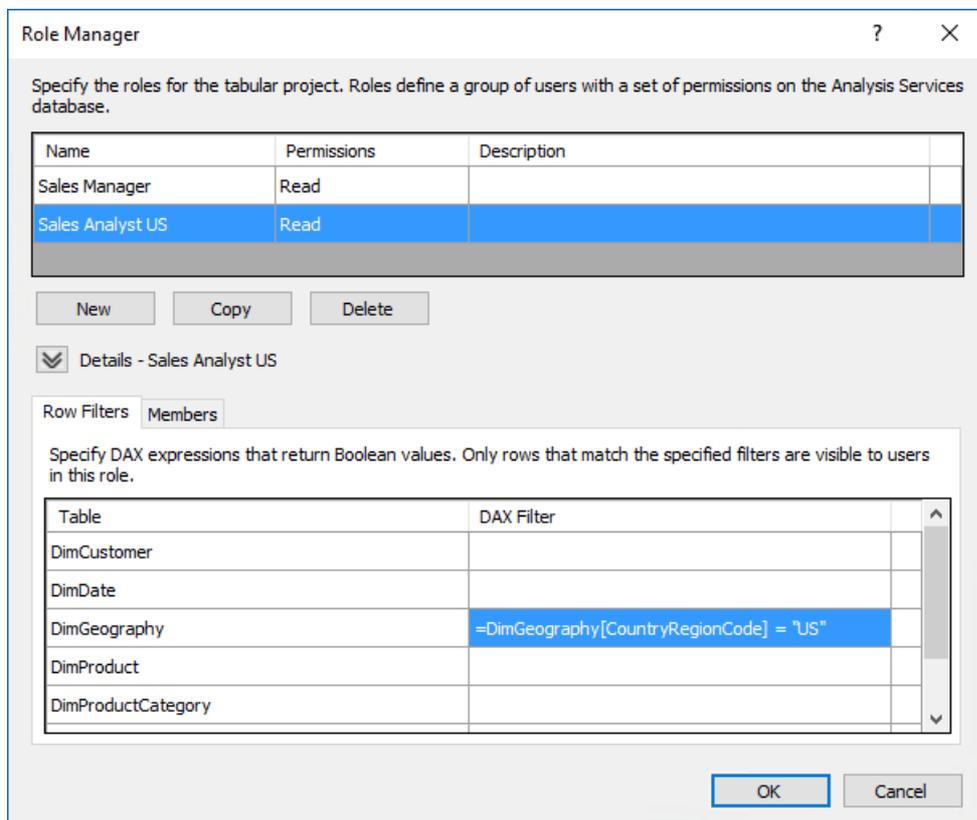
- Optional: Click the **Members** tab, and then click **Add**. In the **Select Users or Groups** dialog box, enter the Windows users or groups from your organization you want to include in the role.

To create a Sales Analyst US user role

- In Role Manager, click **New**.
- Rename the role to **Sales Analyst US**.
- Give this role **Read** permission.
- Click the Row Filters tab, and then for the **DimGeography** table only, in the DAX Filter column, type the following formula:

```
=DimGeography[CountryRegionCode] = "US"
```

A Row Filter formula must resolve to a Boolean (TRUE/FALSE) value. With this formula, you are specifying that only rows with the Country Region Code value of "US" are visible to the user.



- Optional: Click the **Members** tab, and then click **Add**. In the **Select Users or Groups** dialog box, enter the Windows users or groups from your organization you want to include in the role.

To create an Administrator user role

- Click **New**.
- Rename the role to **Administrator**.
- Give this role **Administrator** permission.
- Optional: Click the **Members** tab, and then click **Add**. In the **Select Users or Groups** dialog box, enter the Windows users or groups from your organization you want to include in the role.

What's next?

[Lesson 12: Analyze in Excel.](#)

Analyze in Excel

4/13/2018 • 2 min to read • [Edit Online](#)

In this lesson, you use the Analyze in Excel feature to open Microsoft Excel, automatically create a connection to the model workspace, and automatically add a PivotTable to the worksheet. The Analyze in Excel feature is meant to provide a quick and easy way to test the efficacy of your model design prior to deploying your model. You do not perform any data analysis in this lesson. The purpose of this lesson is to familiarize you, the model author, with the tools you can use to test your model design.

To complete this lesson, Excel must be installed on the same computer as Visual Studio.

Estimated time to complete this lesson: **Five minutes**

Prerequisites

This topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 11: Create roles](#).

Browse using the Default and Internet Sales perspectives

In these first tasks, you browse your model by using both the default perspective, which includes all model objects, and also by using the Internet Sales perspective you earlier. The Internet Sales perspective excludes the Customer table object.

To browse by using the Default perspective

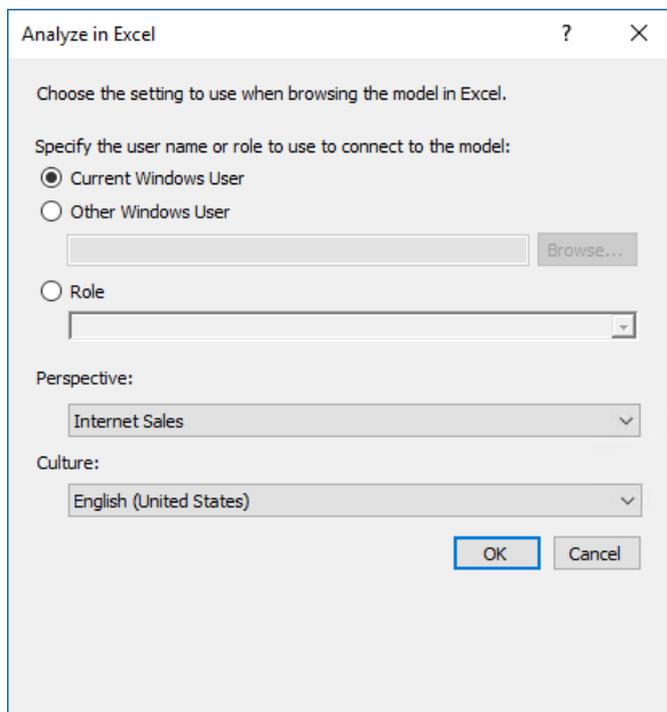
1. Click the **Model** menu > **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, click **OK**.

Excel opens with a new workbook. A data source connection is created using the current user account and the Default perspective is used to define viewable fields. A PivotTable is automatically added to the worksheet.

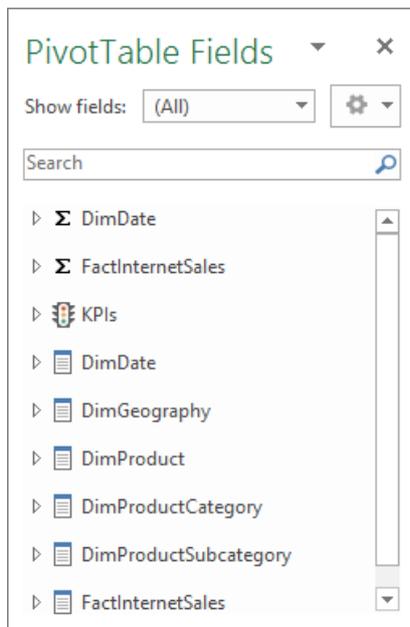
3. In Excel, in the **PivotTable Field List**, notice the **DimDate** and **FactInternetSales** measure groups appear. The **DimCustomer**, **DimDate**, **DimGeography**, **DimProduct**, **DimProductCategory**, **DimProductSubcategory**, and **FactInternetSales** tables with their respective columns also appear.
4. Close Excel without saving the workbook.

To browse by using the Internet Sales perspective

1. Click the **Model** menu, and then click **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, leave **Current Windows User** selected, then in the **Perspective** drop-down listbox, select **Internet Sales**, and then click **OK**.



- In Excel, in **PivotTable Fields**, notice the DimCustomer table is excluded from the field list.



- Close Excel without saving the workbook.

Browse by using roles

Roles are an important part of any tabular model. Without at least one role to which users are added as members, users cannot access and analyze data using your model. The Analyze in Excel feature provides a way for you to test the roles you have defined.

To browse by using the Sales Manager user role

- In SSDT, click the **Model** menu, and then click **Analyze in Excel**.
- In **Specify the user name or role to use to connect to the model**, select **Role**, and then in the drop-down listbox, select **Sales Manager**, and then click **OK**.

Excel opens with a new workbook. A PivotTable is automatically created. The Pivot Table Field List includes all the data fields available in your new model.

3. Close Excel without saving the workbook.

What's next?

Go to the next lesson: [Lesson 13: Deploy](#).

Deploy

4/13/2018 • 2 min to read • [Edit Online](#)

In this lesson, you configure deployment properties; specifying an Azure Analysis Services server to deploy to and a name for the model. You then deploy the model to that instance. After your model is deployed, users can connect to it by using a reporting client application. To learn more, see [Deploy to Azure Analysis Services](#).

Estimated time to complete this lesson: **5 minutes**

Prerequisites

This article is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this lesson, you should have completed the previous lesson: [Lesson 12: Analyze in Excel](#).

IMPORTANT

You must have [Administrator permissions](#) on the remote Analysis Services server in-order to deploy to it.

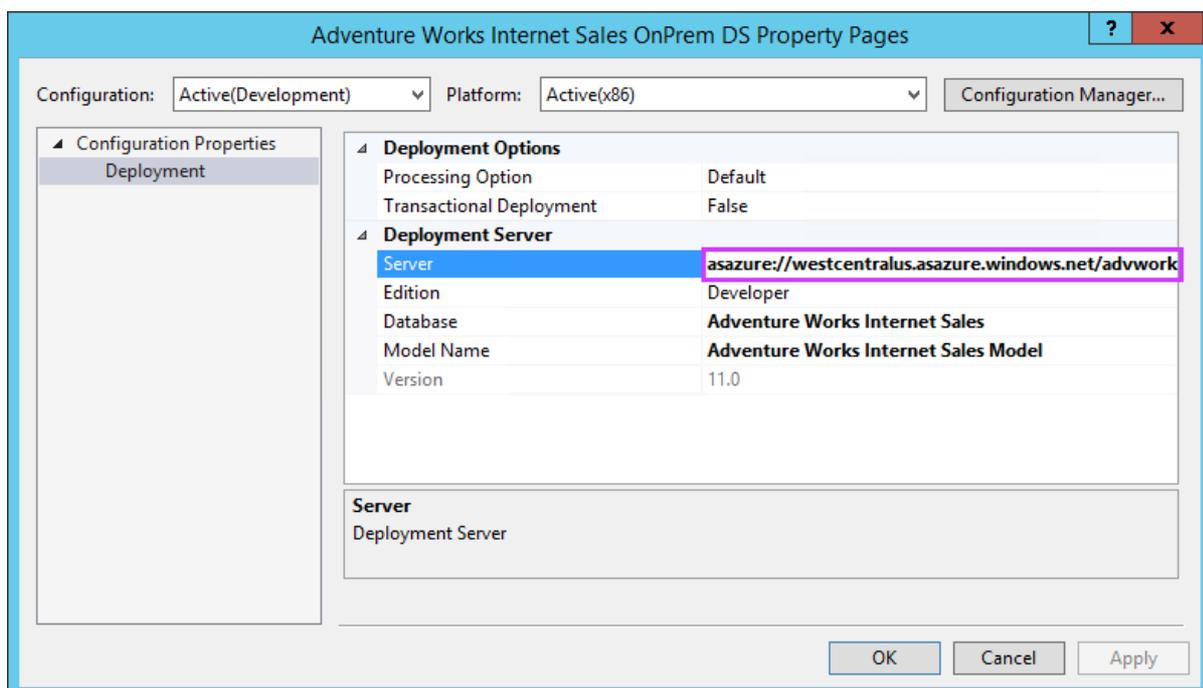
IMPORTANT

If you installed the AdventureWorksDW2014 sample database on an on-premises SQL Server, and you're deploying your model to an Azure Analysis Services server, an [On-premises data gateway](#) is required.

Deploy the model

To configure deployment properties

1. In **Solution Explorer**, right-click the **AW Internet Sales** project, and then click **Properties**.
2. In the **AW Internet Sales Property Pages** dialog box, under **Deployment Server**, in the **Server** property, enter the full server.



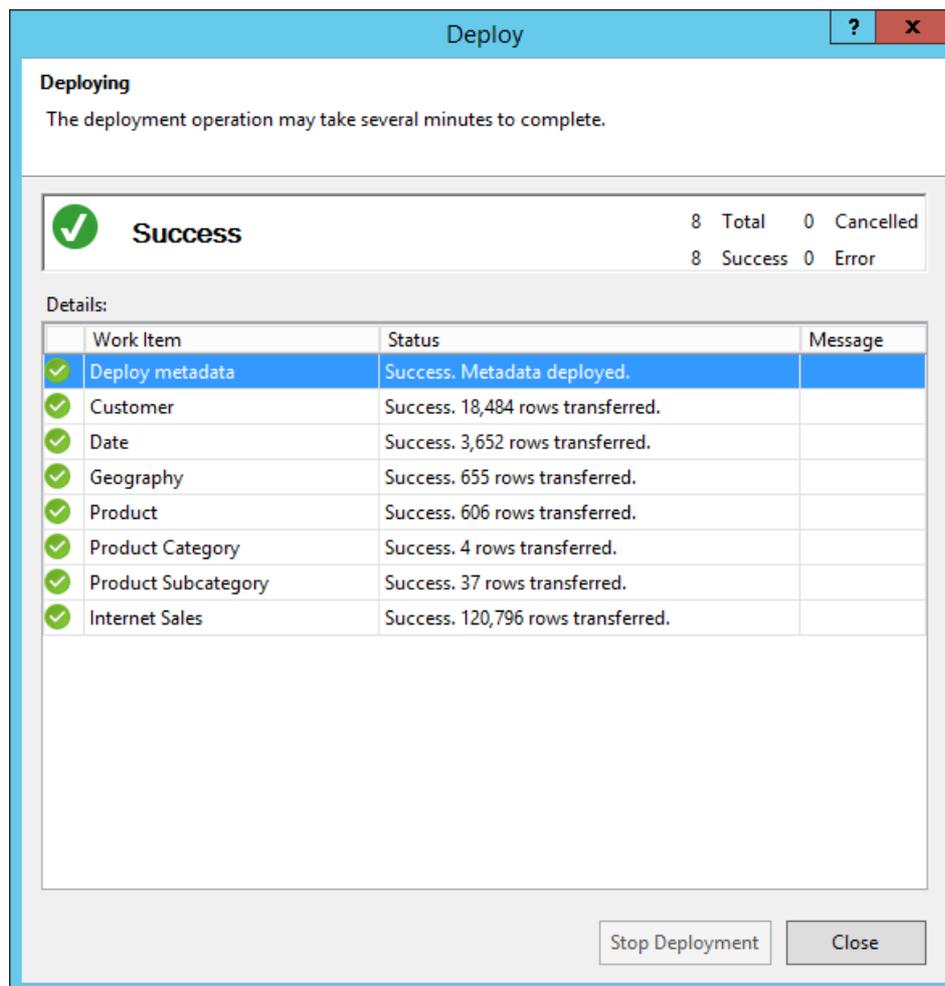
3. In the **Database** property, type **Adventure Works Internet Sales**.
4. In the **Model Name** property, type **Adventure Works Internet Sales Model**.
5. Verify your selections and then click **OK**.

To deploy the Adventure Works Internet Sales

1. In **Solution Explorer**, right-click the **AW Internet Sales** project > **Build**.
2. Right-click the **AW Internet Sales** project > **Deploy**.

When deploying to Azure Analysis Services, you may be prompted to enter your account. Enter your organizational account and password, for example nancy@adventureworks.com. This account must be in Admins on the server.

The Deploy dialog box appears and displays the deployment status of the metadata and each table included in the model.



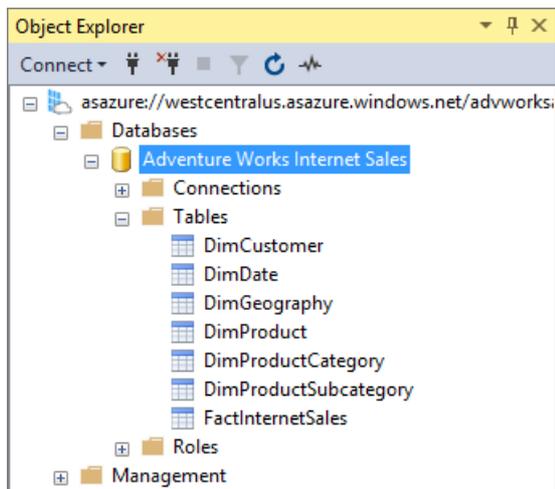
3. When deployment successfully completes, go ahead and click **Close**.

This lesson describes the most common and easiest method to deploy a tabular model from SSDT. Advanced deployment options such as the Deployment Wizard or automating with XMLA and AMO provide greater flexibility, consistency, and scheduled deployments. To learn more, see [Tabular model solution deployment](#).

Conclusion

Congratulations! You're finished authoring and deploying your first Analysis Services Tabular model. This tutorial has helped guide you through completing the most common tasks in creating a tabular model. Now that your Adventure Works Internet Sales model is deployed, you can use SQL Server Management Studio to manage the model; create process scripts and a backup plan. Users can also now connect to the model using a reporting client

application such as Microsoft Excel or Power BI.



What's next?

[Connect with Power BI Desktop](#)

[Supplemental Lesson - Dynamic security](#)

[Supplemental Lesson - Detail rows](#)

[Supplemental Lesson - Ragged hierarchies](#)

Supplemental lesson - Detail Rows

4/13/2018 • 1 min to read • [Edit Online](#)

In this supplemental lesson, you use the DAX Editor to define a custom Detail Rows Expression. A Detail Rows Expression is a property on a measure, providing end-users more information about the aggregated results of a measure.

Estimated time to complete this lesson: **10 minutes**

Prerequisites

This supplemental lesson is part of a tabular modeling tutorial. Before performing the tasks in this supplemental lesson, you should have completed all previous lessons or have a completed Adventure Works Internet Sales sample model project.

What's the issue?

Let's look at the details of the InternetTotalSales measure, before adding a Detail Rows Expression.

1. In SSDT, click the **Model** menu > **Analyze in Excel** to open Excel and create a blank PivotTable.
2. In **PivotTable Fields**, add the **InternetTotalSales** measure from the FactInternetSales table to **Values**, **CalendarYear** from the DimDate table to **Columns**, and **EnglishCountryRegionName** to **Rows**. The PivotTable now gives an aggregated results from the InternetTotalSales measure by regions and year.

	A	B	C	D	E	F	G
1	InternetTotalSales	Column Labels					
2	Row Labels	2010	2011	2012	2013	2014	Grand Total
3	Australia	\$20,909.78	\$2,563,732.25	\$2,128,407.46	\$4,339,443.38	\$8,507.72	\$9,061,000.58
4	Canada	\$3,578.27	\$571,571.80	\$307,604.52	\$1,085,632.65	\$9,457.62	\$1,977,844.86
5	France	\$3,399.99	\$410,845.33	\$648,065.54	\$1,578,511.80	\$3,195.06	\$2,644,017.71
6	Germany		\$520,500.16	\$608,657.98	\$1,761,876.36	\$3,277.83	\$2,894,312.34
7	United Kingdom	\$699.10	\$550,591.22	\$712,700.96	\$2,124,007.29	\$3,713.64	\$3,391,712.21
8	United States	\$14,833.90	\$2,458,285.17	\$1,437,048.73	\$5,462,078.86	\$17,542.85	\$9,389,789.51
9	Grand Total	\$43,421.04	\$7,075,525.93	\$5,842,485.20	\$16,351,550.34	\$45,694.72	\$29,358,677.22

3. In the PivotTable, double-click an aggregated value for a year and a region name. The value for Australia and the year 2014. A new sheet opens containing data, but not useful data.

	A	B	C	D
1	Data returned for InternetTotalSales, Australia, 2014 (First 1000 rows).			
2				
3	FactInternetSales[PromotionKey]	FactInternetSales[CurrencyKey]	FactInternetSales[SalesTerritoryKey]	FactInternetSales[SalesOrderNumber]
4	1	100	9	SO74417
5	1	100	9	SO74506
6	1	100	9	SO74653
7	1	100	9	SO74721
8	1	100	9	SO74757
9	1	100	9	SO74795
10	1	100	9	SO74812
11	1	100	9	SO74845
12	1	100	9	SO74874
13	1	100	9	SO74876
14	1	100	9	SO74938
15	1	100	9	SO74969
16	1	100	9	SO75032
17	1	100	9	SO75090

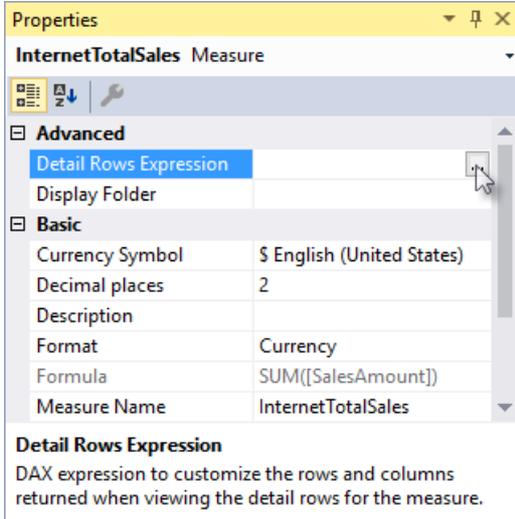
The goal here is a table containing columns and rows of data that contribute to the aggregated result of the

InternetTotalSales measure. To do that, add a Detail Rows Expression as a property of the measure.

Add a Detail Rows Expression

To create a Detail Rows Expression

1. In the FactInternetSales table's measure grid, click the **InternetTotalSales** measure.
2. In **Properties > Detail Rows Expression**, click the editor button to open the DAX Editor.



3. In DAX Editor, enter the following expression:

```
SELECTCOLUMNS(
    FactInternetSales,
    "Sales Order Number", FactInternetSales[SalesOrderNumber],
    "Customer First Name", RELATED(DimCustomer[FirstName]),
    "Customer Last Name", RELATED(DimCustomer[LastName]),
    "City", RELATED(DimGeography[City]),
    "Order Date", FactInternetSales[OrderDate],
    "Internet Total Sales", [InternetTotalSales]
)
```

This expression specifies names, columns, and measure results from the FactInternetSales table and related tables are returned when a user double-clicks an aggregated result in a PivotTable or report.

4. Back in Excel, delete the sheet created in Step 3, then double-click an aggregated value. This time, with a Detail Rows Expression property defined for the measure, a new sheet opens containing a lot more useful data.

	A	B	C	D	E	F	G
1	Data returned for InternetTotalSales, Australia, 2014 (First 1000 rows).						
2							
3	[Sales Order Number]	[Customer First Name]	[Customer Last Name]	[City]	[Order Date]	[Internet Total Sales]	
4	SO74417	Roger	Zhang	Sydney	1/6/2014	4.99	
5	SO74506	Joy	Gutierrez	Perth	1/9/2014	4.99	
6	SO74653	Chad	Raje	Melbourne	1/14/2014	4.99	
7	SO74721	Candace	Srini	North Ryde	1/16/2014	4.99	
8	SO74757	Roger	Sun	Rockhampton	1/17/2014	4.99	
9	SO74795	Bianca	Liu	Springwood	1/18/2014	4.99	
10	SO74812	Angel	Evans	Sunbury	1/19/2014	4.99	
11	SO74845	Jamie	Zhu	Findon	1/20/2014	4.99	
12	SO74874	Erick	Suri	Coffs Harbour	1/21/2014	4.99	
13	SO74876	Nina	Raje	Townsville	1/21/2014	4.99	
14	SO74938	Kellie	Blanco	East Brisbane	1/23/2014	4.99	
15	SO74969	Byron	Navarro	Newcastle	1/24/2014	4.99	
16	SO75032	Isaac	Phillips	Sydney	1/26/2014	4.99	
17	SO75090	Vincent	Zhang	Lane Cove	1/28/2014	4.99	

5. Redeploy your model.

See also

[SELECTCOLUMNS Function \(DAX\)](#)

[Supplemental lesson - Dynamic security](#)

[Supplemental lesson - Ragged hierarchies](#)

Supplemental lesson - Dynamic security

4/13/2018 • 9 min to read • [Edit Online](#)

In this supplemental lesson, you create an additional role that implements dynamic security. Dynamic security provides row-level security based on the user name or login id of the user currently logged on.

To implement dynamic security, you add a table to your model containing the user names of those users that can connect to the model and browse model objects and data. The model you create using this tutorial is in the context of Adventure Works; however, to complete this lesson, you must add a table containing users from your own domain. You do not need the passwords for the user names that are added. To create an EmployeeSecurity table, with a small sample of users from your own domain, you use the Paste feature, pasting employee data from an Excel spreadsheet. In a real-world scenario, the table containing user names would typically be a table from an actual database as a data source; for example, a real DimEmployee table.

To implement dynamic security, you use two DAX functions: [USERNAME Function \(DAX\)](#) and [LOOKUPVALUE Function \(DAX\)](#). These functions, applied in a row filter formula, are defined in a new role. By using the LOOKUPVALUE function, the formula specifies a value from the EmployeeSecurity table. The formula then passes that value to the USERNAME function, which specifies the user name of the user logged on belongs to this role. The user can then browse only data specified by the role's row filters. In this scenario, you specify that sales employees can only browse Internet sales data for the sales territories in which they are a member.

Those tasks that are unique to this Adventure Works tabular model scenario, but would not necessarily apply to a real-world scenario are identified as such. Each task includes additional information describing the purpose of the task.

Estimated time to complete this lesson: **30 minutes**

Prerequisites

This supplemental lesson topic is part of a tabular modeling tutorial, which should be completed in order. Before performing the tasks in this supplemental lesson, you should have completed all previous lessons.

Add the DimSalesTerritory table to the AW Internet Sales Tabular Model Project

To implement dynamic security for this Adventure Works scenario, you must add two additional tables to your model. The first table you add is DimSalesTerritory (as Sales Territory) from the same AdventureWorksDW database. You later apply a row filter to the SalesTerritory table that defines the particular data the logged on user can browse.

To add the DimSalesTerritory table

1. In Tabular Model Explorer > **Data Sources**, right-click your connection, and then click **Import New Tables**.
If the Impersonation Credentials dialog box appears, type the impersonation credentials you used in Lesson 2: Add Data.
2. In Navigator, select the **DimSalesTerritory** table, and then click **OK**.
3. In Query Editor, click the **DimSalesTerritory** query, and then remove **SalesTerritoryAlternateKey** column.
4. Click **Import**.

The new table is added to the model workspace. Objects and data from the source DimSalesTerritory table are then imported into your AW Internet Sales Tabular Model.

5. After the table has been imported successfully, click **Close**.

Add a table with user name data

The DimEmployee table in the AdventureWorksDW sample database contains users from the AdventureWorks domain. Those user names do not exist in your own environment. You must create a table in your model that contains a small sample (at least three) of actual users from your organization. You then add these users as members to the new role. You do not need the passwords for the sample user names, but you do need actual Windows user names from your own domain.

To add an EmployeeSecurity table

1. Open Microsoft Excel, creating a worksheet.
2. Copy the following table, including the header row, and then paste it into the worksheet.

```
|EmployeeId|SalesTerritoryId|FirstName|LastName|LoginId|
|-----|-----|-----|-----|-----|
|1|2|<user first name>|<user last name>|\<domain\username>|
|1|3|<user first name>|<user last name>|\<domain\username>|
|2|4|<user first name>|<user last name>|\<domain\username>|
|3|5|<user first name>|<user last name>|\<domain\username>|
```

3. Replace the first name, last name, and domain\username with the names and login ids of three users in your organization. Put the same user on the first two rows, for EmployeeId 1, showing this user belongs to more than one sales territory. Leave the EmployeeId and SalesTerritoryId fields as they are.
 4. Save the worksheet as **SampleEmployee**.
 5. In the worksheet, select all the cells with employee data, including the headers, then right-click the selected data, and then click **Copy**.
 6. In SSDT, click the **Edit** menu, and then click **Paste**.
- If Paste is grayed out, click any column in any table in the model designer window, and try again.
7. In the **Paste Preview** dialog box, in **Table Name**, type **EmployeeSecurity**.
 8. In **Data to be pasted**, verify the data includes all the user data and headers from the SampleEmployee worksheet.
 9. Verify **Use first row as column headers** is checked, and then click **Ok**.

A new table named EmployeeSecurity with employee data copied from the SampleEmployee worksheet is created.

Create relationships between FactInternetSales, DimGeography, and DimSalesTerritory table

The FactInternetSales, DimGeography, and DimSalesTerritory table all contain a common column, SalesTerritoryId. The SalesTerritoryId column in the DimSalesTerritory table contains values with a different Id for each sales territory.

To create relationships between the FactInternetSales, DimGeography, and the DimSalesTerritory table

1. In Diagram View, in the **DimGeography** table, click, and hold on the **SalesTerritoryId** column, then drag the cursor to the **SalesTerritoryId** column in the **DimSalesTerritory** table, and then release.

2. In the **FactInternetSales** table, click, and hold on the **SalesTerritoryId** column, then drag the cursor to the **SalesTerritoryId** column in the **DimSalesTerritory** table, and then release.

Notice the Active property for this relationship is False, meaning it's inactive. The FactInternetSales table already has another active relationship.

Hide the EmployeeSecurity Table from client applications

In this task, you hide the EmployeeSecurity table, keeping it from appearing in a client application's field list. Keep in mind that hiding a table does not secure it. Users can still query EmployeeSecurity table data if they know how. To secure the EmployeeSecurity table data, preventing users from being able to query any of its data, you apply a filter in a later task.

To hide the EmployeeSecurity table from client applications

- In the model designer, in Diagram View, right-click the **Employee** table heading, and then click **Hide from Client Tools**.

Create a Sales Employees by Territory user role

In this task, you create a user role. This role includes a row filter defining which rows of the DimSalesTerritory table are visible to users. The filter is then applied in the one-to-many relationship direction to all other tables related to DimSalesTerritory. You also apply a filter that secures the entire EmployeeSecurity table from being queryable by any user that is a member of the role.

NOTE

The Sales Employees by Territory role you create in this lesson restricts members to browse (or query) only sales data for the sales territory to which they belong. If you add a user as a member to the Sales Employees by Territory role that also exists as a member in a role created in [Lesson 11: Create Roles](#), you get a combination of permissions. When a user is a member of multiple roles, the permissions, and row filters defined for each role are cumulative. That is, the user has the greater permissions determined by the combination of roles.

To create a Sales Employees by Territory user role

1. In SSDT, click the **Model** menu, and then click **Roles**.
2. In **Role Manager**, click **New**.

A new role with the None permission is added to the list.
3. Click the new role, and then in the **Name** column, rename the role to **Sales Employees by Territory**.
4. In the **Permissions** column, click the dropdown list, and then select the **Read** permission.
5. Click the **Members** tab, and then click **Add**.
6. In the **Select User or Group** dialog box, in **Enter the object named to select**, type the first sample user name you used when creating the EmployeeSecurity table. Click **Check Names** to verify the user name is valid, and then click **Ok**.

Repeat this step, adding the other sample user names you used when creating the EmployeeSecurity table.
7. Click the **Row Filters** tab.
8. For the **EmployeeSecurity** table, in the **DAX Filter** column, type the following formula:

```
=FALSE()
```

This formula specifies that all columns resolve to the false Boolean condition. No columns for the EmployeeSecurity table can be queried by a member of the Sales Employees by Territory user role.

9. For the **DimSalesTerritory** table, type the following formula:

```
= 'Sales Territory'[Sales Territory Id]=LOOKUPVALUE('Employee Security'[Sales Territory Id],  
'Employee Security'[Login Id], USERNAME(),  
'Employee Security'[Sales Territory Id],  
'Sales Territory'[Sales Territory Id])
```

In this formula, the LOOKUPVALUE function returns all values for the DimEmployeeSecurity[SalesTerritoryId] column, where the EmployeeSecurity[LoginId] is the same as the current logged on Windows user name, and EmployeeSecurity[SalesTerritoryId] is the same as the DimSalesTerritory[SalesTerritoryId].

The set of sales territory IDs returned by LOOKUPVALUE is then used to restrict the rows shown in the DimSalesTerritory table. Only rows where the SalesTerritoryID for the row is in the set of IDs returned by the LOOKUPVALUE function are displayed.

10. In Role Manager, click **Ok**.

Test the Sales Employees by Territory User Role

In this task, you use the Analyze in Excel feature in SSDT to test the efficacy of the Sales Employees by Territory user role. You specify one of the user names you added to the EmployeeSecurity table and as a member of the role. This user name is then used as the effective user name in the connection created between Excel and the model.

To test the Sales Employees by Territory user role

1. In SSDT, click the **Model** menu, and then click **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, in **Specify the user name or role to use to connect to the model**, select **Other Windows User**, and then click **Browse**.
3. In the **Select User or Group** dialog box, in **Enter the object name to select**, type a user name you included in the EmployeeSecurity table, and then click **Check Names**.
4. Click **Ok** to close the **Select User or Group** dialog box, and then click **Ok** to close the **Analyze in Excel** dialog box.

Excel opens with a new workbook. A PivotTable is automatically created. The PivotTable Fields list includes most of the data fields available in your new model.

Notice the EmployeeSecurity table is not visible in the PivotTable Fields list. You hid this table from client tools in a previous task.

5. In the **Fields** list, in **Σ Internet Sales** (measures), select the **InternetTotalSales** measure. The measure is entered into the **Values** fields.
6. Select the **SalesTerritoryId** column from the **DimSalesTerritory** table. The column is entered into the **Row Labels** fields.

Notice Internet sales figures appear only for the one region to which the effective user name you used belongs. If you select another column, like City from the DimGeography table as Row Label field, only cities in the sales territory to which the effective user belongs are displayed.

This user cannot browse or query any Internet sales data for territories other than the one they belong to. This restriction is because the row filter defined for the DimSalesTerritory table, in the Sales Employees by

Territory user role, secures data for all data related to other sales territories.

See Also

[USERNAME Function \(DAX\)](#)

[LOOKUPVALUE Function \(DAX\)](#)

[CUSTOMDATA Function \(DAX\)](#)

Supplemental lesson - Ragged hierarchies

4/13/2018 • 2 min to read • [Edit Online](#)

In this supplemental lesson, you resolve a common problem when pivoting on hierarchies that contain blank values (members) at different levels. For example, an organization where a high-level manager has both departmental managers and non-managers as direct reports. Or, geographic hierarchies composed of Country-Region-City, where some cities lack a parent State or Province, such as Washington D.C., Vatican City. When a hierarchy has blank members, it often descends to different, or ragged, levels.

Employee	ResellerTotalSales
Ken J Sánchez	\$80,450,596.98
Brian S Welcker	\$80,450,596.98
Amy E Alberts	\$15,535,946.26
	\$732,078.44
	\$732,078.44
Jae B Pak	\$8,503,338.65
	\$8,503,338.65
Rachel B Valdez	\$1,790,640.23
	\$1,790,640.23
Ranjit R Varkey Chudukatil	\$4,509,888.93
	\$4,509,888.93
Stephen Y Jiang	\$63,320,315.35

Tabular models at the 1400 compatibility level have an additional **Hide Members** property for hierarchies. The **Default** setting assumes there are no blank members at any level. The **Hide blank members** setting excludes blank members from the hierarchy when added to a PivotTable or report.

Estimated time to complete this lesson: **20 minutes**

Prerequisites

This supplemental lesson topic is part of a tabular modeling tutorial. Before performing the tasks in this supplemental lesson, you should have completed all previous lessons or have a completed Adventure Works Internet Sales sample model project.

If you've created the AW Internet Sales project as part of the tutorial, your model does not yet contain any data or hierarchies that are ragged. To complete this supplemental lesson, you first have to create the problem by adding some additional tables, create relationships, calculated columns, a measure, and a new Organization hierarchy. That part takes about 15 minutes. Then, you get to solve it in just a few minutes.

Add tables and objects

To add new tables to your model

1. In Tabular Model Explorer, expand **Data Sources**, then right-click your connection > **Import New Tables**.
2. In Navigator, select **DimEmployee** and **FactResellerSales**, and then click **OK**.
3. In Query Editor, click **Import**
4. Create the following [relationships](#):

TABLE 1	COLUMN	FILTER DIRECTION	TABLE 2	COLUMN	ACTIVE
FactResellerSales	OrderDateKey	Default	DimDate	Date	Yes
FactResellerSales	DueDate	Default	DimDate	Date	No
FactResellerSales	ShipDateKey	Default	DimDate	Date	No
FactResellerSales	ProductKey	Default	DimProduct	ProductKey	Yes
FactResellerSales	EmployeeKey	To Both Tables	DimEmployee	EmployeeKey	Yes

5. In the **DimEmployee** table, create the following [calculated columns](#):

Path

```
=PATH([EmployeeKey],[ParentEmployeeKey])
```

FullName

```
=[FirstName] & " " & [MiddleName] & " " & [LastName]
```

Level1

```
=LOOKUPVALUE(DimEmployee[FullName],DimEmployee[EmployeeKey],PATHITEM([Path],1,1))
```

Level2

```
=LOOKUPVALUE(DimEmployee[FullName],DimEmployee[EmployeeKey],PATHITEM([Path],2,1))
```

Level3

```
=LOOKUPVALUE(DimEmployee[FullName],DimEmployee[EmployeeKey],PATHITEM([Path],3,1))
```

Level4

```
=LOOKUPVALUE(DimEmployee[FullName],DimEmployee[EmployeeKey],PATHITEM([Path],4,1))
```

Level5

```
=LOOKUPVALUE(DimEmployee[FullName],DimEmployee[EmployeeKey],PATHITEM([Path],5,1))
```

6. In the **DimEmployee** table, create a [hierarchy](#) named **Organization**. Add the following columns in-order:

Level1, Level2, Level3, Level4, Level5.

7. In the **FactResellerSales** table, create the following [measure](#):

```
ResellerTotalSales:=SUM([SalesAmount])
```

- Use [Analyze in Excel](#) to open Excel and automatically create a PivotTable.
- In **PivotTable Fields**, add the **Organization** hierarchy from the **DimEmployee** table to **Rows**, and the **ResellerTotalSales** measure from the **FactResellerSales** table to **Values**.

The screenshot shows an Excel spreadsheet with a PivotTable. The PivotTable has two columns: 'Employee' and 'ResellerTotalSales'. The data is as follows:

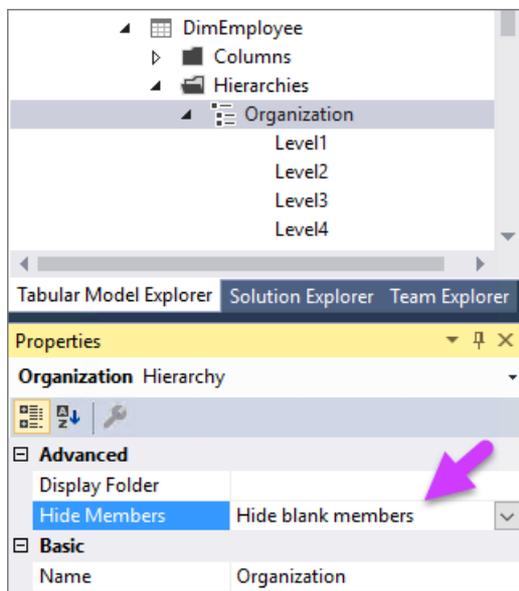
Employee	ResellerTotalSales
Ken J Sánchez	\$80,450,596.98
Brian S Welcker	\$80,450,596.98
Amy E Alberts	\$15,535,946.26
Jae B Pak	\$732,078.44
Rachel B Valdez	\$8,503,338.65
Ranjit R Varkey Chudukatil	\$8,503,338.65
Stephen Y Jiang	\$1,790,640.23
David R Campbell	\$1,790,640.23
Garrett R Vargas	\$4,509,888.93
Jillian Carson	\$4,509,888.93
	\$63,320,315.35
	\$1,092,123.86
	\$3,729,945.35
	\$3,729,945.35
	\$3,609,447.22
	\$3,609,447.22
	\$10,065,803.54

The PivotTable Fields task pane on the right shows 'DimEmployee' expanded to 'Organization', which is placed in the 'Rows' area. 'ResellerTotalSales' is placed in the 'Values' area.

As you can see in the PivotTable, the hierarchy displays rows that are ragged. There are many rows where blank members are shown.

To fix the ragged hierarchy by setting the Hide members property

- In **Tabular Model Explorer**, expand **Tables > DimEmployee > Hierarchies > Organization**.
- In **Properties > Hide Members**, select **Hide blank members**.



- Back in Excel, refresh the PivotTable.

Employee	ResellerTotalSales		
Ken J Sánchez	\$80,450,596.98		
Brian S Welcker	\$80,450,596.98		
Amy E Alberts	\$15,535,946.26		
Jae B Pak	\$8,503,338.65		
Rachel B Valdez	\$1,790,640.23		
Ranjit R Varkey Chudukatil	\$4,509,888.93		
Stephen Y Jiang	\$63,320,315.35		
David R Campbell	\$3,729,945.35		
Garrett R Vargas	\$3,609,447.22		
Jillian Carson	\$10,065,803.54		
José Edvaldo Saraiva	\$5,926,418.36		
Linda C Mitchell	\$10,367,007.43		
Michael G Blythe	\$9,293,903.01		
Pamela O Ansman-Wolfe	\$3,325,102.60		
Shu K Ito	\$6,427,005.56		
Tete A Mensa-Annan	\$2,312,545.69		
Tsvi Michael Reiter	\$7,171,012.75		

PivotTable Fields

Show fields: (All) ⚙️

Search

▾ DimEmployee
 ▾

▾ Organization
 ▾

Drag fields between areas below:

<p>Filters</p> <div style="border: 1px solid gray; height: 20px; width: 100%;"></div>	<p>Columns</p> <div style="border: 1px solid gray; height: 20px; width: 100%;"></div>
<p>Rows</p> <div style="border: 1px solid gray; padding: 2px;">Organization ▾</div>	<p>Values</p> <div style="border: 1px solid gray; padding: 2px;">ResellerTotalSales ▾</div>

Defer Layout Update Update

Now that looks a whole lot better!

See Also

[Lesson 9: Create hierarchies](#)

[Supplemental Lesson - Dynamic security](#)

[Supplemental Lesson - Detail rows](#)

Manage Azure Analysis Services with PowerShell

4/13/2018 • 1 min to read • [Edit Online](#)

This article describes PowerShell cmdlets used to perform Azure Analysis Services server and database management tasks.

Server management tasks such as creating or deleting a server, suspending or resuming server operations, or changing the service level (tier) use Azure Resource Manager (AzureRM) cmdlets. Other tasks for managing databases such as adding or removing role members, processing, or partitioning use cmdlets included in the same SqlServer module as SQL Server Analysis Services.

Permissions

Most PowerShell tasks require you have Admin privileges on the Analysis Services server you are managing. Scheduled PowerShell tasks are unattended operations. The account running the scheduler must have Admin privileges on the Analysis Services server.

For server operations using AzureRm cmdlets, your account or the account running scheduler must also belong to the Owner role for the resource in [Azure Role-Based Access Control \(RBAC\)](#).

Server operations

Azure Analysis Services cmdlets are included in the [AzureRM.AnalysisServices](#) component module. To install AzureRM cmdlet modules, see [Azure Resource Manager cmdlets](#) in the PowerShell Gallery.

CMDLET	DESCRIPTION
Add-AzureAnalysisServicesAccount	Adds an authenticated account to use for Azure Analysis Services server cmdlet requests.
Get-AzureRmAnalysisServicesServer	Gets details of a server instance.
New-AzureRmAnalysisServicesServer	Creates a server instance.
Remove-AzureRmAnalysisServicesServer	Removes a server instance.
Restart-AzureAnalysisServicesInstance	Restarts an instance of Analysis Services server in the currently logged in environment; specified in Add-AzureAnalysisServicesAccount command.
Resume-AzureRmAnalysisServicesServer	Resumes a server instance.
Suspend-AzureRmAnalysisServicesServer	Suspends a server instance.
Set-AzureRmAnalysisServicesServer	Modifies a server instance.
Test-AzureRmAnalysisServicesServer	Tests the existence of a server instance.

Database operations

Azure Analysis Services database operations use the same [SqlServer](#) module as SQL Server Analysis Services. However, not all cmdlets are supported for Azure Analysis Services.

The SqlServer module provides task-specific database management cmdlets as well as the general-purpose Invoke-ASCmd cmdlet that accepts a Tabular Model Scripting Language (TMSL) query or script. The following cmdlets in the SqlServer module are supported for Azure Analysis Services.

CMDLET	DESCRIPTION
Add-RoleMember	Add a member to a database role.
Backup-ASDatabase	Backup an Analysis Services database.
Remove-RoleMember	Remove a member from a database role.
Invoke-ASCmd	Execute a TMSL script.
Invoke-ProcessASDatabase	Process a database.
Invoke-ProcessPartition	Process a partition.
Invoke-ProcessTable	Process a table.
Merge-Partition	Merge a partition.
Restore-ASDatabase	Restore an Analysis Services database.

Related information

- [Download SQL Server PowerShell Module](#)
- [Download SSMS](#)
- [SqlServer module in PowerShell Gallery](#)
- [Tabular Model Programming for Compatibility Level 1200 and higher](#)

Azure Analysis Services samples

4/13/2018 • 1 min to read • [Edit Online](#)

Use the following sample resources to help you learn about and test Analysis Services for your environment.

Code samples

The [Analysis Services](#) repository on GitHub includes open source code samples and community projects.

Tabular model project and database samples

[Adventure Works for Analysis Services](#) on GitHub is the most commonly used sample tabular model project. You can download a VS project or a completed sample tabular model database.

Sample databases on GitHub

These sample databases on GitHub can be used for creating and testing your own models.

[Adventure Works sample databases](#)

[Wide World Importers sample databases](#)

Adventure Works Internet Sales sample model in Azure portal

If you have an Analysis Services server in Azure portal, you can quickly and easily create a sample model. In your server's overview page in Azure portal, click **New model**, and then in **Choose a data source**, select **Sample data**.

Next steps

[Adventure Works tutorial](#)

[Azure On-premises Data Gateway](#)