**An overview of Role-based Security in Analysis Services:**

The two general types of Roles in Analysis Services;

**A discussion of Role and Role Member Objects, focusing upon:**

Ø  The Server Role

Ø  The Database Role

Access Rights and Permissions (including Permissions Inheritance and Permissions Objects)

**Role-Based Security in Analysis Services**

Roles are used in Analysis Services to manage security for Analysis Services objects and data. In basic terms, a role associates the security identifiers (SIDs) of Microsoft Windows users and groups that have specific access rights and permissions defined for objects managed by an instance of Analysis Services.

Two general types of roles are provided in Analysis Services:

* The **Server Role**: a fixed role that provides administrator access to an instance of Analysis Services.
* **Database roles**: roles defined by administrators to control access to objects and data for non-administrator users.

Security in Analysis Services is managed by using roles and permissions. Roles are groups of users. Users, also called members, can be added or removed from roles. Permissions for objects are specified by roles, and all members in a role can use the objects for which the role has permissions. All members in a role have equal permissions to the objects. Permissions are particular to objects. Each object has a permissions collection with the permissions granted on that object and different sets of permissions can be granted on an object. Each permission, from the permissions collection of the object, has a single role assigned to it.

Database and cube level security can be maintained in either SSDT or Management Studio (if you are using a SSAS Project, which helps with deployment and version control, remember each deployment will overwrite the current security unless you use the deployment. My preference is to use SSDT, although the screens are very similar in either Management Studio or SSDT. In order to create a new role, simply right click on Roles and Select New Roles.

As shown on the below screen print, step 1 is to define the Role Name in the properties windows; use a name which is both descriptive and meaningful. If the properties window is not visible, select View > Properties (or hit F4). Next, fill in the Role Description as needed. As important as the name is, the three other check boxes below the role description play a vital role in defining database level access to the members of this role.  Each check box is described next:

* **Full control:** This permission grants full access to this particular SSAS database. Members with this permission have similar access rights as the server role noted above. However, these permissions apply to this database only, and not to all the databases within the SSAS instance. Full control grants members access to add other users and to process and maintain databases, cubes, and dimensions.
* **Process database:** This permission allows a member to process this database and its related cubes and dimensions.
* **Read definitions:** This permission allows role members to read the database metadata. Granting this access, though, does not provide access to read the metadata of other objects within the database such as cube meta data or dimension metadata. This lower level meta data access will be discussed later in the article.

Often, none of these check boxes are checked for regular users of the cube.

Switching to the members tab allows the administrator to add specific users to the role. Membership can be added either by typing the individual users or by using the windows Advanced > Find option. Again, you can add active directory groups, local windows groups, domain users, or local users; however, SQL users cannot be used! The Add members processes is illustrated in the following screen print. Of course for manageability, using groups is recommended.

Moving on to the Data Source tab, intuitively, it would seem that a user would need access to the data source. However, granting access to the data source actually grants permissions to the underlying data sources of the SSAS project or database.

Switching to the Cubes tab finally provides methods which will grant read data access to role members.

* None: Role members are not able to access this particular cube.
* Read: Users can read data from the cube, but not write data back to the cube.
* Read/Write: Users can read data from cube and write data back to the cube.
* Process:
* Role members are able to process this particular cube.

Remember, if access is not specifically granted to the cube, end users will not even see the cube from their client applications.

The Cell Data tab affords administrators the ability to granularly set read, read-contingent, and read/write permissions to role members. Once any of these options are enabled by selecting the appropriate check box, the administrator must enter a MDX expression which defines the cells which are available or restricted for the role members.

The Dimension tab assigns privileges to role members at the dimension level. First, two options exists at the dimensions level, either Read or Read and Write. Roles whose access is set to none will not even see the dimension marked as such. Furthermore, roles can be granted Read Definition access which allows for the role members to read the metadata concerning the dimension. Similar to the Cubes Access Rights, granting Process rights to a role, allows the members to process that particular dimension. The next tab to the right of the dimension tab is the Dimensions data tab; moving from dimension security at the dimension level, the dimension data tab allows the administrator to restrict or deny access to certain attribute values.

One caveat in the use of Dimension Data security; total rows will display the total for all values in the dimension even those that restricted which in turn could allow end users to determine the value of the restricted attributes.

**Conclusion-Security:**

Implementing SSAS Security is as important as every other methods of restricting organizational data.  AS a DBA/ DWA, we are ultimately responsible for providing not only the correct and appropriate data (all very quickly), but we also must provide such data using a method and path which appropriately protects access to a SSAS database with such data. Therefore, SSAS allows us to setup the following security points:

* who administers the database
* who processes the databases, cubes, and dimensions
* who can access the meta data about the cubes and dimensions
* who can read data from the SSAS database
* including drill through
* dimension level and dimension data security

Using SSAS roles impersonation

## 1. Overview

SQL Server Analysis Services (SSAS) relies on Windows authentication and roles  to determine user permissions. When you create a new data connector in Dundas BI using the SSAS data provider, you have the option of connecting using Roles Impersonation, which allows you to control access to cube data on a per-user basis. Users will see different data displayed on dashboards, for example, depending on the Dundas BI user group (and corresponding SSAS role) they belong to.

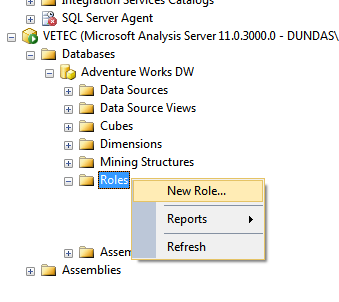
The idea behind roles impersonation is simple and basically consists of three parts:

1. Set up roles in SSAS and restrict access to data accordingly for each role.
2. Create user groups in Dundas BI that correspond exactly to the SSAS roles.
3. Connect to SSAS using the Roles impersonation option.

## 2. Set up roles in SSAS

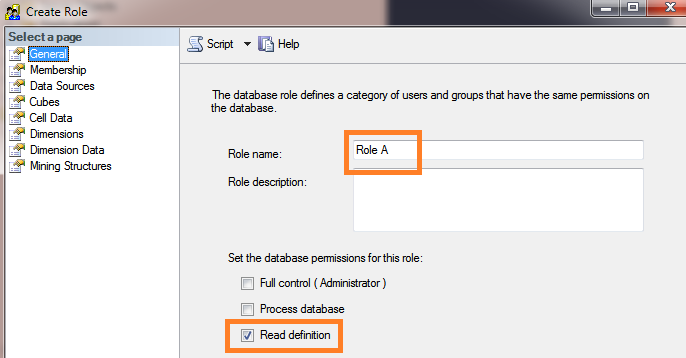
Launch SQL Server Management Studio and connect to Analysis Services.

Expand your SSAS database and right-click on the **Roles** folder. From the menu, select **New Role**.

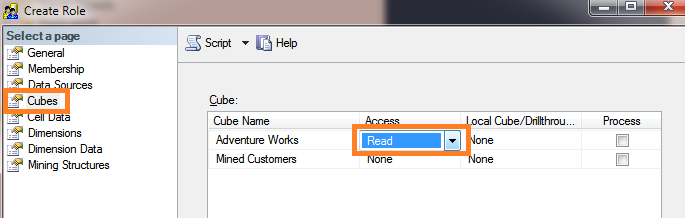
[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-new-role.png)

In the **General** page of the Create Role dialog, set the following fields:

* Role name: Role A
* Set the database permissions for this role: Read definition

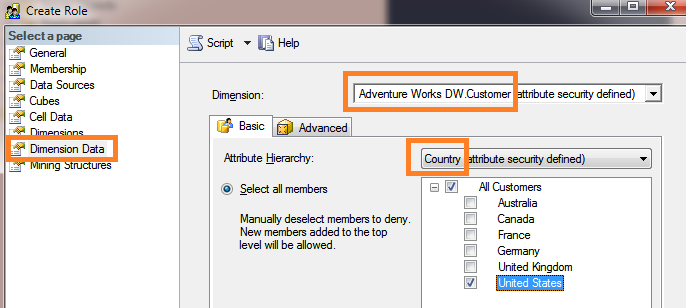
[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-create-role.png)

Go to the **Cubes** page of the Create Role dialog and give read-access to a specific cube.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-read-access.png)

In the **Dimension Data** page, restrict access to cube data for this role. For example:

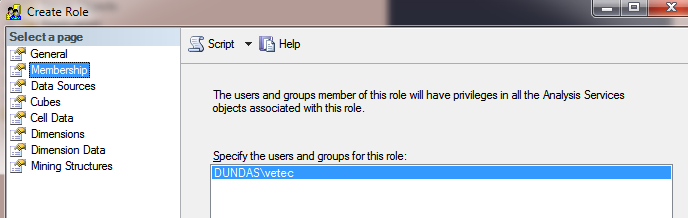
* Select the Customer dimension.
* Select the Country attribute hierarchy.
* Uncheck all members except United States.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-dimdata-rolea.png)

Note

When using role impersonation on a native OLAP cube, the user will see the totals for All the members in the cube. If you want the totals to be the aggregate for only the visible cell values, open the **Advanced** tab in the **Dimension Data** page and enable the **Enable Visual Totals** option.

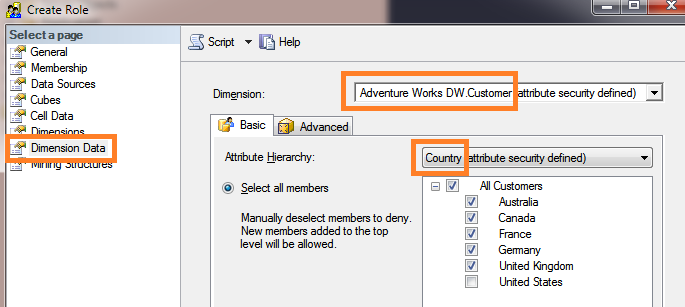
In the **Membership** page, add a domain user to the role. This will be the Windows credentials needed when creating the data connector in Dundas BI.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-membership.png)

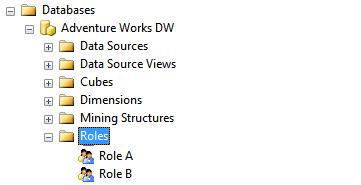
Click **OK** in the Create Role dialog to finish creating Role A.

Next, follow the same steps as above to create Role B. The only difference is in the Dimension Data page where you'll want to restrict access to different cube data for the role. For example:

* Select the Customer dimension.
* Select the Country attribute hierarchy.
* Check all members except United States.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-dimdata-roleb.png)

Refresh the SSAS database node to see the two roles.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-completed.png)

## 3. Set up user groups and accounts in Dundas BI

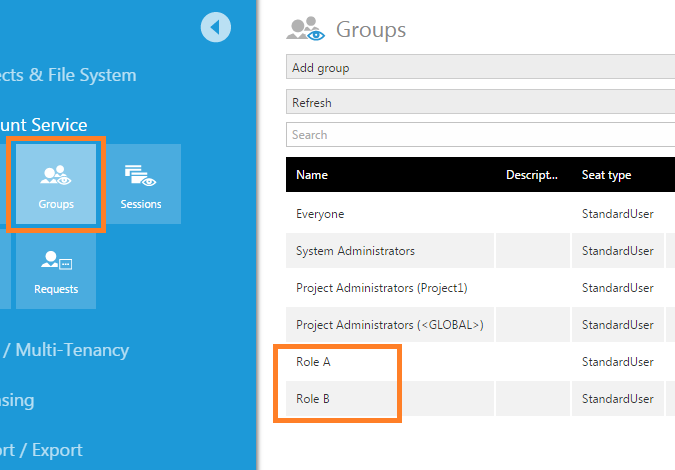
Log on to Dundas BI as an administrator and go to the Admin screen. Expand Account Service and click **Groups**.

For each SSAS role, create a corresponding [group](https://www.dundas.com/support/learning/documentation/administration/add-a-new-group) in Dundas BI with the **exact same name** as the role.

Note that role/group names cannot include any characters from this list:

. , ; ' ` : / \ \* | ? " % $ ! + = ( ) [ ] { } < >

The underscore (\_) character is allowed.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-dbi-groups.png)

Next, create a new user [account](https://www.dundas.com/support/learning/documentation/administration/add-an-account) named User A and add this as a member of the group Role A.

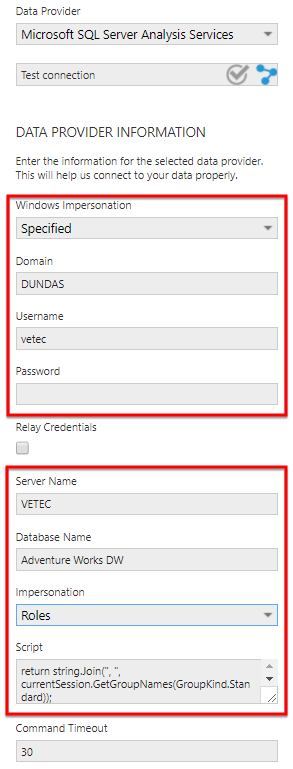
Similarly, create a new user account named User B and add this as a member of the group Role B.

## 4. Create a new project and dashboard

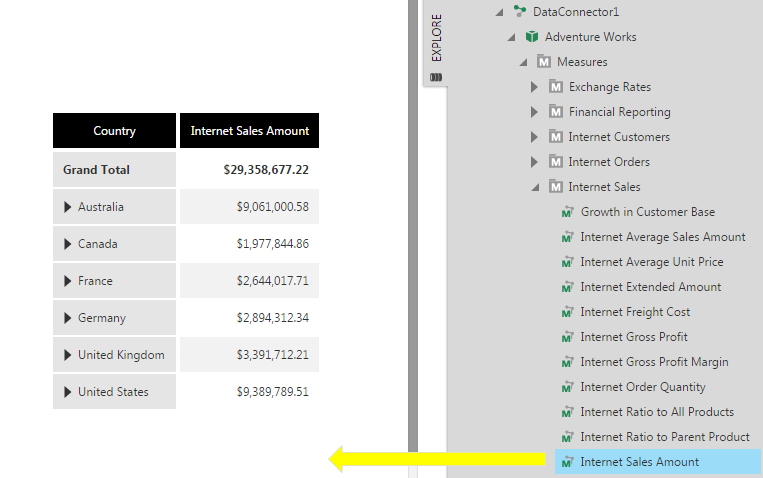
Log on to Dundas BI as an administrator or developer and create a new project (**Project1**) which will be accessible to all users. Make sure Project1 is the active project.

Create a new data connector (**DataConnector1**) using the SQL Server Analysis Services data provider:

* Set **Windows Impersonation** to Specified and enter the domain credentials that were added to the SSAS Membership pages.
* Set the **Database Name** to the name of the SSAS database.
* Set the **Impersonation** field to Roles.
* Leave the **Script** field at its default setting. This C# script [returns the names of the Dundas BI groups](https://www.dundas.com/support/api-docs/net/#html/M_Dundas_BI_AccountServices_Session_GetGroupNames.htm%3FTocPath%3DDundas.BI.AccountServices%7CSession%20Class%7CSession%20Methods%7C_____4) the current user belongs to (excluding built-in Dundas BI groups such as System Administrators and Everyone).

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-data-connector-v5.png)

Then create a new dashboard (**Dashboard1**) by dragging the Internet Sales Amount measure and the [Customer].[Customer Geography] hierarchy from DataConnector1 to the dashboard canvas.

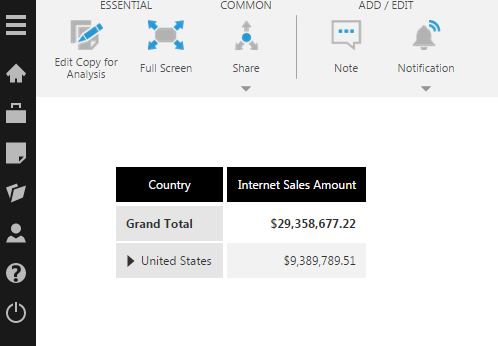
[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-dashboard.png)

Right click on Dashboard1 in the Explore window and check it in.

## 5. View the dashboard as each user

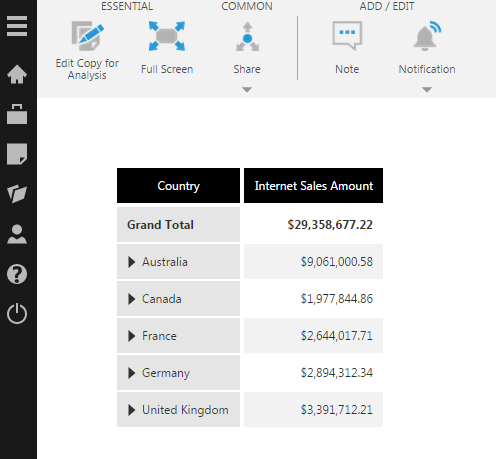
Log out of Dundas BI and log back on as UserA.

View **Dashboard1** and observe that it displays restricted data corresponding to SSAS Role A.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-view-usera.png)

Log out again and log back on as UserB.

View **Dashboard1** and observe that it displays restricted data corresponding to SSAS Role B.

[](https://www.dundas.com/support/images/dbi/support-articles/data-connectors/ssas-roles-view-userb.png)