Paper AD62

Secure SAS® OLAP Cubes with Top Secret Permissions

Stephen Overton, Zencos Consulting, Raleigh, NC, USA

ABSTRACT

SAS OLAP technology is used to organize and present summarized data for business intelligence applications. It features flexible options for creating and storing aggregations to improve performance and brings a powerful multi-dimensional approach to querying data. This paper focuses on managing security features available to OLAP cubes through the combination of SAS metadata and MDX logic.

INTRODUCTION

SAS Metadata allows organizations to control many aspects of OLAP security. Like many components registered in metadata, OLAP cubes reside within a metadata folder. Metadata permissions are inherited from parent folders or from explicit permissions set on the OLAP cube. Authorization permissions for OLAP Cubes can be managed generically in SAS metadata or within the OLAP cube on major components such as dimensions and hierarchies. You can also specify fine-grain permission conditions which filter specific slices of aggregate data the OLAP cube is built from using member-level security.

- **Authorization Permissions** – General access to cubes or major components of a cube (dimensions, hierarchies, levels, measures, calculated measures).
- **Member-Level Security** – Fine-grain access to members or slices of information driven from specific source data within a cube. Can utilize identity-driven permissions based on the end user querying the OLAP cube.

This paper assumes the reader has a basic understanding of OLAP technology and a basic understanding of SAS metadata security concepts.

AUTHORIZATION PERMISSIONS FOR OLAP CUBES

Authorization permissions are very analogous to the file permissions of SAS metadata. Authorization permissions can be set on the OLAP cube object at a high level, granting the ability to read or write to the cube as a whole, which is similar to basic folder permissions in metadata. Taking a step further, read access to components defined within the OLAP cube can be managed similar to managing explicit permissions on objects with SAS metadata folders.

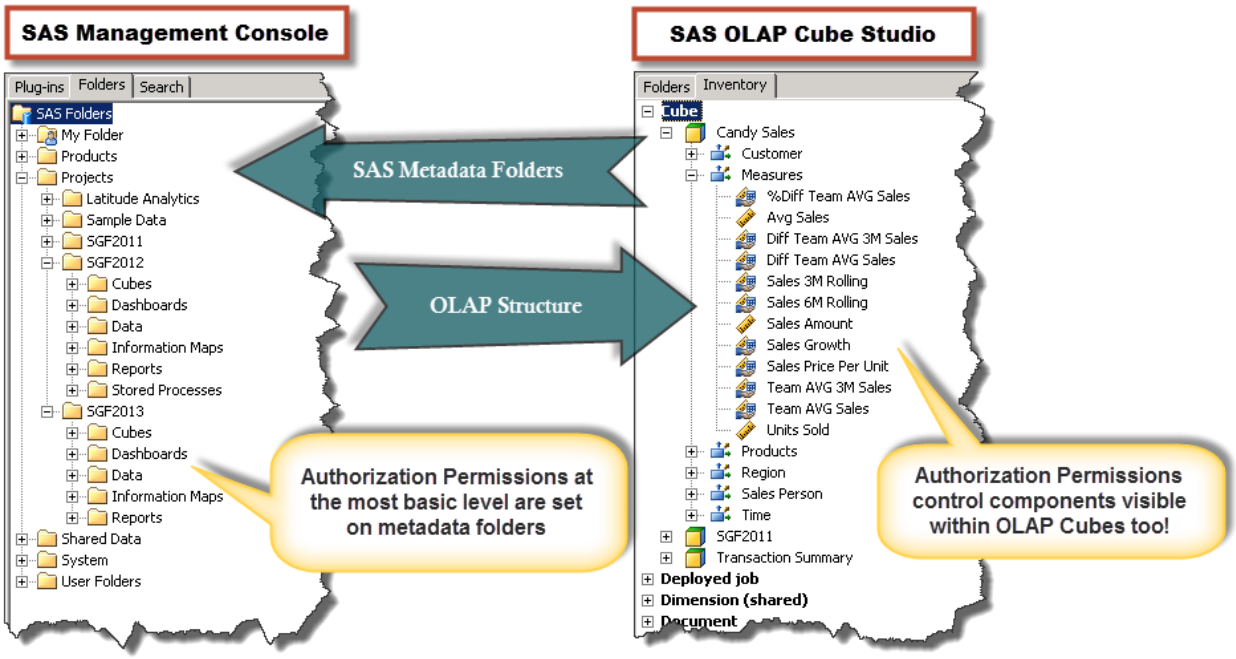

Figure 1: Authorization permissions compared to SAS Metadata folders.

AUTHORIZATION PERMISSIONS IN ACTION

The following example shows how two different roles can access different components within an OLAP cube. The Candy Sales cube will be referenced throughout this paper.

Basic User

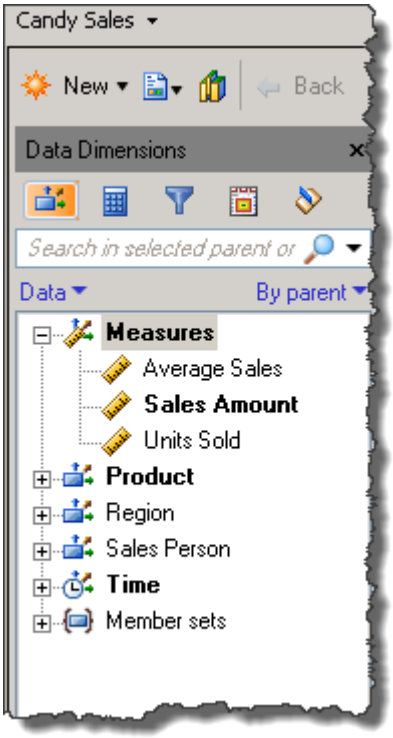

Figure 2: Basic user has access to simple measures and all dimensions except the Customer dimension.

Power User

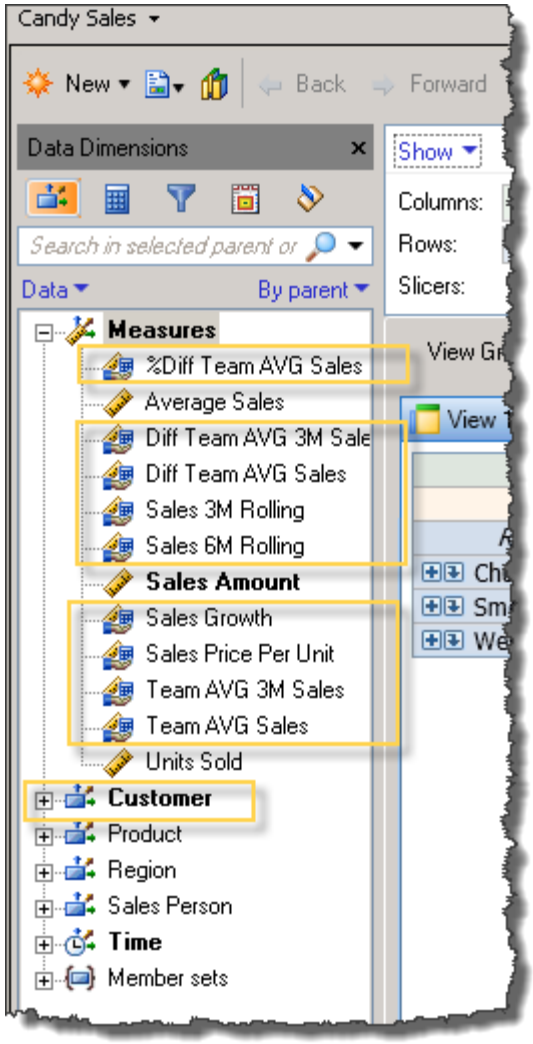

Figure 3: All dimensions and measures available to a power user within the Candy Sales cube.

As shown in Figure 3, the power user can access more complex measures and the entire Customer dimension. Any combination of permission patterns can be defined for different layers of users or groups to control access for different components within an OLAP cube. Access to the major components of OLAP such as dimensions, hierarchies, levels, measures, and calculated measures are controlled by SAS Metadata and the SAS OLAP Server. These components are managed using SAS OLAP Cube Studio® or SAS Management Console®. A SAS Platform Administrator is recommended to assist with, and most likely apply permissions to production environments.

HOW TO SECURE GENERAL ACCESS TO OLAP DIMENSIONS AND MEASURES

Continuing with the examples shown in Figure 2 and Figure 3, there are two types of users: users with full access by default, and users which are set as basic users that have limited access and capabilities.

As shown below in Figure 4 within SAS OLAP Cube Studio, right click a dimension, hierarchy, or a measure within the Measures tree, then select Properties. Select the Add button in the upper right corner of the properties window to add a SAS group to the Effective Permissions list for the dimension, hierarchy, or measure selected.

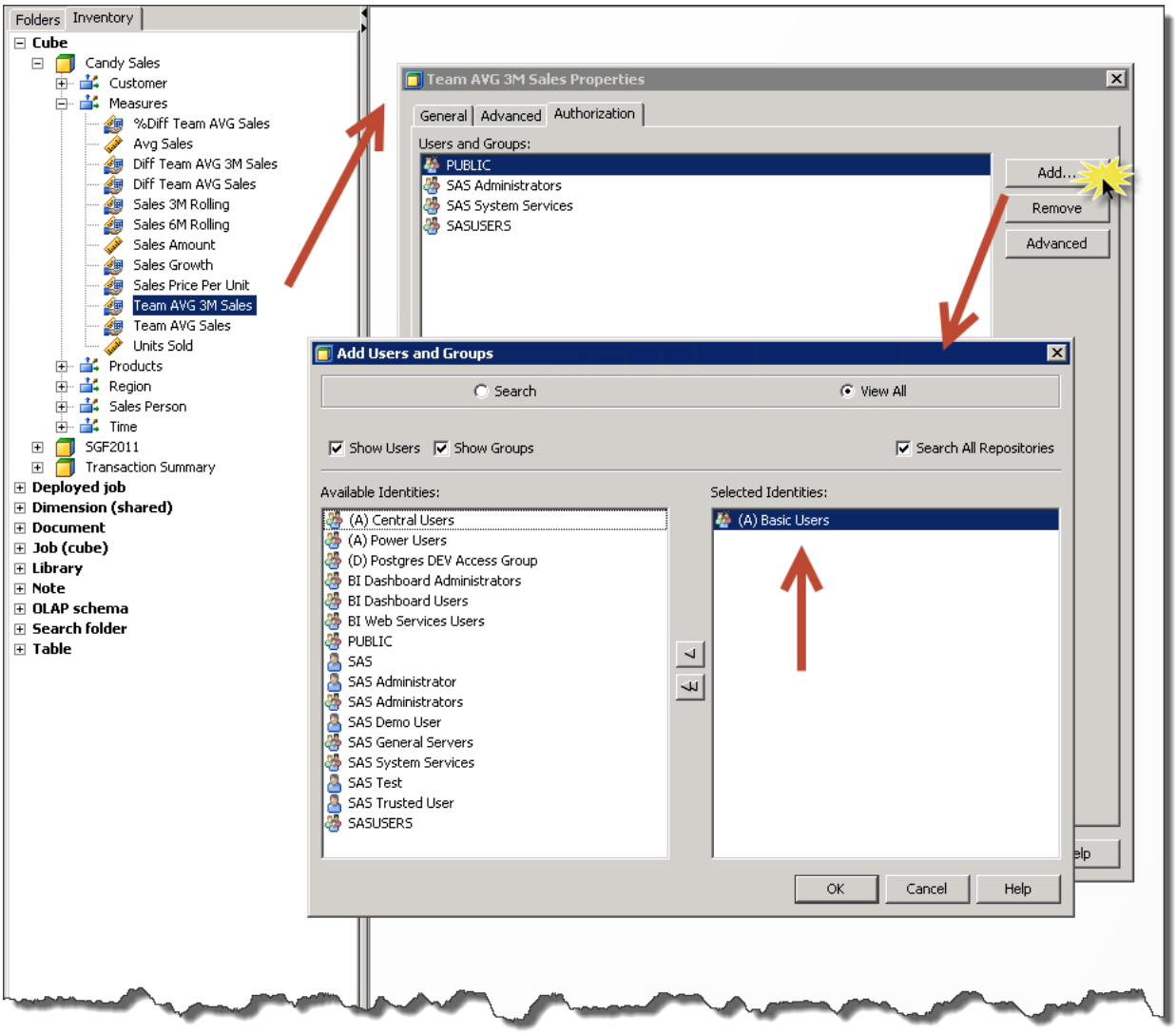

Figure 4: Applying Authorization Permissions to an OLAP measure.

Once the SAS group is added to the Selected Identities for this object, click OK to return to the properties window.

As shown in Figure 5, click Deny for each permission (ReadMetadata, WriteMetadata, Read) to restrict access to this measure.

These steps are taken to restrict access to these OLAP components in the Candy Sales cube:

- Team AVG 3M Sales
- % Diff Team AVG Sales (Calculated Measure)
- Diff Team AVG 3M Sales (Calculated Measure)
- Diff Team AVG Sales (Calculated Measure)
- Sales 3M Rolling (Calculated Measure)
- Sales 6M Rolling (Calculated Measure)
- Sales Growth (Calculated Measure)
- Sales Price Per Unit (Calculated Measure)
- Customer (Dimension)

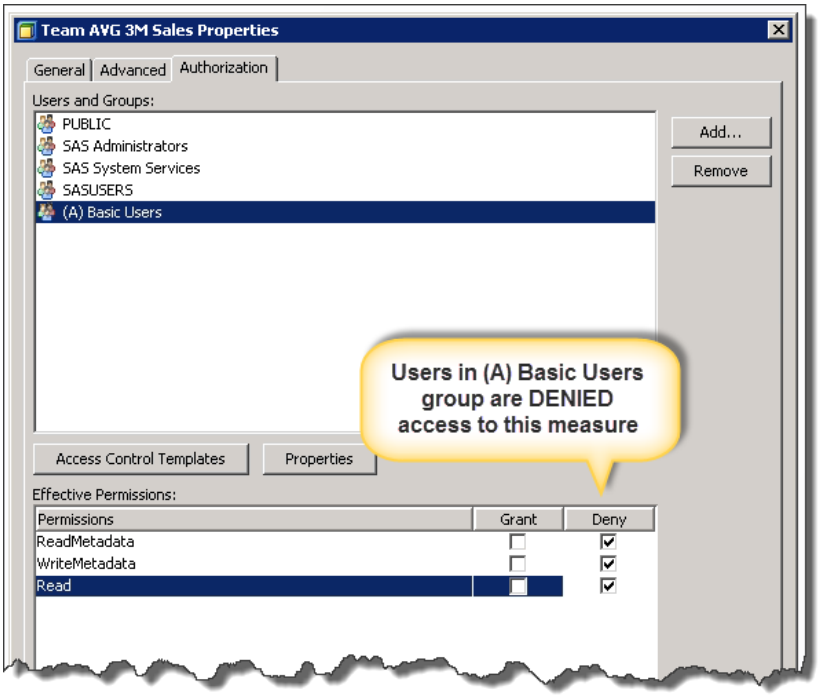

Figure 5: Applying DENY permission pattern to deny access to an OLAP measure.

Authorization permissions are the most basic way to secure an OLAP cube. It is analogous to securing SAS libraries, data tables, or folders in metadata. When planning the security of an OLAP cube, think of it as its own file system just inside the OLAP cube. The natural hierarchical organization of OLAP makes this easy to understand and build permissions around.

MEMBER-LEVEL SECURITY

Managing access to specific slices of data is accomplished with OLAP member-level security. Member-level security is useful for filtering slices of data within an OLAP cube, assuming the desired slice can be specified from a level in the OLAP cube. Each filter defined consists of an MDX expression that is used to subset at particular dimension of the OLAP cube. In other words, member-level security uses SAS metadata to automatically apply an MDX expression to the slicer axis of an MDX query. Since the member-level security uses MDX, the OLAP Server is used for enforcement at query time. Therefore, additional aggregation tuning may need to be considered if MDX is filtering a specific dimension.

IMPORTANT CONSIDERATIONS FOR OLAP MEMBER-LEVEL SECURITY

- Member-level permission conditions can only be specified on OLAP dimensions.
- The SECURITY_SUBSET option can affect results. When SECURITY_SUBSET=YES, totals and subtotals are recalculated at run time based on what members the requesting user can access. When SECURITY_SUBSET=NO, totals and subtotals are not recalculated and include all members in aggregate counts. This option is useful if you wish to restrict access to specific members but include the summarized counts in totals and subtotals.
- Members returned by the MDX expression must belong to the dimension on which the permission condition is defined. The returned set of members cannot be a union of members from other dimensions.
- A permission condition that filters a non-default hierarchy must include at least one member of the default hierarchy. If a requesting user does not have access to any members in the default hierarchy, then the query fails with a permissions error.

HOW TO SECURE ACCESS TO MEMBER-LEVEL DATA WITHIN AN OLAP CUBE

For example, data within the Candy Sales cube is filtered so that only a user associated with a particular region can access the aggregate data about that region. As shown below in Figure 6, within SAS OLAP Cube Studio navigate to the cube and right click the Region dimension to view the properties.

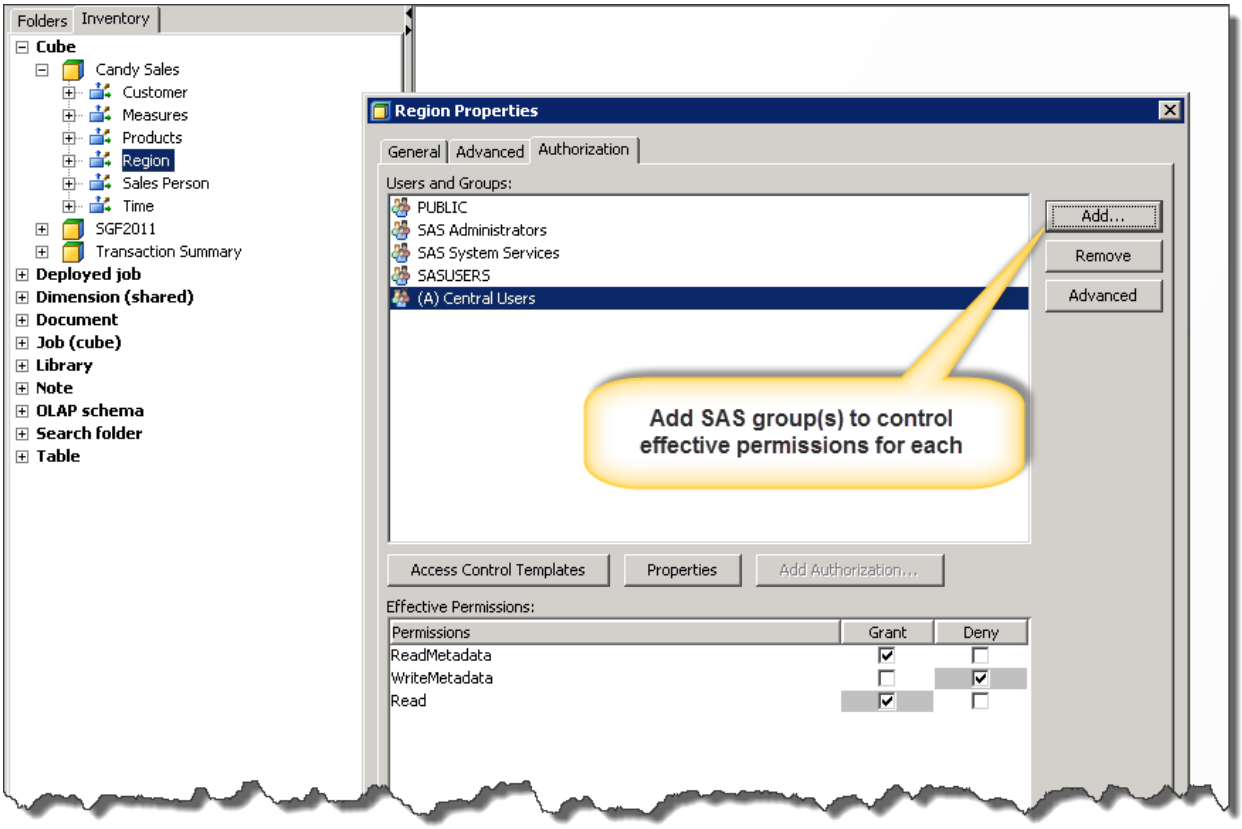

Figure 6: Applying Member-Level security to an OLAP dimension.

As shown in Figure 7, the (A) Central Users group is added to manage member-level security of users in this group. If the Add Authorization button is greyed out, set an explicit grant on the Read permission to enable it.

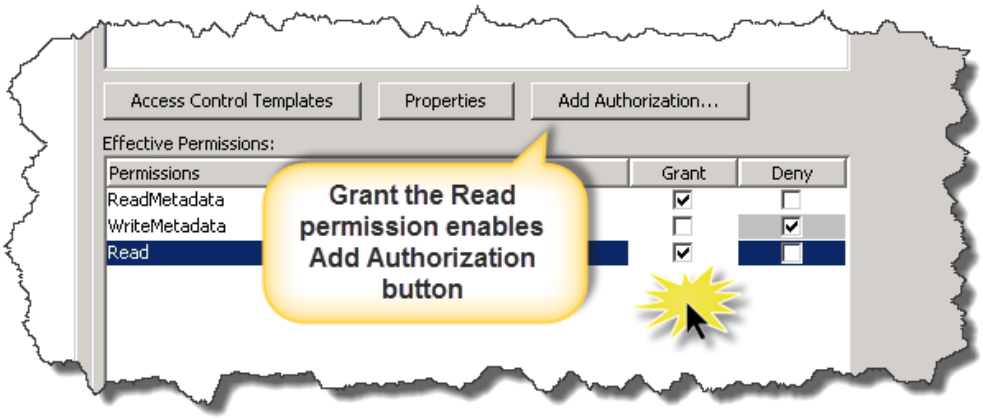

Figure 7: Applying permissions to add member-level authorizations.

Clicking the Add Authorization button allows further customization to the access levels within this dimension. As shown in Figure 8 below, a specific MDX expression is defined which filters data in the Candy Sales cube for users in the (A) Central Users group.

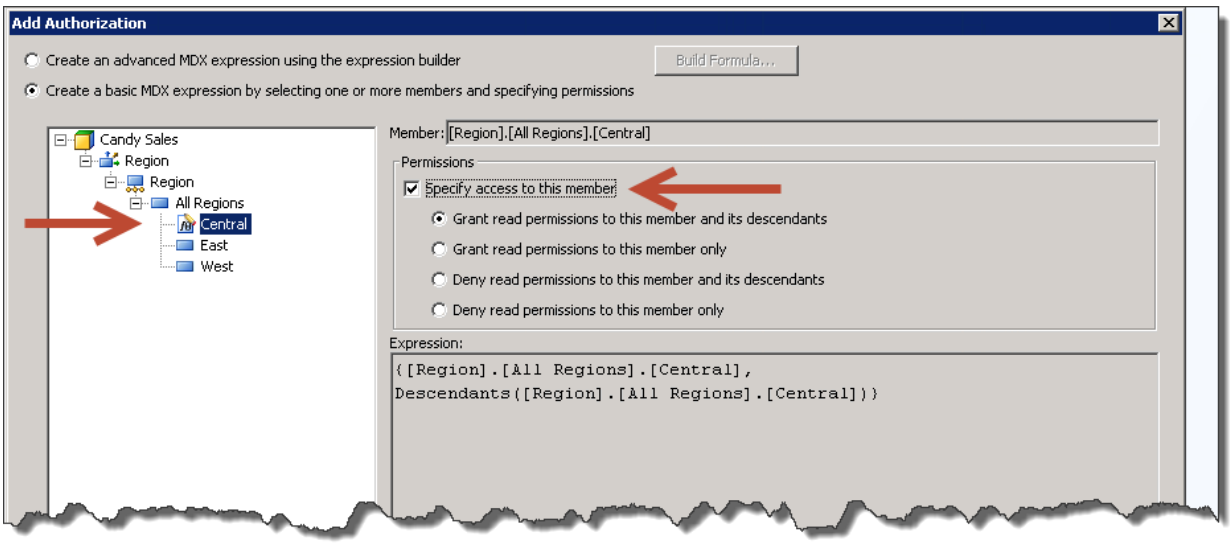

Figure 8: Add Authorizations to enable member-level security.

As shown in Figure 9, users that are members of the (A) Central Users group can only see data in the Central member of the Region dimension. Furthermore, regardless of whether the Central region is in the OLAP query output visually, the OLAP query will always be filtered for the Central region.

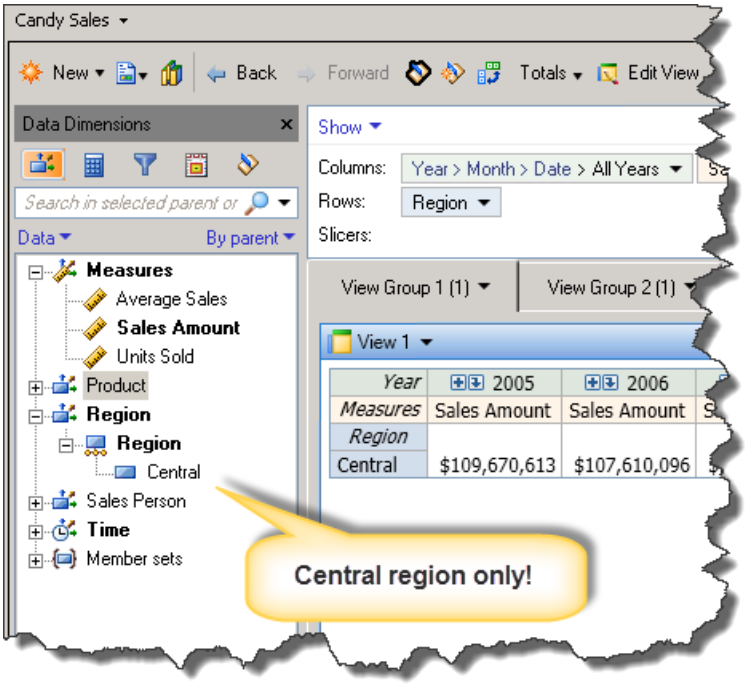

Figure 9: Member-Level security limits the visibility for specific users to only the Central region.

When the cube is accessed by a user who is a member of the (A) Central Users group, the MDX expression is automatically applied to the cube. This security model is reflected across the entire environment this SAS metadata server manages. Therefore, accessing this cube through applications such as SAS Web Report Studio® or SAS BI Dashboard® will also filter the Candy Sales cube.

IDENTITY-DRIVEN PERMISSIONS

Taking OLAP member-level security one step further, identity-driven permissions can be used to enforce access based on who a user is and what they can do. Identity-driven permissions derive user-specific items such as the user ID, name, or groups and assume these pieces of data are available in the OLAP cube. In other words, the user-specific data intended to be used to filter data within the OLAP cube must be an actual dimension of the OLAP cube.

IMPORTANT CONSIDERATIONS FOR IDENTITY-DRIVEN PERMISSIONS

While group membership information can be retrieved for the requesting user, it is generally easier to manage identity-driven security by focusing on a user ID since this uniquely identifies a person. If user or group information is used to filter OLAP data, careful designing and planning must be done to ensure the source data is structured accurately to support security requirements. For instance, if OLAP data is to be filtered for a specific user, the user name or ID must be defined as a level in an OLAP dimension. To make things more complex, the user ID or name level could potentially be defined across many hierarchies within the dimension. If one hierarchy is used to filter the cube, another hierarchy from the same dimension cannot be referenced in the output of the cube because the tuple is referenced on multiple axis.

- **Recommendation** – Try to keep things simple and design the source data so the accountable user ID or name is the only level related to the fact data.
- **Key point** – When working with many permission conditions, be conscious of other member-level security permissions which may restrict a portion or all data within another dimension that identity-driven permissions are intended to filter. In other words, watch out for overlapping filters!

HOW TO SECURE ACCESS TO OLAP DATA BASED ON USER IDENTITY

As shown in Figure 10, use the custom MDX expression editor to access Identity-Driven properties when adding an authorization to an OLAP dimension.

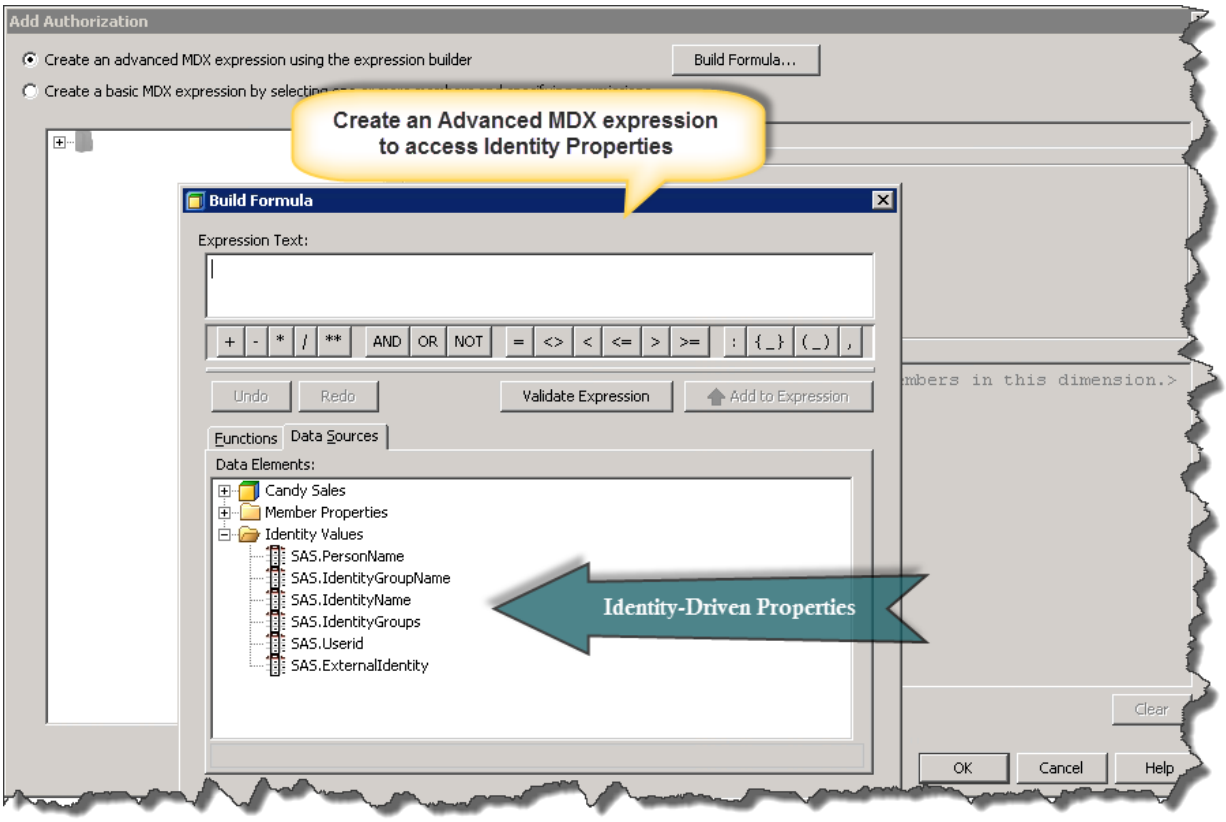

Figure 10: Using MDX expressions to create identity-driven permissions.

Extending the Member-Level security example in this paper, security can be simplified by defining a single permission condition which contains a dynamic reference directly to the user accessing the OLAP cube. This minimizes the number of group authorizations used to accomplish a single logical restriction. A single authorization can be defined which dynamically references the user's ID in metadata to filter OLAP data using a dimension which contains the user's ID. Before continuing, the Region member-level security was removed so that overlapping security policies would not affect this example. In this example, the User ID level is used to filter data based on who the user is.

As shown in Figure 11, from SAS OLAP Cube Studio select the properties of the dimension that contains the level that identifies the accountable user. In this example the Sales Person dimension is used. From here the (A) Basic Users group is added to define identity-driven permissions for users in this group. In this example, this enforces permissions for basic users only, and allows other advanced users full access. This security model is effective for the source data used in this example because user IDs are uniquely associated with each record.

Another way to think about this security model is that these are "Sales Person" user IDs. Therefore, it may be reasonable to assume that only sales persons need restrictions and upper management can have full access.

Add an explicit grant for the Read permission of the SASUSERS group. Next click the Add Authorization button to begin defining the custom MDX expression which limits members of the Sales Person dimension based on the requesting user accessing data from this cube.

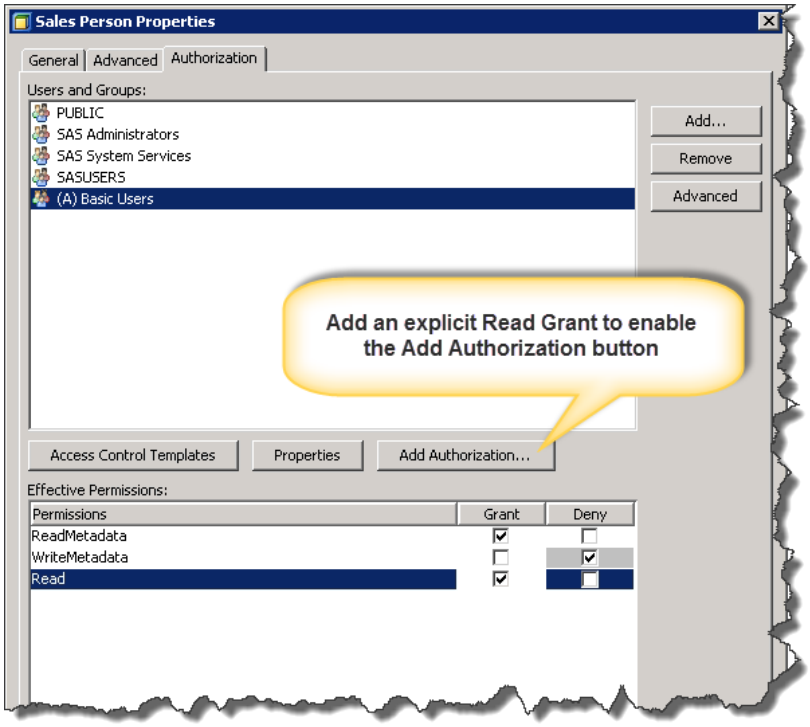

Figure 11: Adding a group to enable identity-driven permissions within an OLAP cube.

MDX is used to enforce limitations on the OLAP cube. As shown below in Figure 12, click the Build Formula button to start defining the MDX expression which is used by the OLAP server to filter data in the respective OLAP cube.

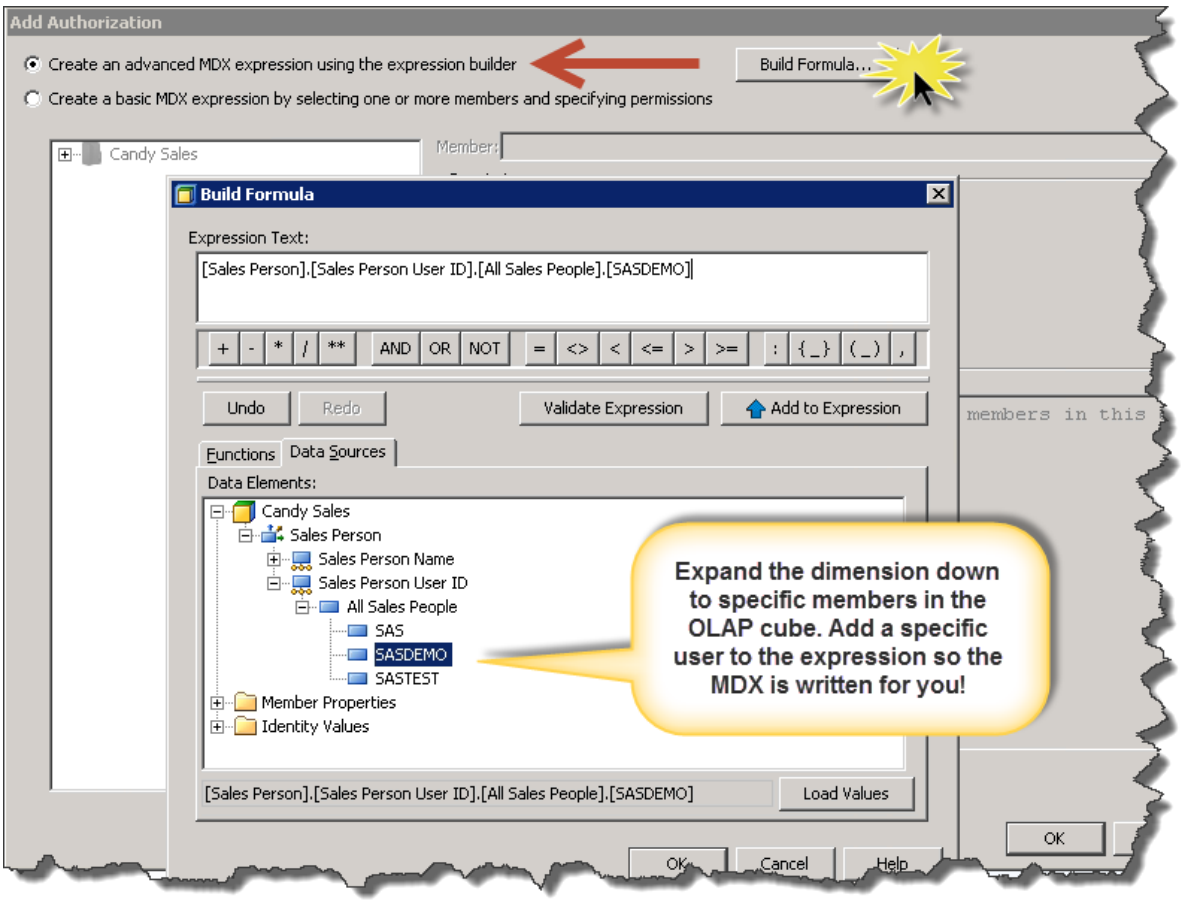

Figure 12: Building an MDX expression to enforce identity-driven permissions.

Within the expression builder, the easiest approach is to let SAS write the MDX for you. Drill down to the dimension → Hierarchy → Level, then double click (or click the Add to Expression button) any user ID to populate the expression with the appropriate MDX filter.

Best Practice – Make the identifying attribute (name or ID) uppercase in the source data to ensure a proper lookup.

As shown in Figure 13, replace the user ID at the tail end of the MDX expression, within the square brackets. Then expand the Identity Value folder and double click (or click the Add to Expression button) the SAS.Userid identity-driven property to replace the user ID within the MDX expression. Be sure to remove the double quotation marks. These are not needed and may be interpreted incorrectly as part of the actual member value.

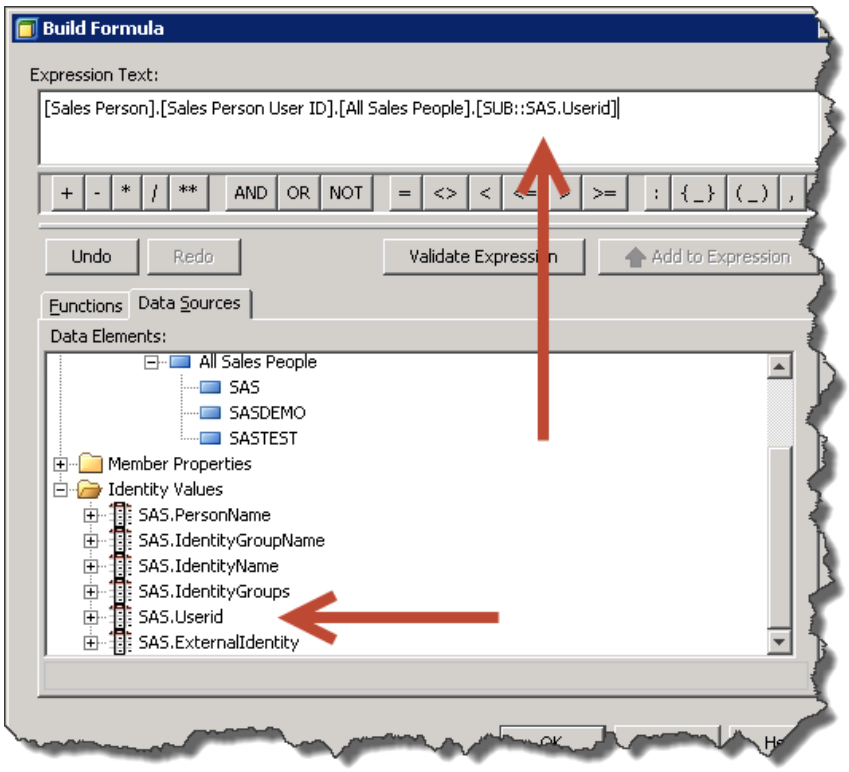

Figure 13: Enhancing MDX with dynamic identity values.

The full MDX expression defined will always be applied for users within the (A) Basic Users metadata group.

```
[Sales Person].[Sales Person User ID].[All Sales People].[SUB::SAS.Userid]
```

Is translated to the following at run time:

```
[Sales Person].[Sales Person User ID].[All Sales People].[SASDEMO]
```

As shown below in Figure 14, it is normal to see an error because the identity property is not looked up during the validation.

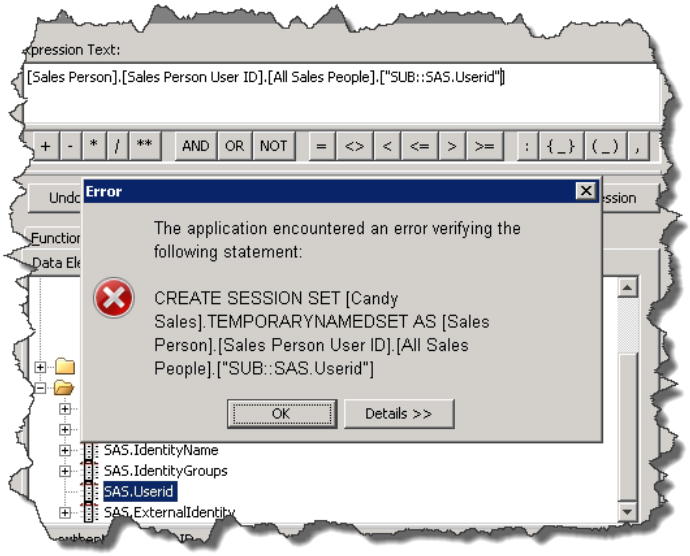

Figure 14: Error received in SAS OLAP Cube Studio when defining identity-driven permissions.

AUTHORIZATION AND MEMBER-LEVEL SECURITY PERMISSIONS IN ACTION

Figure 15 shows what components a user with limited capabilities can access based on the previous examples in this paper. This combines both authorization permissions and member-level security which uses identity-driven properties. Authorization permissions are used to restrict this user to only 3 basic measures and to hide the Customer dimension entirely. Identity-driven permissions are used to restrict the aggregate data available to only this specific user.

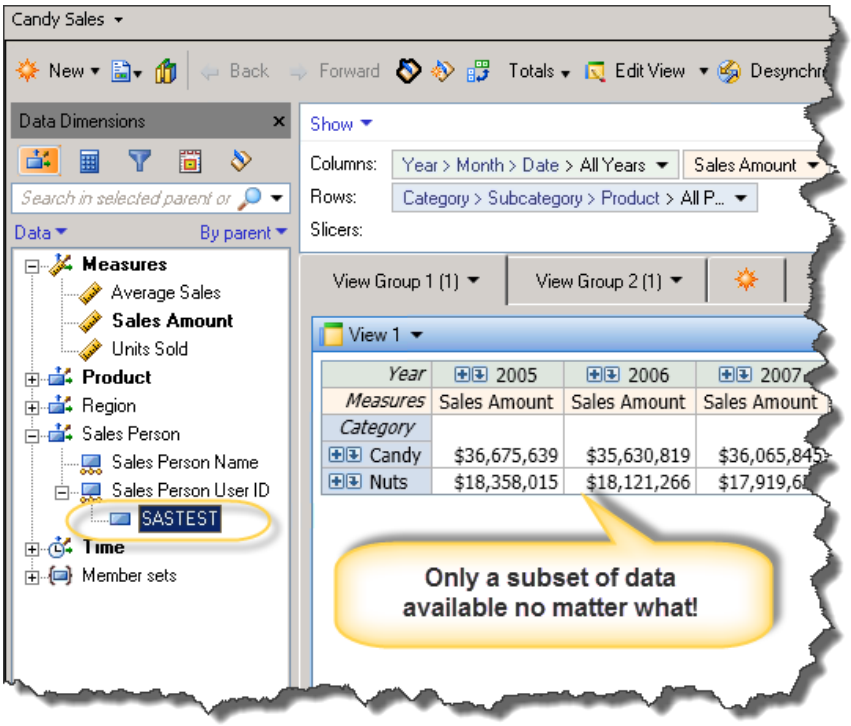

Figure 15: Authorization permissions limit measures and dimensions available in the Candy Sales cube. Identity-driven permissions limit the aggregate data available at query time based on the user accessing the cube.

RECOMMENDED READING

- *SAS 9.4 OLAP Server User Guide*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Stephen Overton
Zencos Consulting (http://www.zencos.com)

Email: soverton@zencos.com
Website: http://www.stephenoverton.net/
LinkedIn: http://www.linkedin.com/in/overton