Using MaxL Scripts to Create, Alter, & Grant Filters

Creating security filters and assigning them to different users/groups can be a time consuming process, especially if it is done manually. Luckily, there are some simple MaxL statements that can be used to significantly expedite the process. Here are the 3 that I've found to be most useful:

- Create Filter
- Alter Filter
- Grant Filter

Create Filter:

The MEMBER-EXPRESSION must be enclosed in single quotation marks. It can be a comma-separated list as well (this also pertains to the Alter Filter syntax). Notice in the example below how commas are used to separate 3 different dimensions (Year, Measures, & Product) in the create filter syntax:

- create filter Sample.Basic.filter1 read on '@IDescendants("Year"), @IDescendants("Measures"), @IDescendants(Product")';

For the FILTER-NAME portion, the application and database must be included preceding the filter name. This syntax will be used for Create, Alter, & Grant.

After running the batch, open EAS to verify that the filter was created correctly (I've included a generic version of my batch & MaxL files at the end of this post in case they may be helpful). Right click on the database and select Edit->Filters:

A list of all filters in the database will appear:

Select edit and the member specification assigned to the filter will pop up. All 3 dimensions that are outlined in the MaxL command should be accounted for:

Many times, the filter will need to be updated after it has been created. There is also a command line function for that…

Alter Filter:

For this example, we'll add another dimension into the filter. Let's add read access for @IDescendants("East"). Here's an example of the Alter Filter syntax:

- alter filter Sample.Basic.filter1 add read on '@IDescendants("East")';

After running the batch file, the filter now reflects the change that was made:

Now that the filter is built, it can be assigned to a user, group, or multiples of both using the Grant Filter command line function. However, prior to assigning a filter to a user/group, the user/group must be provisioned to have filter access to the application. This is done through Shared Services. We'll use "Test_User1" as a sample user. Right click on "Test_User1" and select Provision:

Expand down on the Sample application until Filter appears.

Highlight “Filter” and bring it across to the right side of the screen:

The selected roles should display “Filter” under Sample:

Click Save. Now, “Test_User1” is provisioned for the Sample application and the filter can be applied using the Grant Filter MaxL command.

Grant Filter:

Example of the Grant Filter syntax:

- grant filter Sample.Basic.filter1 to Test_User1;

To verify that “filter1” has been granted to “Test_User1”, head back to Shared Services and right click on Sample->Assign Access Control:

Select “User Name” from the dropdown menu in the top left and click search. Highlight “Test_User1” and click the right arrow to bring the user to the Selected box on the right. Click Next:

“Test_User1” has been granted “filter1” and the user’s access should reflect this change:

Batch File:

call MaxlPath “MaxL File Path” Sample Basic userID password ServerId filter_log

MaxL File:

login $3 $4 on $5;

spool on to “Log File Path”;

create filter Sample.Basic.filter1 read on ‘@IDescendants(“Year”), @IDescendants(“Measures”), @IDescendants(“Product”)’;

alter filter sample.basic.filter1 add read on ‘@IDescendants(“East”)’;

grant filter Sample.Basic.filter1 to Test_User1;

logout;

spool off;

exit;

To take a deeper dive into the filter functionality, or to clarify any issues, check out the Essbase Technical Reference:

https://docs.oracle.com/cd/E40248_01/epm.1112/essbase_tech_ref.pdf