SAS Metadata Querying and Reporting Made Easy: Using SAS Autocall Macros
Jiangtang Hu
d-Wise, Morrisville, NC

ABSTRACT
Metadata is the core of the modern SAS system (aka, SAS Business Analysis Platform) and SAS offers various techniques to access it via SAS data step functions, procedures, libname engines and Java interface. Furthermore, SAS also provides bunch of autocall macros which are well packaged for metadata querying and reporting using techniques above, and might be not well known for some SAS programmers.

In this paper, I will go through such metadata autocall macros to get quick results against SAS metadata like users, libraries, and most important, permissions. For best display of SAS metadata, SAS ODS Reporting Writing Interface technique is also used for this demo (again, it's not new and it's in SAS system folder of sample codes which are omitted by most SAS programmers). All demo codes can be found in Github, https://github.com/Jiangtang/SESUG. Metadata browsing configurations are also supplied for users of SAS Display Manager, SAS Enterprise Guide and SAS Data Integration Studio respectively.

INTRODUCTION
Suppose you got a new task, to produce a report that lists all grants and denies of all folders in a specific metadata server. This is important because the permission to study folders is key of metadata security. To get started, you can browse such information for a single folder in SAS Management Console (in 'Properties' then 'Authorization'):

<table>
<thead>
<tr>
<th>SAS Folders Properties (Read-Only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
</tr>
<tr>
<td>Users and groups:</td>
</tr>
<tr>
<td>SAS Administrators</td>
</tr>
<tr>
<td>SAS System Services</td>
</tr>
<tr>
<td>SAS Developers</td>
</tr>
<tr>
<td>PUBLIC</td>
</tr>
<tr>
<td>Apps Programmers</td>
</tr>
<tr>
<td>Clinical Administrators</td>
</tr>
<tr>
<td>IT Administrators</td>
</tr>
<tr>
<td>Access Control Templates</td>
</tr>
<tr>
<td>Effective Permissions:</td>
</tr>
<tr>
<td>Permissions</td>
</tr>
<tr>
<td>ReadMetaData</td>
</tr>
<tr>
<td>WriteMetaData</td>
</tr>
<tr>
<td>CheckInMetadata</td>
</tr>
<tr>
<td>Read</td>
</tr>
<tr>
<td>Write</td>
</tr>
<tr>
<td>Create</td>
</tr>
<tr>
<td>Delete</td>
</tr>
<tr>
<td>Administrator</td>
</tr>
</tbody>
</table>
Particularly, you need to produce a report like the following (this is for a single folder only):

**Metadata Security Report**

<table>
<thead>
<tr>
<th>Location</th>
<th>/</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Name</td>
<td>BIP Service</td>
</tr>
<tr>
<td>Public Type</td>
<td>RootFolder</td>
</tr>
<tr>
<td>Metadata URI</td>
<td>omsobj:SoftwareComponent/AEMTC9US:A00C001</td>
</tr>
<tr>
<td>Created</td>
<td>25Feb2012:23:59.04</td>
</tr>
<tr>
<td>Updated</td>
<td>28Jun2014:13:01:17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person Group</th>
<th>ReadMetadata</th>
<th>WriteMetadata</th>
<th>CheckInMetadata</th>
<th>Read</th>
<th>Write</th>
<th>Create</th>
<th>Delete</th>
<th>Administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In which,

- 🟢 means grant
- 🔴 means deny

The white box shows an explicit setting
The grey box shows the permission is inherited from its top level
The green box shows the permission is inherited from ACT

For this task, you need:

1. Get all the names of folders and subfolders recursively, with detailed metadata information like permissions, types, URIs.
2. Generate the report with the proper layout, icons and colors.

Now, for resources, how many hours do you plan to complete this task? I know it totally depends on the skill of individual programmers, but after going through this paper, I guarantee you can make it by 1 minute (the SAS running time not included of course!).

Also, this paper will go through the SAS autocall macros for other metadata querying tasks like libraries, users, authorization domains.

It begins with how to connect to SAS Metadata Server on your SAS programming platforms.
A. SET UP

A.1 SAS DISPLAY MANAGER USER
I prefer to use this traditional SAS interface to get my work done. It’s quick and easy for interactive debugging. To get it connected with a SAS Metadata Server, go to “Solutions” -> “Accessories” -> “Metadata Browser” and type in such information regarding the server you play with:

Save it and let it go. Next time when you submit codes with metadata requires, this window will pop up for your confirmation. You don’t need to type these metadata server connection codes in your programs.

A.2 SAS ENTERPRISE GUIDE USER
If you’re the SAS Enterprise Guide user, the metadata connection profile was probably ready for you. If not, go to “Tools” -> “Options” -> “Administration” -> “Connections”, and supply the information similar above:
And you’re ready to go.

A.3 SAS DATA INTERGRATION STUDIO USER

SAS Data Integration Studio users don’t need to do anything additional to connect to the SAS Metadata Server as long as you are in an active SAS Data Integration Studio session. You can either place a SAS Code node to your job or simply open a (very minimal) SAS session (“Tools” -> “Code Editor”).

A.4 GITHUB (OPTIONAL)

All the demo codes are hosted in Github. To best get use of it, you might think about to create a Github account (it’s free). To get started, check this paper out:

*The Hitchhiker’s Guide to Github: SAS Programming Goes Social*

B. SAS AUTOCALL MACROS FOR METADATA MANAGEMENT

B.1 PERMISSIONS QUERING AND REPORTING

The 1-minute solution of the open question above is:

```
*step 1/2: get the permission data;
%mdsecds(folder="/",includesubfolders=YES,membertypes="Folder")

*step 2/2: get the report (with the proper layout, icons and colors!);
%secrpt(permdata=mdsecds_join)
```

The first macro, %mdsecds, is located in one of the SAS autocall macro folders (available for SAS 9.*):
along with other metadata related macros with names beginning with “md” and some deprecated ones beginning with “oma”.

The metadata security reporting macro, `%secrept`, is in a sample folder. In my current session, it’s

```
C:\Program Files\SASHome\SASFoundation\9.4\core\sample
```

 `%secrept` is not an SAS autocall macro, but we can easily make it by copying it to any SAS autocall macro folder. Remember that you do need do a minimal modification to make this ‘sample’ macro to work:

You can simply delete such metadata server connection lines since you’re already connect to the server by methods descripted in session A. SAS ODS Reporting Writing Interface technique is used in this macro.

**B.2 ADDITIONAL NOTES ON METADATA PERMISSION MACRO**

The macro used above, `%mdsecds`, has 4 underlying macros to support it:

1. `%mdsecco`: get objects
2. `%mdsecgp`: get permissions
3. `%mdsectr`: transformation
4. `%mdsecvw`: create view for final output dataset, `mdsecds_join`.

A best diagram of the program flow is from SAS official documentation, *SAS(R) 9.4 Intelligence Platform: Security Administration Guide*:
Note %mdsecgo is a very versatile macro for metadata querying. You can assign multiple metadata types like Folder, Library, OLAP Schema, Stored Process and etc. to fit your needs:

```
%mdsecds (membertypes="Library");
%mdsecds (membertypes="Folder");
%mdsecds (membertypes="StoredProcess");
%mdsecds (membertypes="OLAPSchema");
%mdsecds (membertypes="Report");
```

where the parameter “membertypes” is ‘TypeName’ showed in the following picture (SAS Management Console - >"Folders" ->"System" ->"Types"). For example, membertypes="Folder" is Tree as ‘MetadataType’.
B.3 USERS

To get detailed metadata information against users, the one line of code you need is

```plaintext
*get user and group metadata;
%mdueextr(libref=work)
```

then you will get a very rich set of information in the library you assigned including:

- Authorization domains (got by supporting macro `%mduadms`)
- Emails, phones, locations
- Persons, logins and groups, etc.
For your information, these macros are used for user related task (import and export):

1. mdumap
2. mduimpc
3. mduimpl
4. mduimplb
5. mducmp
6. mduchgl
7. mduchglb
8. mduchgv
9. mdufoobjs

CONCLUSION

The SAS autocall macros for metadata querying are far away full replacements of the established techniques like SAS data step functions, procedures and Java interface, but they can serve as good start points for beginners to get quick results. To leverage the power of SAS autocall macros, readers are strongly recommended to check out the macro folders shipped with SAS products.

REFERENCES

   http://analytics.ncsu.edu/sesug/2013/PA-04.pdf

   http://support.sas.com/documentation/cdl/en/bisecag/67045/HTML/default/viewer.htm#p1h2c11xffn6xcn1gg9adnt7yhrb.htm
ACKNOWLEDGMENTS

Thanks my d-Wise colleague John Leveille to bring me to the SAS Metadata world.

CONTACT INFORMATION

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

Jiangtang('JT') Hu
Life Sciences Consultant
d-Wise Technologies, Inc.
1500 Perimeter Park Dr., Suite 150, Morrisville, NC, 27560
www.d-Wise.com

(O) 919-334-6096
(C) 919-801-9659
(F): 888-563-0931
(O) Jiangtang.Hu@d-Wise.com
(P) jiangtanghu@gmail.com
http://jiangtanghu.com/

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.