Paper 2060-2015

Automating Your Metadata Reporting

Rupinder Dhillon, Bell Canada

Darryl Prebble, Prebble Consulting Inc.

ABSTRACT

Keeping track of which of your SAS® Data Integration Studio (DI Studio) programs are updating which tables, or even where your programs are stored can be a full-time job in itself. Even the best of intentions tend to fall out of date the moment things get organized and deployed. What if there was a way to automate your metadata reporting to keep it current, and also create an audit trail of history while you're at it?

INTRODUCTION

DI Studio stores all of the information about your program on the SAS® Metadata Server. Using a variety of tools at our disposal we can leverage that information to build an easy-to-query table that any user can leverage for information about your programs. This information can include what tables are used in the program, who last updated the program and where the program is stored. In a small shop this may not pose much of a problem, but once your programs number in the thousands this can be a very useful tool!

Note that SAS 9.4 does offer a wider arrange of pre-built macros and tools that will pull useful information from the Metadata server for you. These are not discussed here; instead we will focus on how to query raw metadata in an effort to present the most flexible and robust Metadata querying solution that you can apply to any reporting requirement. We have also tried to keep the approach general so that it is applicable to all versions of SAS that use the SAS Metadata server.

We will be using the Metadata Browser in PC-SAS® (9.4), the SAS® XML Mapper (9.4), and SAS® Enterprise Guide (6.1) to illustrate how to query the Metadata Server and then store the results in a useful manner.

SAS METADATA 101

SAS Metadata was first introduced in SAS 9. All of a sudden, we could store information about our data, our users, our servers and our processes that would help manage how all components of SAS were used and worked together. Metadata will tell us:

- Who are your users and what can they access?
- Where is your data and how can you connect?
- What SAS processes exist and what data do they use?
- Who created my SAS processes and how can I run them?

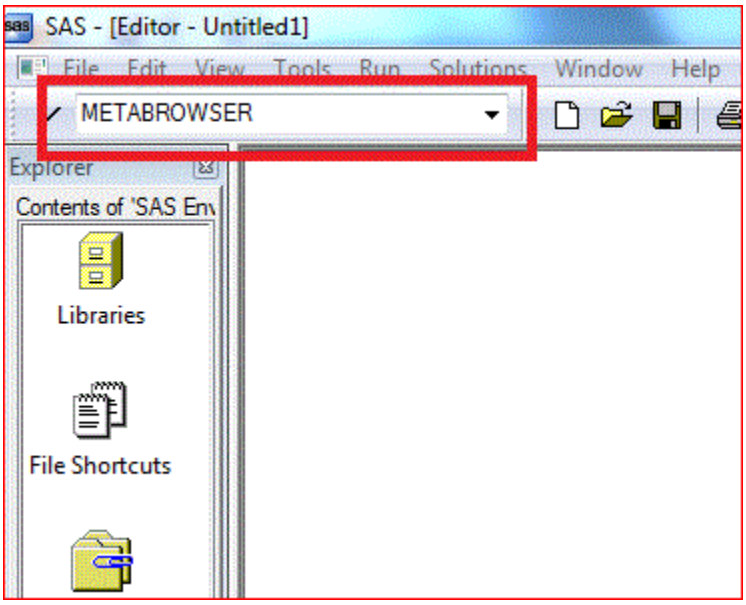
This "data about data" is stored in repositories within the SAS Metadata server and makes up the most important component of today's SAS platform. You can imagine that there is a wealth of information about your SAS system here; the challenge is how to get that information out.

PULLING AND INTERPRETING METADATA

METADATA BROWSER

You will need to plan out what it is that you are looking to pull from the SAS® Metadata Server – it unfortunately does not store things in a simple table format. To take a look at the metadata, you'll need to open up a PC-SAS instance and type 'METABROWSER' into the command line as shown below in Figure 1.

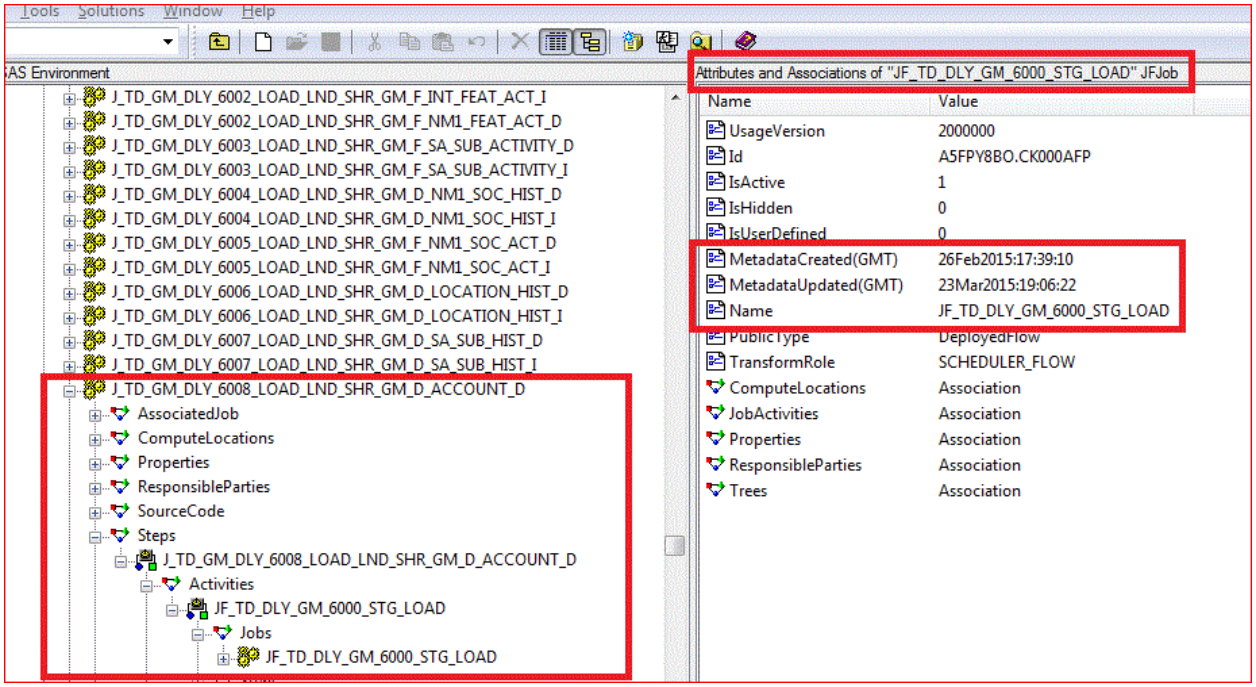
Figure 1
Opening the Metadata Browser in PC-SAS



You will be prompted for the name of your server and your credentials, so make sure you've noted those ahead of time.

For our example we are going to look at some Jobflow information. Expanding the JFJob tree, we pick a job that we are familiar with so that we can map out the information that we are looking for. In Figure 2, we have picked a job and expanded it out until we can see all of the attributes that we are interested in.

Figure 2
JFJob Attributes



Note that we can now see that the job (J_TD_GM_DLY_6008_LOAD_LND_SHR_GM_D_ACCOUNT_D) is attached to a flow (JF_TD_DLY_GM_6000_STG_LOAD). We can build an XML pull based on this information, knowing how each of the nodes is connected to the other as we drill down through the GUI:

```
filename outxml "/data/mi/data/outbox/cntl_dis_meta/job details report.xml";

proc metadata
in='
<GetMetadataObjects>
 <Reposid>$METAREPOSITORY</Reposid>
 <Type>Job</Type>
 <Objects/>
 <NS>SAS</NS>
<Flags>388</Flags>
<Options>
 <Templates>

 <Job Id="" Name="">
        <JFJobs/>
</Job>
        <JFJobs>
                <JFJob/>
        </JFJobs>
                <JFJob Id="" Name="" MetadataCreated="" MetadataUpdated="">
                        <Steps/>
                </JFJob>

... CONTINUED ...

</Templates>
 </Options>
</GetMetadataObjects>'

 out=outxml
 ;
run;
```

Note the section in yellow above. This part is somewhat generic – the meaning of which is outside the scope of this paper. If we focus on the code in-between the yellow highlighted parts, you can see that we move sequentially from Job -> JFJobs, until we reach the JFJob object where we find some attributes that we're interested in. More attributes are readily available as you drill down further – this paper leaves it to you to dig a little deeper.

There is a pattern of opening an object and subsequently closing that object that you will follow through the coding of the data pull. Notice that we open the Job object (ie. <Job …) and specify the Id and Name values as attributes that we want to pull. Then we move down the line opening up JFJobs and subsequently closing Job (ie. </Job>). This pattern repeats as we drill deeper.

After running this code, we've got a raw text file filled with XML code detailing our jobs and jobflow details… How do we interpret this XML file?

XML MAPPER

We need to create an XML Map so that SAS® will know how to interpret the file we've created. This is quite easy using the SAS® XML Mapper 9.4 tool. Open the program, go to File -> Open XML, then select your output file from earlier.

Beware of special characters – if the import errors out, you can look at the log to find out what line/column is causing the issue. Many times this can be due to a special character that simply needs to be removed to facilitate the creation of the map.

Once the file has been imported, go to Tools -> AutoMap using XML. Last, you need simply to save the map that you've created by going to File -> Save XMLMap.

INTERPRETING THE XML FILE USING SAS® ENTERPRISE GUIDE

Now we are set to read in the XML file using our XML Map.

```
filename  SXLEMAP '/data/outbox/cntl_dis_meta/job details report.map';
libname   outxml xmlv2 xmlmap=SXLEMAP;

proc sql;

* Read in XML data (only 5 connections available at a time otherwise) ;
create table job       as select * from outxml.job;
create table jfjobs    as select * from outxml.jfjobs;
create table jfjob     as select * from outxml.jfjob;


create table jobflow_details as
select jf.id           label='' as JF_METADATA_ID
,jf.name               label='' as JF_NAME
,jf.metadatacreated - 14400   format=datetime26. label='' as JF_METADATACREATED_DTTM
,jf.metadataupdated - 14400   format=datetime26. label='' as JF_METADATAUPDATED_DTTM
,job.id                label='' as JOB_METADATA_ID
,job.name              label='' as JOB_NAME

from job job

left join jfjobs jfs
       on(
              jfs.job_ordinal = job.job_ordinal
              )
left join jfjob jf
       on(
              jf.jfjobs_ordinal = jfs.jfjobs_ordinal
              )
```

As discussed previously, due to the drill-through nature of the data the output is created in a format that forces you to join the object tables (each object has it's own table) together once you've read it in. Each table will contain a primary key and a foreign key. You can follow the bouncing ball if you will, from one table to the next until you've joined everything together into a master table.

(Note: the datetime manipulation above was simply to convert to EST.)

Figure 3

Example result set

Answerset 1

	JF_METADATA_ID	JF_NAME	JF_METADATACREATED_DTTM	JF_METADATAUPDATED_DTTM	JOB_METADATA_ID	JOB_NAME
1	A5FPY8BO.CK000AFP	JF_TD_DLY_GM_6000_STG_LOAD	2/26/2015 13:39:10	3/17/2015 11:52:12	A5FPY8BO.C0000DPO	J_TD_GM_DLY_6008_LOAD_LND_SHR_GM_D_ACCOUNT_D

APPLICATIONS

There are countless reporting and general applications that you can build to leverage SAS metadata. Here are some that we have been able to build and deploy.

- Job Changes Report: Daily tracker of which new jobs have been deployed and which existing jobs have been updated. This allows us to not only track productivity but also compliments the Change Control process
- Job Flow/Job/Table Inventory: Automated way of tracking which job and job flow updates which set of tables.
- Master Inventory Migration: This has been vital to our 9.2 to 9.4 migration project. It allowed us to see which jobs were actively deployed in a flow while others may have been decommissioned from the active job flows but not moved to archive folders. It also allowed us to systematically pull a full inventory of Metadata objects that we would need to migrate; allowing us to build our initial migration plan and track progress.
- Keeping track of managing Users and Groups
- Updating existing Metadata and creating new Metadata: YIKES, why would you want to do that programmatically? A dangerous proposition but could have its uses when you consider needing to update a large group of users, or update the metadata for a large set of tables within a single library. This is not something that we've had to try; we prefer the more systematic Metadata management solutions offered through the GUI's (SMC and DIS).

These are just some ideas of the types of reports that you can create by tapping into the metadata in an automated way. To get a better idea of the other types of reports that can be created, the best place to start would be to understand the Metadata Object types that are stored in the Metadata Server. Once your code is ready, it can be deployed and scheduled so that the reporting is completely automated.

Our next challenge will be to use Metadata and login information to get a better understanding of active SAS usage among our users. We want to be able to see who our active SAS users are; what are they using and how often? What are our most valuable SAS assets and how far reaching is our SAS platform across the organization? Stay tuned…

CONCLUSION

Though just an introduction to what you can do, clearly there is a wealth of information available to you in the metadata that is just waiting to be tapped. You can build historical tables tracking changes over time,

keep tabs on what tables are being updated by what jobs, or quickly see which flows reference a specific job.

REFERENCES

Jugdish Mistry, J2L Limited, Amersham, UK, SGF 2013 “SAS® Metadata Reporting: Extracting Information from SAS Metadata”. *SAS Global 2013 Conference*

Available at http://support.sas.com/resources/papers/proceedings13/518-2013.pdf

SAS, Cary NC. “Extending the Metadata Security Audit Reporting Capabilities of the Audit and Performance Measurement Package.” SAS Enterprise Excellence Center

Available at https://support.sas.com/rnd/emi/docs/Ebiapm92.Metadata.Security.Audit.Reporting.pdf

Michael Kilhullen, SAS Institute Inc., Cary, NC. “Developing Custom Metadata Reports for SAS® Data Integration Studio Michael Kilhullen”, *SAS Global 2012 Conference*

Available at http://support.sas.com/resources/papers/proceedings12/120-2012.pdf

RECOMMENDED READING

- *SAS® 9.4 Metadata Model: Reference*
- *SAS® Open Metadata Interface: Reference and Usage*
- *SAS® 9.4 Language Interfaces to Metadata*

CONTACT INFORMATION

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

Rupinder Dhillon
Bell Canada
Rup45.dhillon@gmail.com

Darryl Prebble
Prebble Consulting Inc.
dprebble@gmail.com