

Paper 1275 – 2017

Read SAS® Metadata in SAS® Enterprise Guide

Piyush Singh, Prasoon Sangwan, TATA Consultancy Services Ltd.
Steven D Randolph, Eli Lilly and Company

ABSTRACT

SAS® Management Console has been key tool to interact with SAS® Metadata Servers. But sometime users need much more than what SAS Management Console can do. This paper contains couples of SAS macros which can be used in SAS® Enterprise Guide/ PC SAS® to read SAS® Metadata. These macros read users, roles and groups registered in metadata. SAS macros given in this paper explain how these macros can be executed in SAS® Enterprise Guide and how to change these macros to meet other business requirement. There may be some tools available in the market, which can be used to read SAS® Metadata but this paper helps in achieving most of them within SAS client like PC SAS and SAS Enterprise Guide by avoiding any additional plugin.

INTRODUCTION

SAS Management Console is one of the key and primary tool for SAS Administrators. The activities like add new user, access removal, group creation, ACT etc., there are so many administrative activities which administrators do with SAS Management Console. This provides a centralized user interface which allow to manage entire SAS Metadata for SAS Environment. This is completely made to manage the metadata repository for SAS environment.

But in most of the cases end users don't get access to SAS Management Console. This is more administrative tool instead of user tool. But what if user want to see their own access? What if they want to know what kind of access they have been granted? Do you want, your users to contact administrator for each and every thing?

This paper explains couple SAS macros, which help end users to get their SAS Metadata privileges. This helps you to provide user control to read metadata and reduce the administrator dependency. Paper describes SAS macros which can be run by users themselves and get their access related information. SAS macros given in this paper also helps administrators to create different report from SAS metadata directly, which is not directly available with SAS Management Console.

This paper explains two important SAS macros:

- List Registered Users in SAS Metadata
- List Existing Roles in SAS Metadata

MACRO TO READ SAS METADATA USERS

Below is the first macro to read the SAS Metadata Repository. If you need to create the list of all registered users from SAS Metadata, then you can run this macro with required parameters (macro variables as per your SAS environment) in your SAS Enterprise Guide Session.

Below SAS code can be changed as per your coding expertise. For example, metadata port in below code is hard coded and if you need to run the macro with different version of SAS then you need to change the port in SAS code. But if you want, you can make it macro variable as we have done for metadata server name and other parameters. Following parameters needs to change before running the macro:

Metadataserver – This is the name of SAS Metadata Server, which you want to read the users from. This is the server name where your Metadata is installed for your SAS environment.

Metauser – This is the user id who is running the below SAS code. For example, SAS administrators are the one who have access to read all users registered in metadata. Not everyone can read other's access information. If you are end user and want to read your metadata information, you can run below code for you (fourth point explain, what we mean by "for you").

Metapass – This is the password of metauser, who is running the below macro. Administrator may need to run this macro with sasadm@saspw (unrestricted id) to get complete information. It totally depends on your need and requirement.

User – This is user's system id, which you want to search. As per above second point, if you are end user and you want to run this for you, you can give your system id. Administrators can give any user id to check if that user is registered in SAS metadata or not.

Syntax – There are different ways to call the macro. You can modify the code, if you want to add more or less parameters. Few syntaxes are given below:

- Check a specific user's access:

```
%metauser (metaserver=<metadata_server_name>, user=<specific_user>);
```

- List of all metadata registered users:

```
%metauser (metaserver=<metadata_server_name>, user=<all>);
```

- If you need to run from other administrative id like sasadm@saspw

```
%metauser (metaserver=<metadata_server_name>, metauser=<sasadm@saspw>,
metapass=<Administrator_Password>, user=<specific_user>);
```

SAS Code

Below the full code for metauser macro:

```
%macro metauser (metauser=, metapass=, user=);

%let metaserver = %str(<metadata_server_name>);
options metaserver="&metaserver"
metaport=<PORT>
metauser="&metauser"
metapass="&metapass"
metarepository="Foundation"
metaprotocol=BRIDGE;
ods listing close;

data metauser_info;
length uri name DisplayName group groupuri $256;
call missing (uri, name, Displayname, group, groupuri);
n=1;

nobj=metadata_getnobj ("omsobj:Person?@Id contains '."',n,uri);

if nobj=0 then
    put 'No User';
else
```

```

do while (nobj > 0);
rc=metadata_getattr(uri, "Name", Name);
rc=metadata_getattr(uri, "DisplayName", DisplayName);

a=1;
grpassn=metadata_getnasn(uri,"IdentityGroups",a,groupuri);
  if grpassn in (-3,-4) then
    do;
      group="No groups";
      output;
    end;

    * list all group associated with user*;
  else
    do while (grpassn > 0);
      rc2=metadata_getattr(groupuri, "Name", group);
      a+1;
      output;
      grpassn=metadata_getnasn(uri,"IdentityGroups",a,groupuri);
    end;

    n+1;
    nobj=metadata_getnobj("omsobj:Person?@Id contains '."',n,uri);
end;

keep name DisplayName group;
run;

data user_id;
length uri name identity LoginObjId Identitytype identuri $256 UserId $50 Id
identityid $ 17;
call missing (uri, Id, name, identity, LoginObjId, identuri, UserId,
Identitytype,identityid);
n=1;
nobj=metadata_getnobj("omsobj:Login?@Id contains '."',n,uri);

if nobj=0 then
  put 'No Persons available.';
else
  do while (nobj > 0);
    rc=metadata_getattr(uri, "UserId", UserId);

    a=1;
    grpassn=metadata_getnasn(uri,"AssociatedIdentity",a,identuri);

    if grpassn in (-3,-4) then
      do;
        identity="No associated Identity";
        output;
      end;
    else
      do while (grpassn > 0);
        identrc=metadata_resolve(identuri,Identitytype,identityid);
        rc2=metadata_getattr(identuri, "Name", identity);
        a+1;
        output;
        grpassn=metadata_getnasn(uri,"AssociatedIdentity",a,identuri);

```

```

        end;

        n+1;
        nobj=metadata_getnobj("omsobj:Login?@Id contains '."',n,uri);
end;

keep identity UserId;
run;

data member (rename=(identity=name));
set user_id;
run;

proc sort data=metauser_info;
by name;
run;

proc sort data=member;
by name;
run;

data metauser_info2 (keep= name UserId DisplayName group);
merge metauser_info (in=in1) member (in=inr);
by name;
    if in1;
    if UserId= ' ' then
        UserId='NotRegistered';
run;

%if %lowercase("&user") eq "all" %then
    %do;
        data metauser_info2;
        set metauser_info2;
        run;
    %end;
%else
    %do;
        data metauser_info2;
        set metauser_info2;
        if UserId="&user";
        run;
    %end;

proc print data=metauser_info2;
by name;
run;

%mend metauser;

%metauser(user=all);

```

As we discussed earlier, you can change the required parameters in the code as per your need. You can further enhance this macro, but current macro helps to read user's name, display name, associated groups. IF you want to add more parameters in report, you can add those parameters like when users were added to metadata.

Below is the sample output from running the macro to get the user information:

----- name=devnz12 -----			
Display			
Obs	Name	group	UserId
1	NetezzaD	Netezz_d	nza425
2	NetezzaD	Netezz_a	nza428
3	NetezzaD	Netezz_f	nza429
----- name=testora -----			
Display			
Obs	Name	group	UserId
4	OracleDB	Oracle_d	orat58
5	OracleDB	Oracle_a	orat59
6	OracleDB	orat60_f	orat60

Output 1. Output from a “%metauser (user=all) ;” Run

MACRO TO READ SAS ROLES

What if you want to see the roles defined in SAS Metadata? You might need to get user and groups associated with specific roles. Obviously, you can login to SAS Management Console to find these information, but is this efficient way?

This macro explains, how you can run the below SAS code in your enterprise Guide session and fetch the report for defined roles and groups. Like previous macro, it provides the basic information but you can change and enhance this as per need. Final outcome of the macro is to create a data set which contain all the role, name, description and associated identity. If you want to check any specific role, then you can add the data steps to filter. Your code needs to be dynamic in that case, so that if someone is running for all roles then it reads entire data set without any filter.

Like previous macro, metauser and metapass both macro variables are for the user’s with whom you want to run the code. This is more SAS administrative macro and if you want to get complete list of roles and groups then you may need to run this with unrestricted user. Unrestricted id has full access and can bring the detailed report for you.

Syntax – If you want to call the below macro in your enterprise Guide session then, you can call like: %metaroles; . It executes with your credentials but If you want it to run with some other administrative id like sasadm@saspw then you can call like: %metaroles (metauser=, metapass=) ;

SAS Code

Below the full code for metauser macro:

```
%macro metaroles(metauser=, metapass=);
options metaserver='<metadata_server_name>'
metaport=<port>
metaprotocol='bridge'
metauser="&metauser"
metapass="&metapass"
metarepository='Foundation';

data Rolemembers;
    length id IdentType IdentId $17 RoleName Role_obj Role Uri  Role_desc
    ident_uri $ 256 identName  $60 type $32;
    call missing(type, id, RoleName, Role_obj, Role_desc, Role Uri,
    Ident Uri, IdentId, IdentName, IdentType);

    Role_obj=cat("omsobj:IdentityGroup?IdentityGroup[@GroupType='Role']");
    Roles=metadata_resolve(Role_obj,type,id);

    if (Roles >0) then
        do n = 1 to Roles;
            call missing(Role Uri, RoleName,
            Ident Uri, IdentId, IdentName, IdentType);
            nobj=metadata_getnobj(Role_obj,n,Role Uri);
            rc= metadata_getattr(Role Uri, "DisplayName", RoleName);

            if (RoleName="") then
                rc= metadata_getattr(Role Uri, "Name", RoleName);
                rc= metadata_getattr(Role Uri, "Desc", Role_desc);

            mems=metadata_getnasn(Role Uri,"MemberIdentities",1,Ident uri);

            if (mems >0) then
                do j=1 to mems;
                    ids=metadata_getnasn(Role Uri,"MemberIdentities",j,ident Uri);
                    rc=metadata_getattr(ident Uri, "Id", IdentId);

                    rc= metadata_getattr(ident Uri, "DisplayName", IdentName);

                    if (IdentName="") then
                        rc= metadata_getattr(ident Uri, "Name", IdentName);
                        rc= metadata_getattr(ident Uri, "PublicType", IdentType);

                    put IdentName= RoleName= IdentType=;
                    output;

                    call missing(Ident Uri,IdentId, IdentName, IdentType);
                    end;
                else
                    do;
                        put IdentName= RoleName=;
                        IdentName=" ---- ";
                        IdentType=" ---- ";
                        output;
                        call missing(IdentType,IdentName);
                    end;
            end;
        end;
```

```

        call missing(Role Uri, RoleName);
    end;
run;

data available_roles (keep=RoleName Role_Desc IdentName IdentType);
set Rolemembers;
run;

proc sort data=available_roles;
by RoleName;
run;

/* If you don't want the ROLES which are not in used */

/*data available_roles;*/
/*set available_roles;*/
/*if IdentType = " ---- " then delete;*/
/*run;*/

proc print data=available_roles;
run;

%mend metaroles;
%metaroles;
```

Below is the sample output from running the macro to get the SAS defined Roles in metadata:

Obs	Ident Type	RoleName	Role_desc	Ident Name
1	Usr_Grp	AddMSOffice:Analytics	Analytics For MSOffice Users.	Public
2	----	AddMSOffice:OtherUser	Role of light users.	----
3	Usr_Grp	EG User	Capabilities for SAS EG Users	Report

Output 2. Output from a “ %metaroles;” Run

If you want to create fancy report, can write extra SAS code based the final data set created. Current macro is just printing the final data set to output but you can create different kind of reports. The ley of these macros to understand the metadata architecture and read the appropriate variables to find the required information.

CONCLUSION

SAS Macros given in this paper provides the foundation for reading the SAS Metadata. Specially for platform administrators, it's very important to understand the building block of metadata repository and read them quickly/ efficiently. There may be some tools available to get metadata reports but having your own SAS code provides the flexibility to customize your report and providing user capability to read their information, makes your (administrator) job easy.

REFERENCE

Singh, Piyush. and Pohl, Gerhardt M. Mar-2014. “Enhancing SAS® Piping Through Dynamic Port Allocation” *Proceedings of SAS Global Forum 2014*. Washington, DC, USA: SAS Institute. Available at: <http://support.sas.com/resources/papers/proceedings14/1826-2014.pdf>

Sangwan, Prasoon. Gupta, Tanuj. and Singh, Piyush. Apr-2016. “Key Requirements for SAS® Grid Users” *Proceedings SAS Global Forum*. Las Vegas, NV, USA: SAS Institute. Available at: <http://support.sas.com/resources/papers/proceedings16/7140-2016.pdf>

Singh, Piyush. Sangwan, Prasoon. Kumar, Veera. and Sanger, Todd. Nov-2016. “SAS® Enterprise Guide for SAS® Visual Analytics LASR Server” *Proceedings of South Central SAS User Group 2016*. San Antonio, TX, USA: SCSUG. Available at: <http://www.scsug.org/wp-content/uploads/2016/11/SAS%C2%AE-Enterprise-Guide-for-SAS%C2%AE-Visual-Analytics-LASR-Server.pdf>

RECOMMENDED READING

- SAS® 9.4 Language Interfaces to Metadata
- Base SAS® Procedures Guide

CONTACT INFORMATION

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

Piyush Singh

piyushkumar.singh@tcs.com

Prasoon Sangwan

prasoon.sangwan@tcs.com

Steven D Randolph

randolph_steven_d@lilly.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.