Paper 055-2013

How to list all users that have access to a SAS Information Delivery 4.3 Portal Page for SAS® Global Forum 2013

Bernt Dingstad, IF Insurance Corporation, Oslo, Norway

ABSTRACT

This paper describes how to access SAS Metadata from Base SAS client and make simple listings of often very urgent information and in the end distribute this information utilizing the SAS EBI framework.

INTRODUCTION

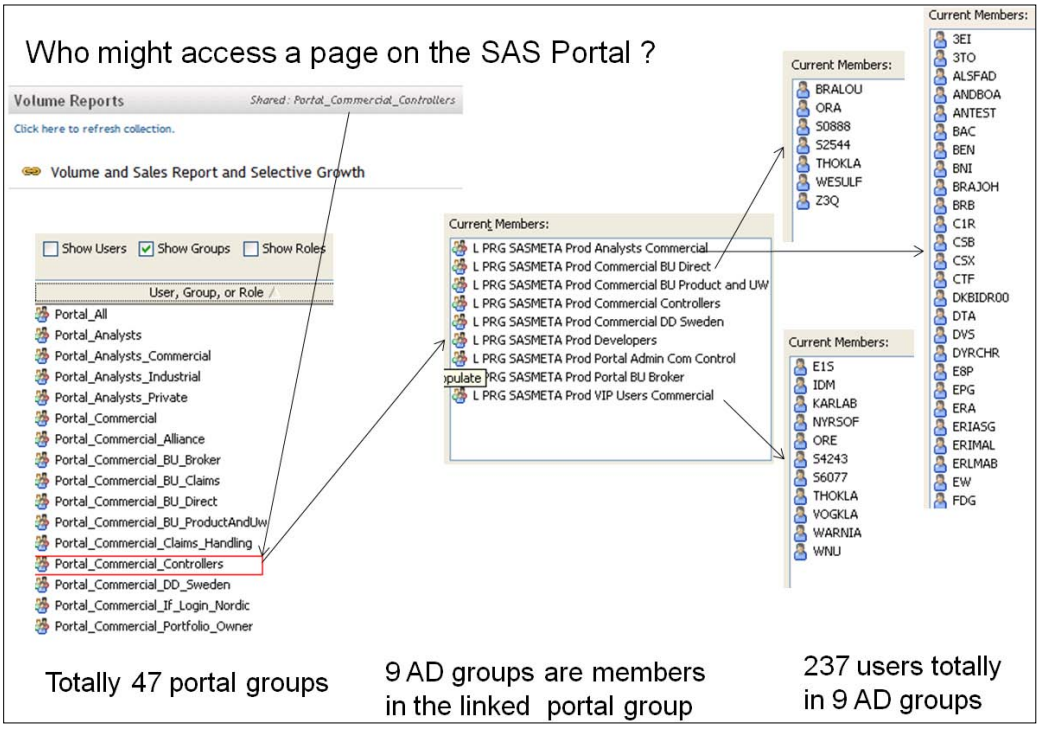
In the SAS Information Portal 4.3 there is no built-in report for viewing who have access to the different pages. There is just one SAS metadata group that can be mapped to each page and often; there is a need of having more than one group connected to each SAS portal page. The solution is then to create a Portal group and then include the all needed groups into this group. The challenge is then how to list all users who have access to the specific Portal page afterwards.

This paper will give an example on how this can be done by writing SAS code utilising SAS metadata function in a standard SAS data step.

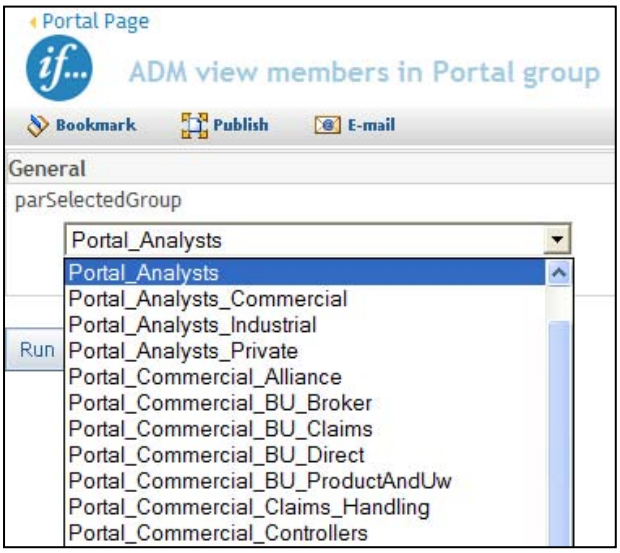
BACKGROUND

In large organization there are often a lot of different user groups that shall have access to information on different level in the organization. Then different groups are created in SAS Management Console depended on different needs in the organization. In some cases you like to give two or more groups access to the same information on the SAS Portal. One limitation/weakness in the SAS Portal is the fact that there are no possibilities to share a Portal object to more than one group defined in a SAS metadata repository. Many SAS EBI sites have then created specific SAS Portal groups with all needed groups included. But by implementing this workaround there has been more difficult for the administrators keeping track on who allowed accessing the different SAS Portal objects.

The figure below illustrates this challenge:



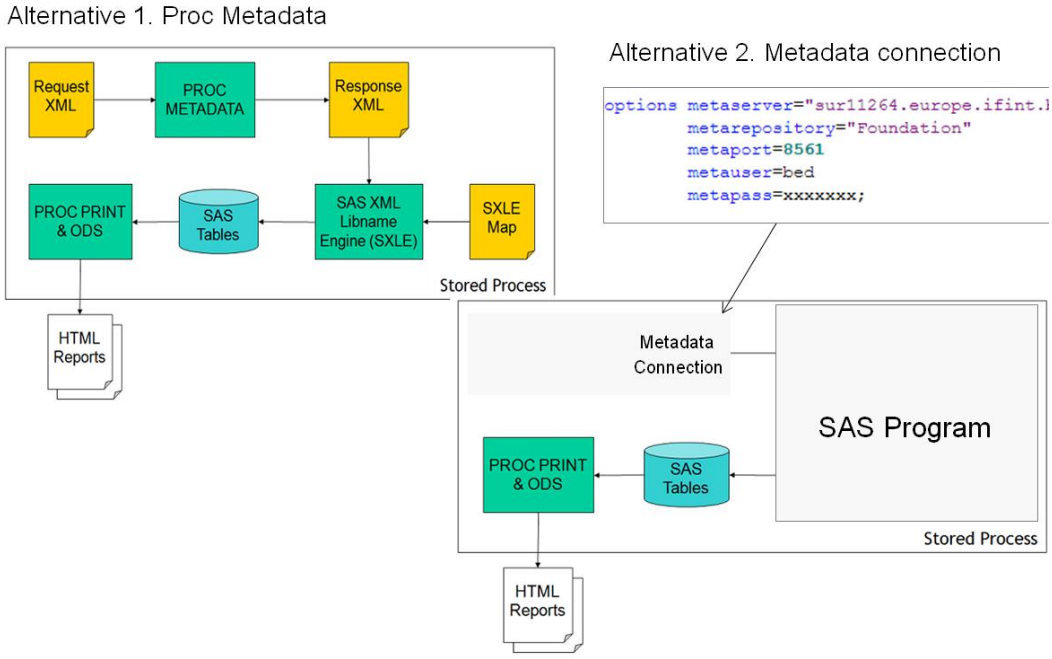
The idea is that we make one SAS Stored process (below named ADM view members in Portal group) with a drop down list containing all groups in SAS metadata begging with text “Portal_”. When the user select on of the portal groups in the list, a SAS program reads SAS metadata and retrieve all groups and corresponding members that have access to the specific Portal group. The figure below illustrate how this stored process looks like from an SAS Portal user point of view with just one dropdown list and a “Run” button for starting the SAS stored process.



For solving this challenge there are different possibilities to read information from SAS Metadata on how the “Portal groups”, on the left side are connected to the metadata groups in the middle. The two different methods are

1. Using Proc Metadata (alternative 1 below) or
2. Writing a custom Base SAS code (alternative 2) using a metadata connection and some SAS metadata functions for getting exactly your needed information.

Figure below illustrate the two alternatives for accessing SAS metadata information:



This paper is covering the alternative 2.

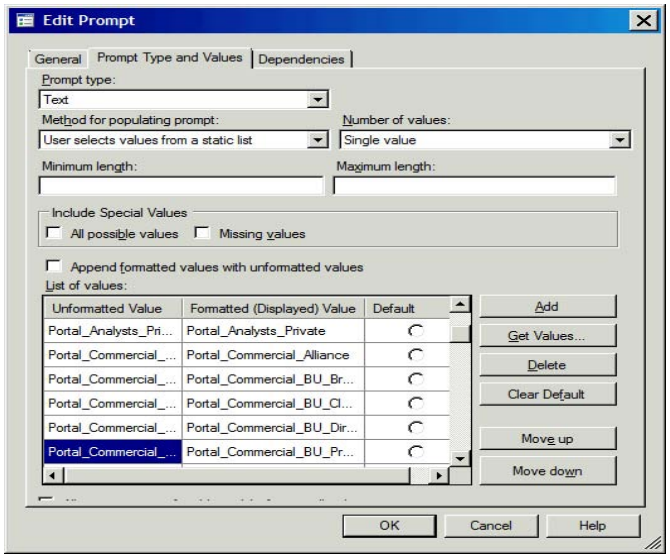
The first step is to establish a connection to SAS metadata. That is done by the following lines of SAS code:

```
options metaserver="sur11264.europe.ifint.biz"
        metarepository="Foundation"
        metaport=8561
        metauser=bed
        metapass="password on SAS metadata server";
```

Then we have to make a SAS dataset that contains all groups in SAS metadata that starts with text "**Portal**_". That is solved by the following SAS codes:

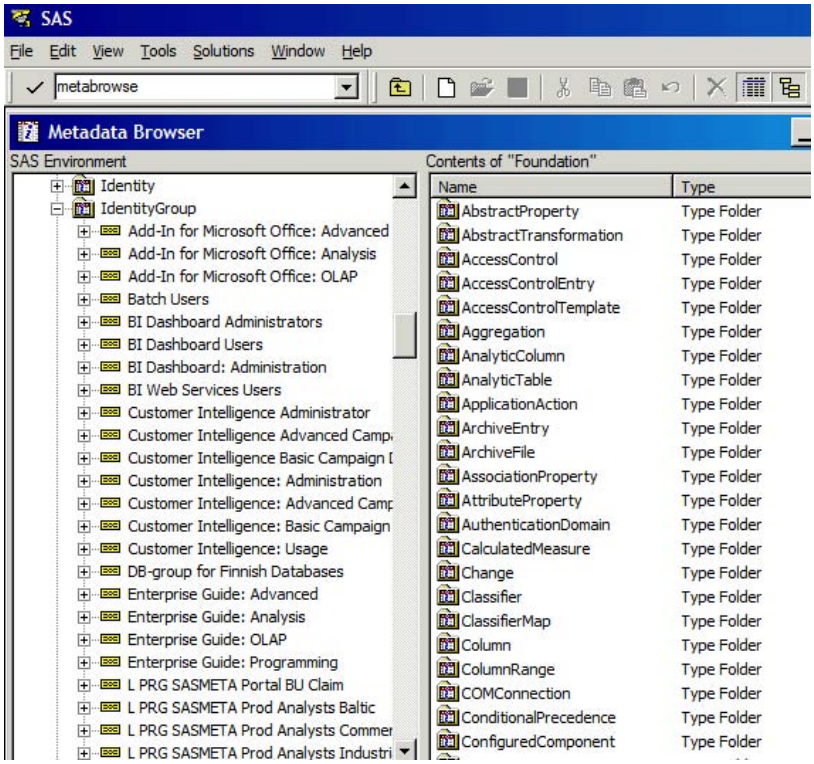
```
data Portal_groups (keep= group);
    length uri group $256;

    a=1;                    /* Counter for groups             */
    uri='';                 /* Unified Resource Identifier    */
    group='';               /* Name of Portal group           */

    /* Determine how many portal groups objects are defined. */
    nobj=metadata_getnobj("omsobj:IdentityGroup?@name contains 'Portal_'",a,uri);

    if nobj=0 then put 'No Group available.';

    else if  (nobj > 0) then do;
    /* Fetch all group names */
       rc=metadata_getattr(uri, "Name", Group);

       do while (a<=nobj);
          output;
          a+1;      /* Next Group */
          nobj=metadata_getnobj("omsobj:IdentityGroup?@name contains
                                 'Portal_'",a,uri);
          rc=metadata_getattr(uri, "Name", Group);
       end;
    end;
run;
```

If you store this dataset in a common location, this dataset can then be used for making a parameter prompt list in a SAS stored processes as shown below:



NEXT STEP, WRITING THE STORED PROCESS FOR LISTING ALL MEMBER

The next step is actually to make a SAS program that execute when a Portal page is selected in the stored process prompt. By using the **metabrowse** command in the command field in Base SAS, you will open the explorer tree for all metadata objects in the SAS Foundation metadata repository as shown below.



By selecting I**denityGroup** you are at the level where your SAS program has to start looking for the group which is selected as input variable. The following code makes a SAS dataset member including all members in groups on the levels Group/subgroup:

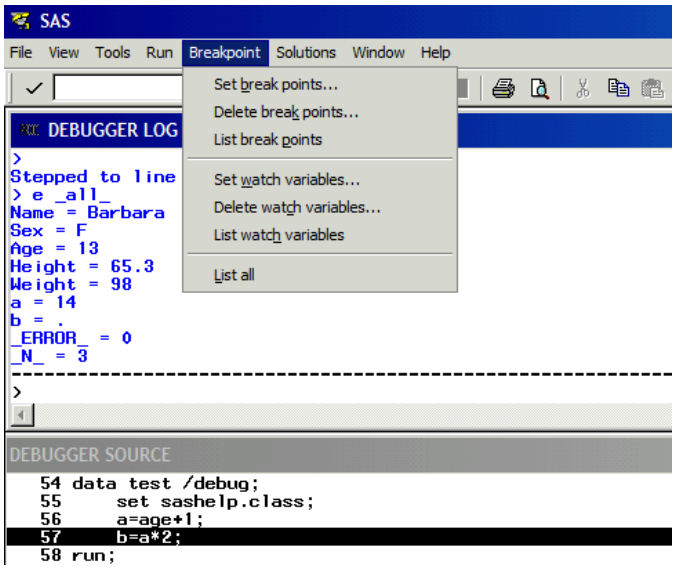
```
data members;
    length uri SelectedGroup group groupuri personSign personDESC personSign personID $256;

    /* initialize counters */
    n=1;  a=1;  b=1;
    /* Initialize variables to missing. */
    uri=''; SelectedGroup='';group=''; groupuri=''; personID=''; personDESC=''; personSIGN='';
    par=symget('parSelectedGroup'); /* Fetch global variable */
    nobj=metadata_getnobj("omsobj:IdentityGroup?@name='"||par||"'",n,uri);

    if nobj=0 then put 'No Group available.';
    else if  (nobj > 0) then do;
        /* Get the group association information for the current group */
        grpassn=metadata_getnasn(uri,"MemberIdentities",a,groupuri);
        /* retrive group name */
        rc2=metadata_getattr(groupuri, "Name", group);

        if grpassn in (-3,-4) then do;
            group="Group has no groups under itselves";
            output;
        end;

        /* If the portal group has more groups included, loop through the list */
        /* and retrieve the name, ObjectID and Desc for each person            */
        else do while (grpassn > 0);

            ant_users=metadata_getnasn(groupURI,"MemberIdentities",a,personID);
            b=1;
            /*Numbers of persons each group     */
            ant_users=metadata_getnasn(uri,"MemberIdentities",b,groupuri);
            /*Loop through all users in the group*/
            do while (b<=ant_users);
                rc2=metadata_getattr(personID, "Name", personSIGN);
                rc2=metadata_getattr(personID, "Object ID", personID);
                rc2=metadata_getattr(personID, "Desc", personDESC);
                output;
                /* Next person*/
                b+1;
                /*Fetching next personID */
                rc=metadata_getnasn(groupuri,"MemberIdentities",b,personID);
            end;
            /* Next Group */
            a+1;
            grpassn=metadata_getnasn(Uri,"MemberIdentities",a,groupURI);
            rc2=metadata_getattr(groupURI, "Name", group);
        end;
    end;
    keep group personSIGN personDESC;
run;
```

USE DEBUG OPTION WHEN YOU DOING COMPLEX SAS DATASEPS

To examine and control the values of the variable in a SAS datastep I have great experience using **/debug** as a dataset option like this:

```
data test /debug;
    set sashelp.class;
    a=age+1;
    b=a*2;
run;
```

You are then able to step through all lines in a data step and examine how values are assigned. This is of great value if you have a complex SAS data step with a loop inside another loop. You can examine all values typing **e <variableName>** or **e all** for all variables. You also have possibilities to use available functions as shown in *Run* and *Breakpoint* dropdown menu below:



CONCLUSION

By writing tailor made SAS code, you are able to list all information you need stored in a SAS metadata repository. Use **Metabrowse** to explore how the SAS metadata are structured and stored and start with some simple code and put on some more complex loops and iterations when you are familiar with the basics. You are then able to deliver reports and information that hopefully have great value for your organization.

REFERENCES

What is a URI?
http://support.sas.com/documentation/cdl/en/lrmeta/60739/HTML/default/viewer.htm#a003118273.htm

DATA Step Functions for Reading and Writing Metadata
http://support.sas.com/documentation/cdl/en/lrmeta/60739/HTML/default/viewer.htm#a003091925.htm

SAS Debug option
http://www.sfu.ca/sasdoc/sashtml/lgref/z0379345.htm