Paper 371-2012

How to Administer a 9,000-User Community on a Multiple SAS® Metadata Environment

Frank Baars and Edwin Nijsen, SNS REAAL, Alkmaar, Netherlands

ABSTRACT

Growing from a relative simple SAS® 9.1.3 BI, SAS® Data Integration Studio environment with 5 metadata servers and up to 600 users to a complex structure of in total 15 metadata servers and potentially 9,000 users, our user administration activities needed a serious update. As user administration was still done by hand, a new automated approach on user administration was essential. This paper proposes an automated user administration solution including single sign-on facilities and role-based access for a multiple SAS platform environment.

INTRODUCTION

SNS REAAL is a medium-large Dutch banking and insurance company consisting of 4 business units spread across the Netherlands with approximately 9,000 employees. Until two years ago our enterprise SAS® environment consisted of 5 metadata servers; 4 separate servers with SAS® 9.1.3 Data Integration Studio (DI) and SAS® 9.1.3 Business Intelligence (BI) for development, testing, acceptance and production and 1 server for SAS® Strategic Management 4.1 and 5 web servers. All user administration activities were done by hand, on each separate metadata server. A time consuming process, vulnerable to human errors for a user population of approximately 600 users.

After an intensive selection process the SAS® 9.2 BI suite was chosen to be the enterprise-wide Business Intelligence platform for SNS REAAL. Next to implementing SAS® 9.2 the organization also requested to implement the SAS® Financial Management Solution and to update SAS® Strategic Performance Management 2.4 to SAS® Strategic Performance Management 5.2. On top of that a whole new BI development environment was implemented in the form of 5 separate BI sandbox platforms.

All these changes and the increase in number of potential users forced administrators of the SAS® environment to enhance and automate the user administration process. In order to comply with legal regulations the company has a role-based access administration in place. To make sure that the SAS® environment would meet these compliancy rules it was necessary to connect it to the enterprise role-based access administration and implement a role-based access model for the authorization of the SAS® environment stored within the metadata.

The last business requirement on the implementation of SAS® 9.2 as the enterprise Business Intelligence platform was to implement the single sign-on facility for which automated user administration came in very handy.

OUTLINE OF THE ENTERPRISE SAS® ENVIRONMENT

The enterprise SAS® environment is fully implemented on Windows servers and consists of:

- 3 metadata servers for SAS® 9.2 Data Integration Studio: 1 development, 1 testing and 1 acceptance platform.
- 1 metadata server for SAS® 9.2 Enterprise Business Intelligence: 1 production platform.
- 2 metadata servers for the SAS® Financial Management 5.1 Solution: 1 testing and 1 production platform.
- 2 metadata servers for the SAS® Strategic Performance Management 5.2 Solution: 1 testing and 1 production platform.
- 5 metadata servers for SAS® 9.2 BI sandbox platforms: 1 server for each BI development department.

Note: Some of these servers are virtual servers

Note: Compute and web servers are not mentioned since they are not relevant to this paper.

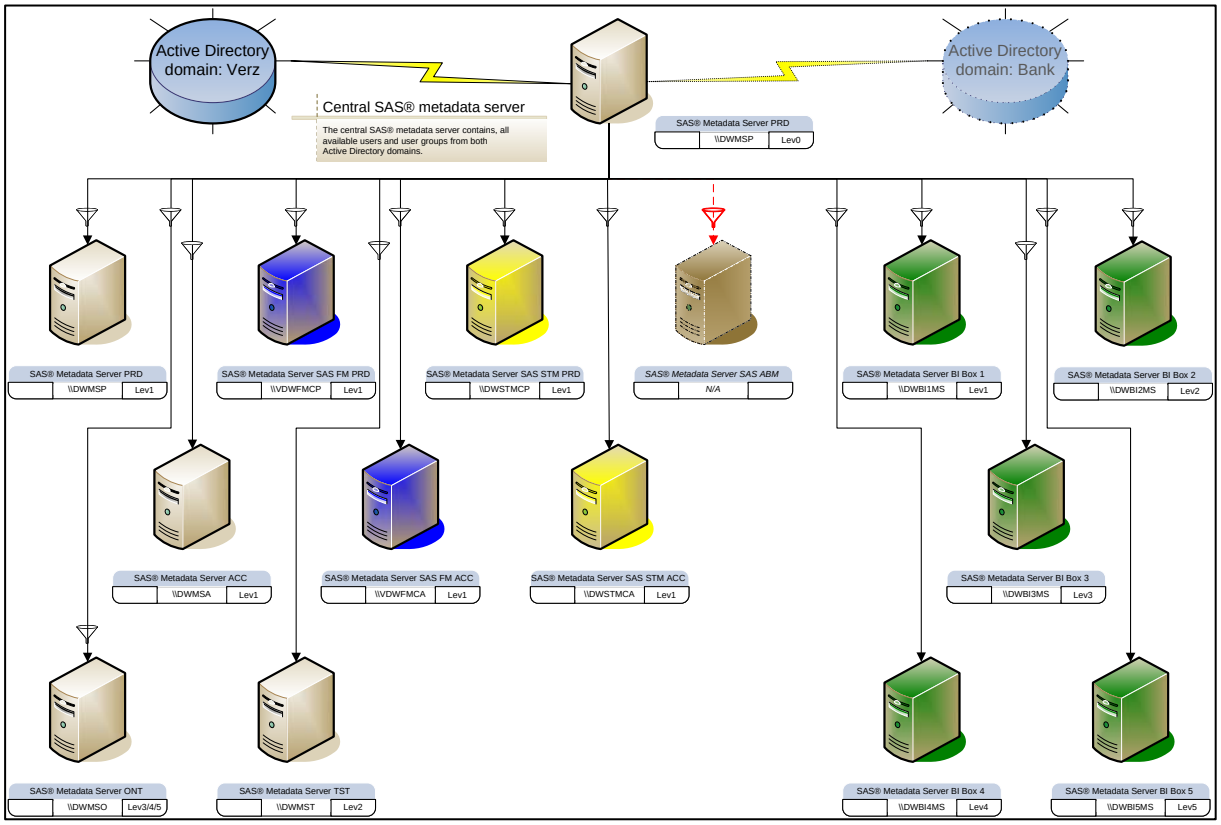

Figure 1. Overview of the enterprise metadata servers structure

THE SYNCHRONIZATION PROCESS

All employees are administered in the enterprise security provider: Active Directory, which made it possible to use the SAS® sample program, IMPORTAD.SAS as a jumpstart for our synchronization process. This baseline was adapted to the conditions at our site. For example the necessity to query two separate Active Directory domains and multiple problems with diacritic characters used in usernames or users existing on both domains. We decided to extract all identities from both domains and a selection of Active Directory groups, only those groups that give access to our SAS® servers or provide users with SAS® applications.

The extraction of approximately 9,000 users exceeded the maximum query limit of Active Directory, to circumvent this limit we used a loop-statement to filter all identities from the database:

```
DO filternr=0000 to 9999;
  filter="(&(objectCategory=Person)(objectClass=User)
         (&(sAMAccountName="||put(filternr,z4.)||"*)) )";
  %ldapextrpersons;
END;

filter="(&(objectCategory=Person)(objectClass=User)(&(sAMAccountName=F*)) )";
%ldapextrpersons
```

We've used multiple filters to query all the identities from Active Directory. Most users have a 7-digit login containing only numbers, the first filter is used to query these. The DO-statement is used to query Active Directory in portions of 1,000 logins at a time. The second filter is used to query functional user accounts which start with a certain character, in this case: 'F'.

CHALLENGES FOUND AND SOLVED DURING DEVELOPMENT

As mentioned earlier our site has two separate Active Directory domains. Most users are member of just one domain, but some are member of both, which led to multiple identities with matching names and/or logins. To circumvent this problem we added the name of the Active Directory domain to the login and we added the domain name to the displayName of the user.

When adding users and groups from the metadata server of SAS® Financial Management to the mySQL database underneath, using the standard SAS ® Data Integration jobs we discovered that a double parentheses in a username results in an error. To solve this problem we added a data cleansing step to the extraction process.

These cleansing steps are done after both Active Directory domains are queried for identities using this DATA step:

```
Data &extractlibref..ldapusers;
   Set &extractlibref..ldapusers_Totaal;
   Length DomainLogin $50;
   by CheckName;
   /* replacing illegal character (i.e. □) by an underscore */
   distinguishedName = tranwrd(distinguishedName,'1A'x,'_');
   displayName = tranwrd(displayName,'1A'x,'_');
   streetAddress = tranwrd(streetAddress,'1A'x,'_');
   cn = tranwrd(cn,'1A'x,'_');
   DomainLogin =
reverse(substr(reverse(strip(distinguishedName)),10,4))||"\"||sAMAccountName;

/* In case of two identities (from different Active Directory domains) with exactly
the same username, add name of the domain to the displayName of the identities. */
   if not (first.CheckName and last.CheckName) then do;
      displayName = trim(cn) ||"  ("||
reverse(substr(reverse(strip(distinguishedName)),10,4)) ||"_"||
trim(sAMAccountName) ||")";
   end;

/* Replacing double parentheses by single parenthesis; because of errors with MySQL
identities in the Financial Management Solution */
   displayName = trim(tranwrd(displayName,'))',')'));
   displayName = trim(tranwrd(displayName,'((','('));
Run;
```

After these cleansing steps the user and group information is queried from a separate SAS® metadata server instance on the production server called the Central Metadata Synchronization Environment (CMSE) using the %MDUEXTR macro. After the %MDUCMP comparison macro and %MDUCHGV macro which validates change tables, all possible question marks concerning users from who the external key (ExternalIdentity) is altered in Active Directory are filtered and put into a separate dataset. The number of filtered users is reported to the SAS® administrators in an exceptionlist. After a second run of the %MDUCHGV macro all users and groups are loaded into the CMSE by using the %MDUCHGL macro.

The CMSE is then used to synchronize all other SAS® 9.2 metadata servers within the enterprise network. By a second run of the %MDUEXTR macro the data are prepared as datasets for distribution to the other metadata servers.

After the update of the CMSE is finished, all 15 metadata servers are being synchronized using filters to ensure only necessary users and groups are synchronized with each of the metadata environments. For example: only users that are member of the Active Directory user group for the application SAS® Financial Management Studio or SAS® Financial Management Information Delivery Portal will be synchronized to the Financial Management metadata server).

The %MDUGRPAC macro is used to secure all new user groups added to each of the metadata servers with the Access Control Template (ACT) "SAS Administrator Settings".

After completion of all synchronizations all SAS® logs are checked for errors. Both the results from all synchronizations and the error messages are reported in an email to the SAS® administrators.

```
/***  Check logfile for errors  ***/
%Macro SCAN4ERROR (logdir=, logfile=) ;
  %let opt_notes  = %sysfunc (getoption (notes, keyword)) ;
  %let opt_source = %sysfunc (getoption (source, keyword)) ;
  %let opt_mprint = %sysfunc (getoption (mprint, keyword)) ;

  options nonotes nosource nomprint ;

  %global log_error;
  %let log_error = 0;

%put Scanning &logfile. for errors ;
  data _null_ Templog(drop=record);
    infile "&logdir.\&logfile" lrecl=2000 truncover ;
    input record $2000. ;
    if (indexw (record, 'ERROR:') )
      and (substr (record, 1, 6) ne 'MPRINT' and substr( scan(record,2,' '),1,1) ne
'+' ) then do ;
        call symput ('log_error', 1) ;
        bestand = "&logfile";
        resultaat = "FOUT";
        foutregel = _N_;
        Attrib foutmelding LENGTH= $300;
        foutmelding = Substr(TRIM(record),1,300);
        output Templog;
    end ;
  run ;

  %if &log_error = 0
  %then %do;
    DATA Templog;
       bestand = "&logfile";
       resultaat = "OK";
       foutregel = .;
       foutmelding = "";
    run;
    %end;

  options &opt_notes &opt_source &opt_mprint ;
%Mend SCAN4ERROR;
```

Partial code used to scan the log files to be processed for errors

```
   /* nrdir       number of logfiles to be processed */
   /* loglocatie  directory for logfiles             */
   /* dir1-dirx   logfile filename                   */
   %DO i=1 %TO &nrdir;
      /* Check log for errors */
      %scan4error (logdir=&loglocatie. ,logfile=&&dir&i);

      Data SynchLog_Details_Temp;
         Attrib Bestand     LENGTH = $50
                Proces      LENGTH = $20
                Datum       LENGTH = 8. Format=Date10.0
                Resultaat   LENGTH = $10
                Foutregel   LENGTH = 8.
                Foutmelding LENGTH = $300;
         Set Templog;

         Datum=Input(Reverse(Substr(Reverse(Strip(Bestand)),
         Index(Reverse(Strip(Bestand)),'_')+1,8)),yymmdd8.);
         Proces=Substr(Bestand,1,index(Bestand,compress(year(Datum)))-2);
         if indexw(Foutmelding,'%put ERROR:')=0 then output;
```

```
   Run;

   Proc Append Data = SynchLog_Details_Temp
               Base = Check.SynchLog_Details
               Force;
   Run;
%END;

/*Append processed log files to Synchlogdir to be able to skip them next time */
Proc Append Data = work.SynchLogDir2
            Base = Check.SynchLogDir;
Run;

/* Copy Synchlogdir2 to checklib for use in the next step */
 Data check.SynchLogDir2;
    Set work.SynchLogDir2;
 Run;
```

In addition to the Active Directory information, we also query the enterprise's database for role-based access security (RBA). All available users and their roles are added to the CMSE.

Changes in the Active Directory database are made daily by the IT services department (e.g. new employees, employees changing department, removing resigned employees) as well as changes to functional roles are maintained in the RBA by its administrators. To update these changes the user synchronization process is scheduled every night to run unattended. Synchronization of each metadata server is defined as a separate process to minimize dependencies. Each morning the only thing the SAS® administrators need to do is to check synchronization update email with the results.

EXTRA SYNCHRONIZATION STEP FOR SAS® SOLUTIONS

For both SAS® Strategic Performance Management and SAS® Financial Management it's necessary to run three standard SAS® Data Integration jobs (‘solnsvc_1300_load_users’, ‘solnsvc_1400_load_groups’ and ‘solnsvc_1500_load_user_x_group’). These jobs transfer users and groups from the metadata server to the mySQL database. These jobs are scheduled after the metadata synchronization.

Changes in the Active Directory database are made daily by the IT services department (e.g. new employees, employees changing department, removing resigned employees) as well as changes to functional roles are maintained in the RBA by its administrators. To update these changes the user synchronization process runs every night unattended. Synchronization of each metadata server is defined as a separate process to minimize dependencies. Each morning the only thing the SAS administrators need to do is check their mailbox for an e-mail with the results of synchronization.

By implementing this solution every user automatically gains access to the necessary objects and data according to his or her role(s).

IMPLEMENTING ROLE-BASED ACCESS

Role-based access is a best practice used world-wide to implement and maintain security settings throughout an enterprise. As part of the financial services industry, a reliable security system is an important focus for our company. A separate database was implemented to administer functional and additional roles for the role-based access (RBA) security.

Because management information is one of the main focus areas for our SAS platform, security is an important part of the SAS administration task. In order to benefit from the knowledge gained in the process of setting up role-based access security, we decided to reuse the roles defined in the RBA.

Once added to the metadata these roles can be used to grant rights to SAS metadata objects such as folders, libraries, cubes, reports and information maps.

In the BI production environment all metadata objects which can be shared to users are given their own usergroup and ACT (Access Control Template). Each metadataobject usergroup contains one or more rolegroups. Since the release of SAS® Management Console 9.2 its possible to see of which groups a group is member of. This makes this hierarchical structure possible.

Each new product delivered by one of the BI developers needs to be accompanied by a list of functional roles to which the object must be delivered.

SECURITY REPORTS

Another advantage of this security setup is the possibility to report on the security settings within the SAS metadata using the %MDSECDS macro. This macro function exports all security settings to a set of SAS datasets which can be then used to create security audit reports.

To clarify what kind of security reports can be thought of, two examples for reports that can make audits from the security officer a bit easier are given.

EXAMPLE 1

This report shows read and write access permissions set on the library DM_CMI.

Security Report for DM_CMI

ObjName=DM_AGENT

identityname	Delete	Read	Write
LA MIC Datamart CMI	Denied Indirectly	Granted Indirectly	Denied Indirectly
PUBLIC	Denied Indirectly	Denied Indirectly	Denied Indirectly
SAS General Servers	Denied Indirectly	Granted Indirectly	Denied Indirectly
SAS System Services	Denied Indirectly	Denied Indirectly	Denied Indirectly
SASAdministrators	Denied Indirectly	Granted Indirectly	Granted Indirectly
SASSched_P	Granted Indirectly	Granted Indirectly	Granted Indirectly
SASUSERS	Denied Indirectly	Denied Indirectly	Denied Indirectly

ObjName=DM_BANCAIRMUT

identityname	Delete	Read	Write
LA MIC Datamart CMI	Denied Indirectly	Granted Indirectly	Denied Indirectly
PUBLIC	Denied Indirectly	Denied Indirectly	Denied Indirectly
SAS General Servers	Denied Indirectly	Granted Indirectly	Denied Indirectly
SAS System Services	Denied Indirectly	Denied Indirectly	Denied Indirectly
SASAdministrators	Denied Indirectly	Granted Indirectly	Granted Indirectly
SASSched_P	Granted Indirectly	Granted Indirectly	Granted Indirectly
SASUSERS	Denied Indirectly	Denied Indirectly	Denied Indirectly

ObjName=DM_BANCAIRSTAND

identityname	Delete	Read	Write
LA MIC Datamart CMI	Denied Indirectly	Granted Indirectly	Denied Indirectly

Figure 2. Part of a security Report on library DM_CMI

EXAMPLE 2

In this report the members of the role “Rol R00001” are listed.

This report does not require use of the %MDSECDS macro.

Report Listing

Name=Rol R00001

memName	memDesc
Lu<censored> (Stefan)	User
Ke<censored> (Sebastiaan)	User
Be<censored> (Maureen)	User
Da<censored> (Francien)	User
At<censored> (Jan)	User
Ku<censored> (Cobie)	User
Ha<censored> (Pieter)	User
Gr<censored> (Rick)	User
Kr<censored> (Ben)	User
Lu<censored> (Rene)	User

Figure 3. Functional Role Group Members

SINGLE SIGN-ON

With the introduction of SAS® 9.2 the possibilities for implementing single sign-on were warmly welcomed by the end-user community. Except for the Financial Management 5.1 applications all the SAS applications are configured to benefit from Windows Integrated Authentication or the web authentication process.

For the use of Single sign on in the web environment we had to add an extra web authentication domain to the metadata users.

Because we already had the synchronization running for the AD groups we added the creation of a web authentication domain entry for each user to the user synchronization process. A cleansing step was added to remove the duplicate web authentication domain entries in the normalized login table.

Relevant changes to the original IMPORTAD.SAS code:

Declaration of the WebAuthDomain in section 1.

```
%let MetadataAuthDomain=DefaultAuth;

%let WebAuthDomain=WebAuth;
```

The other additions are made in section 3.

Addition of an extra data entry for each login.

```
      if sAMAccountName NE "" then do;

         /* setup login values */
         /* we need to prefix the login user id with the domain id */

         if "&WindowsDomain" = "" then
            userid = sAMAccountName ;
         else
            userid = "&WindowsDomain\" || sAMAccountName ;

         password ="";
         authdomkeyid = 'domkey' || compress(upcase("&MetadataAuthDomain"));

         output &logintbl;

         /* Extra entry for web based Single Sign On */

         userid = sAMAccountName ;
         password ="";
         authdomkeyid = 'domkey' || compress(upcase("&WebAuthDomain"));

         output &logintbl;
      end;
```

Creation of the authentication domain record.

```
data &authdomtbl;
     %defineauthdomcols;  /* Macros to define Table authdomain from %mduimpc */
     authDomName="&MetadataAuthDomain";
     keyid='domkey' || compress(upcase("&MetadataAuthDomain"));
     output;
     /* Extra autenthication domain for web base Single Sign On */
     authDomName="&WebAuthDomain";
     keyid='domkey' || compress(upcase("&WebAuthDomain"));
     output;
run;
```

Removal of duplicate UserID by authentication domain entries

```
proc sort data=&logintbl;
     by userid descending keyid;
run;

data &logintbl &logintbl.dubbel;
  set &logintbl;
  by userid descending keyid;
    if first.userid then output &logintbl;
    else output &logintbl.dubbel;
run;
```

CONCLUSION

A growing SAS user community requires a proper user administration. This can be achieved by creating an automated user administration solution. The SAS sample program IMPORTAD.SAS is a great start for implementing this in a windows environment.

Combining this solution with role-based access, the changes in user permissions are automatically updated in the SAS metadata.

Not only will this process lead to less user adminstrion tasks and fewer errors (because less is done by hand), but every user has exactly the permissions to data and reports he or she should be able to see conforming the roles defined within the company.

Every security officer will be pleased to hear you can ensure this authorization because of the automation and can create security reports on demand.

As a bonus the adding a webauthentication domain to each metadata user to implement single sign on for the web environment became very simple.

REFERENCES

- SAS Institute Inc. 2009 “SAS(R) 9.2 Intelligence Platform: Security Administration Guide” user import macros. Available at http://support.sas.com/documentation/cdl/en/bisecag/61133/HTML/default/viewer.htm#a003175544.htm
- Forrest Boozer, Christoper Zogby “Case Study in Synchronizing Identities in the SAS®9 Metadata Server with an Enterprise Security Provider” Proceedings of the SAS Global Forum 2007 Conference. Available at http://support.sas.com/rnd/papers/sgf07/sgf2007-syncmetadata.pdf

CONTACT INFORMATION

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

Frank Baars
SNS REAAL
P.O. Box 274
1800 BH Alkmaar, Netherlands
Work Phone: +31 72 548 6272
E-mail: Frank.Baars@snsreaal.nl
Web: http://www.snsreaal.nl

Edwin Nijsen
SNS REAAL
Work Phone: +31 72 548 6459
E-mail: Edwin.Nijsen@snsreaal.nl
Web: http://www.snsreaal.nl