

**Extending the Metadata Security Audit Reporting
Capabilities of the Audit and Performance
Measurement Package**

October 2010

ENTERPRISE EXCELLENCE CENTER

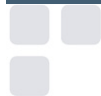


Table of Contents

1	Introduction.....	1
2	Metadata Security Audit Data	2
2.1	Where does the data come from?.....	2
2.2	What data is collected?	2
2.2.1	The AUDIT_TRANSACTIONS and Related Data Sets	4
2.2.2	The AUDIT_ACCESSCONTROLDETAILS Data Set	6
3	Metadata Security Audit Reporting	7
3.1	The Metadata Security Audit Reports Shipped with the Package	7
3.2	Extending Metadata Security Audit Reporting	8
3.2.1	Sample Scenario: Introduction	8
3.2.2	Sample Scenario: Data Extraction	8
3.2.3	Sample Scenario: Creating the Report	9
3.2.4	Sample Scenario: Surfacing the Report.....	11
4	Conclusion.....	13
5	For More Information and References	14
	Appendix A: The AUDIT_TRANSACTION Family of Data Sets	15
	Structure of Data Sets.....	15
	List of Valid Record Type and Record Event Values.....	17
	Matrix of Populated Columns by Record Type	19
	Appendix B: The AUDIT_ACCESSCONTROLDETAILS Data Set.....	20
	Structure of Data Sets.....	20
	Explanation of Effective Permission Values.....	21
	Appendix C: Important Files, Programs and Code Modules.....	22
	Batch/Script Files	22
	SAS Macro Programs	22
	HTML Files.....	22
	Stored Processes.....	22
	Appendix D: Sample Message Line and Matching Data Record.....	23
	Access Control Change Transaction	23
	Access Control Details Information	24

1 Introduction

The SAS 9.2 Enterprise Business Intelligence Audit and Performance Measurement package collects and reports on three categories of data. This includes status information for the SAS servers and Web applications that make up a SAS Enterprise Business Intelligence deployment; information about changes to metadata security; and information about the usage of the various types of content (e.g. SAS reports, information maps, tables). The metadata security audit information allows organizations to track security changes made to the various objects stored in the metadata. The package surfaces some of this data in the Metadata Audit reports, a set of reports available through the sample web interface contained in the package. However, these reports only surface a portion of the data available. Organization might be interested in extending this reporting to include all or some of the other data collected. This document describes the metadata security audit information that is collected; explains where it comes from, how it is structured and how it can leveraged for additional reports.

2 Metadata Security Audit Data

2.1 Where does the data come from?

The metadata security audit data is generated by the SAS Metadata Server as users interact with it. Leveraging the enhanced logging facility of SAS 9.2, including its support for ARM 4.0, the metadata server writes messages to log files. When the package is installed, the SAS Metadata Server logging configuration is modified. Additional messages related to security changes are generated and written to specific log files. Once the package is deployed, these log files are periodically read, processed and the metadata audit information is extracted into SAS data sets. This data is used for the existing audit reports and is available for additional reporting. Figure 1 depicts the process at a conceptual level.

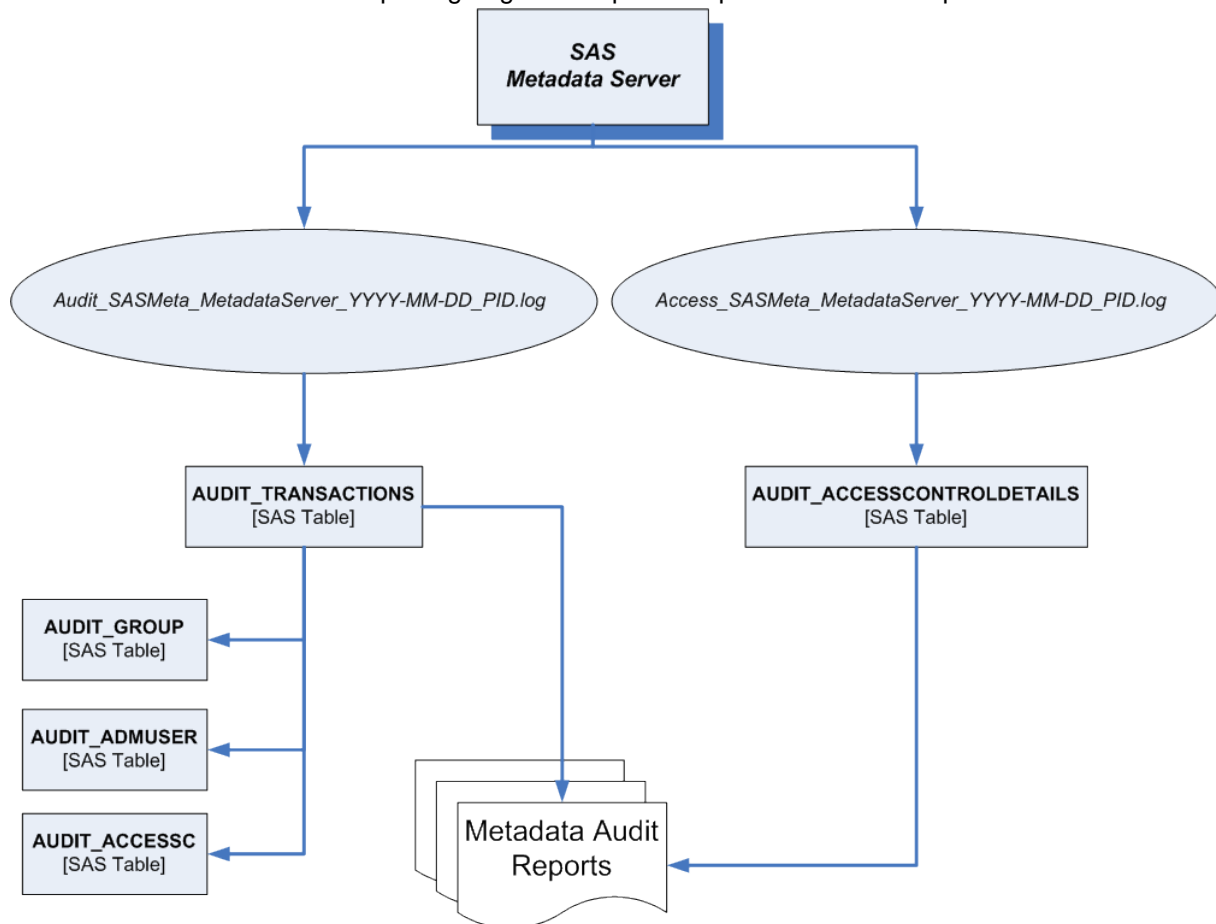


Figure 1: Overview

2.2 What data is collected?

The information collected focuses on two things: changes made to metadata objects related to security and changes made to the security controls for any of the objects in the metadata repository. An example of a security-related metadata object is a Login object within the SAS Metadata Server. A Login object represents a set of credentials available to gain access to some resource. The metadata audit data

captures information about Login objects over time as users create, modify or delete them. In addition to capturing events related to the security-related objects within the metadata, the metadata audit logs also capture changes made to the security controls (the effective permissions) assigned to any object within the metadata. For example, the metadata server may include an object representing a specific SAS table. In addition to the table's name and the columns within the table, the metadata will also include security information such as identifying which users have read or write access to it. When changes are made to these controls, messages are written to the metadata security audit log files.

Table 1 identifies the security objects for which metadata security audit information is collected. Messages are written to the metadata log files when any of these objects are added, modified or deleted. Log messages are also generated when any of these actions are attempted but fail because the user is not authorized to perform the action.

Metadata Objects		Description
Access Control		Access Control objects represent the limits that are defined for specific metadata objects.
Access Control Template		Access Control Template objects represent a set of specific Access Control limits that can be granted or denied as a unit.
Authentication Domain		The authentication domain object is used to represent where a set of credentials (i.e. a Login object) is valid.
Identity	These objects represent an identity in some form; specifically	
	Group	A group (0,1 or more) of users
	Person	A specific user
	Role	Roles objects are used to control access to specific application functionality (as opposed to controlling access to specific content). Just as users can be added to a Group, users can also be assigned a specific Role. A user who has been assigned a specific Role can access functionality that other users, who have not been assigned that Role, cannot.
Internal Login		An InternalLogin is a set of credentials that has no link to any system outside of SAS; it exists only within the SAS Metadata Server. InternalLogin credentials can be used to access the SAS Metadata without authenticating to an outside authentication provider.
Login		These objects represent a set of credentials (a user ID and, optionally, a password) that are authenticated by an external authentication provider. A Login is used to validate the identity of the user accessing the SAS Metadata Server or to access an external system such as a database or specialized server.
Member		Member objects identify group membership and role assignments. They represent the list of members (Person identity objects) belonging to a specific group (Group identity object) or the users (Person identity objects) that have been assigned a specific role (Role identity object).
Permissions		Permission objects represent specific access privileges to metadata objects (e.g. READ, READMETADATA, DELETE.) that can be granted (or denied) to users.
Protected Password		Protected Passwords represent a password needed to

Metadata Objects	Description
	access Tables, Connections or data through a protected pass-through connection.

Table 1: Metadata Objects and Activity Captured in the Metadata Audit Log Files

In addition to information related to metadata objects, the audit log files also capture information about other user activity of interest. This includes activity related to access to the metadata repository by administrative users, successful connections to the metadata server by clients and connection attempts that failed due to authentication errors.

2.2.1 The **AUDIT_TRANSACTIONS** and Related Data Sets

The log files are processed and the metadata security audit information is extracted into a number of SAS data sets. As shown in Figure 1, the **ARTIFACT.AUDIT_TRANSACTIONS** data set contains information extracted from the primary log file called **AUDIT_SASMeta_MetadataServer_YYYY-MM-DD_PID** (where **YYYY-MM-DD** represent the four digit year, month and day the log file was opened and **PID** represents the process ID of the SAS Metadata Server). This data set is the data source for most of the metadata security audit reports that are included in the package.

Each record in the data set corresponds to a message line from the log file. Each record in the data set contains fields that allow it to be mapped back to a specific line in a log file. Other fields contain the date and time the log message was written and the identity of the user who performed the activity. The record type (**A_RecordT**) field identifies the category of the log message described. This identifies the type of object that changes or the type of activity that is recorded. A second field (**A_RecordEvent**) further clarifies the specific action or activity.

Table 2 shows the possible values for the **A_RecordT** field and the types of objects or activity records, which each value represents. A complete description of the table structure is included in Appendix A.

Objects /Message Category		A_RecordT	
Access Control		"AccessControl"	
Access Control Template		"AccessControlTemplate"	
Administrative User		"AdminUser"	
Authentication Domain		"AuthenticationDomain"	
Authentication Error		"AuthenticationError"	
Client Connection		"ClientConnection"	
Group Membership		"Group"	
Identity Objects	Groups of User	"Identity"	and A_IdentityType= "IdentityGroup"
	User		and A_IdentityType= "Person"
	Role		and A_IdentityType= "Role"
Internal Login		"InternalLogin"	
Login		"Login"	
Permissions		"Permission"	
Protected Password		"ProtectedPassword"	
All Other Activity		"Metadata"	

Table 2: List of A_RecordT Values

The other fields in the **ARTIFACT.AUDIT_TRANSACTIONS** table provide additional details about the action. The fields that are populated with data vary based on the record type and event. For example, while the IP address and port number fields are relevant for records representing a user’s attempt to establish a connection, these fields are not relevant for a record representing a change in group memberships. Therefore, in the first case, the **A_ClientIPAddr** and **A_ClientPort** fields would be populated while in the latter case the fields would be empty. A matrix showing which columns are populated for each of the record types is also included in Appendix A.

A small number of other SAS data sets are derived from the **ARTIFACT.AUDIT_TRANSACTIONS** data set. These data sets are focused on specific categories of message and contain a subset of the records and fields that are relevant to the subject of each data set. These data sets can be used to make reporting more efficient when reports are focused on a specific subset of activity.

These data sets include:

- **ARTIFACT.AUDIT_ADMUSER** – Information about user identities that accessed the metadata server with special administrative access.
- **ARTIFACT.AUDIT_GROUP** – Information about changes in group membership and role assignments
- **ARTIFACT.AUDIT_ACCESSC** – information about changes to Access Control Templates and access controls on individual metadata objects.

2.2.2 The **AUDIT_ACCESSCONTROLDETAILS** Data Set

The SAS data set, **ARTIFACT.AUDIT_ACCESSCONTROLDETAILS**, provides detailed information about the access control settings for any metadata objects that are modified. This data is extracted from the second metadata audit log file, **Access_SASMeta_MetadataServer_YYYY-MM-DD_PID.log** (where **YYYY-MM-DD** represent the four digit year, month and day the log file was begun and **PID** represents the process ID of the SAS Metadata Server). Information about the structure for this data set is included in Appendix B. Due to the structure of the log messages, unlike the other audit data sets, the records in this data set are derived from multiple lines in the log file. Each record captures the permissions granted and or denied to a specific identity as shown on the **Authorization** tab of the **Properties** dialog box for the object in the SAS Management Console. Therefore, there will be multiple rows (one for each of these identities) in this table for any metadata object that is modified. For each permission, the record indicates whether the identity has been granted or denied this permission. It also indicates whether the grant or denial was done explicitly (i.e. for this identity by name) or was inherited from a parent object or set through an Access Control Template (ACT). It is important to understand two limitations of this data. Each row of this table reflects the settings for particular point in time; it does not indicate which permissions were modified. Therefore, reporting on changes made to the permissions for a given metadata object over time will require additional processing. In addition, although it is possible for users to create new permissions, information about these new permissions and their status will not be captured in this data set. The default table structure only supports the standard set of permissions.

3 Metadata Security Audit Reporting

3.1 The Metadata Security Audit Reports Shipped with the Package

The package includes a set of pre-defined reports leveraging the metadata security audit data. These reports are available from the upper right panel of the sample user interface. The interface allows the user to specify a report format and a time period in addition to selecting a specific report. Behind the scenes, the user's request is passed to a SAS stored process (**AuditReports**) that generates the actual report and returns it to the browser. These are dynamic reports generated on-demand by the user. It is important to understand that although the report is generated on-demand, the data is not live. The data used in these reports reflects the last time the log files were processed; by default, the package processes log files once a day.

Table 3 lists the metadata security audit reports that are shipped as part of the package. In addition to a short description for each report, the table includes the filter criteria used to select the records to include in the report. All of the reports use data from the **ARTIFACT.AUDIT_TRANSACTIONS** data set and are generated by the **AuditReports** stored process. The only exception is the *Access Control Change Details* report. This report is linked to from the *Access Control Changes* report and generated by a second stored process, **AccessControlDetail**, using data from the **ARTIFACT.AUDIT_ACCESSCONTROLDETAILS** data set.

Sample Report Name
<p>Access Control Changes – identifies changes to access controls; links to a second, more detailed, report (see below) for information about specific changes made.</p> <p>Record Filter: <i>A_RecordT contains "AccessControl"</i></p>
<p>Administrator – identifies the users that have been granted some form of administrator access to the SAS Metadata Server and the level of access.</p> <p>Record Filter: <i>A_RecordT="AdminUser"</i></p>
<p>Authentication Errors – identifies failed authentication attempts; these could indicate mistyped or forgotten passwords or attempted trespass.</p> <p>Record Filter: <i>A_RecordT = 'AuthenticationError'</i></p>
<p>Groups Changes – identifies changes in group membership and changes in role assignments</p> <p>Record Filter: <i>A_RecordT="Group"</i></p>
<p>Login Not Authorized – identifies unauthorized attempts to modify logins (user credentials).</p> <p>Record Filter: <i>A_RecordT = 'Login' + A_RecordEvent contains 'Not Authorized'</i></p>
<p>User IDs Added – identifies any new logins that have been added.</p> <p>Record Filter: <i>[A_RecordT='Login' + A_RecordEvent contains 'Added']</i> OR <i>[A_RecordT='InternalLogin' + A_RecordEvent contains 'Changed']</i></p>

User IDs Removed – identifies any logins that have been removed.

Record Filter:

[A_RecordT = 'Login' + A_RecordEvent contains 'Removed']

OR

[A_RecordT='InternalLogin' + A_RecordEvent contains 'Removed']

Access Control Change Details – provides details about the current access control permissions for an object selected from the **Access Control Changes** report.

The relevant records from **ARTIFACT.AUDIT_ACCESSCONTROLDETAILS** related to a specific access control change.

Table 3: The Sample Metadata Security Audit Reports

3.2 Extending Metadata Security Audit Reporting

Organizations are not limited to the metadata security audit reports that are shipped with the package. The underlying data can be leveraged to produce different reports and or to report on activity not included in the standard reports.

3.2.1 Sample Scenario: Introduction

For example, a customer recently contacted SAS Technical Support asking if it was possible to create a report that tracked the custom roles that had been defined in their deployment and the users that had been assigned those roles. Not only is this possible, but the data is already being captured by the package and can be found in the **ARTIFACT.AUDIT_TRANSACTIONS** data set and its derivatives.

3.2.2 Sample Scenario: Data Extraction

The first step is to identify the data that is needed. The customer is looking for information about “roles”. Based on the information in Table 2, the relevant rows can be identified by filtering the data on the **A_RecordT** field. The table also shows that another field, **A_IdentityType**, must also be included to filter out other rows that deal with other forms of identity such individual users and group. Because the customer is interested only in the creation of new roles, a final criterion uses the **A_EventType** field to limit the extracted data to those that show this. The following pseudo-code puts all of this together to show the appropriate filter criteria needed to answer the first part of the question (identifying the new custom “roles”):

```
where (A_RecordT = "Identity" and A_IdentityType= "Role") /* filter: Role rows*/
and (A_RecordEvent eq: "Added"); /*filter: new Role Additions */
```

The second part of the question, identifying the users which have been assigned the new roles, can also be answered with the data in the **ARTIFACT.AUDIT_TRANSACTIONS** data set.

```
where (A_RecordT eq "Group" and A_RecordEvent eq: "Added") /*filter: group adds */
and {role in the list created above}
```

Because the **ARTIFACT.AUDIT_GROUPS** data set only contains records with the needed record type, using it as the input data set would improve efficiency. The following pseudo-code shows how all of the needed data could be extracted.

```
create table newroles as
  select  a_identityname,    /* name of new role */
         a_objid           /* metadata id of new object */
  from artifact.audit_transactions
  where  a_recordt='Identity' and a_identitytype='Role'
        and (index(a_recordevent,'Added IdentityType') gt 0);

create table memberAdds as
  select  a_identityname,    /* identity assigned new role */
         a_identitytype,    /* type of identity assigned new role */
         a_identitytargetobjid /* object-id of the role being assigned */
  from artifact.audit_group
  where  index(a_recordEvent,"Added") gt 0);

create table newRoleMembers as
  select  r.a_identityname as newRoleName,
         m.a_identityname as userid,
         m.a_identitytype as persontype
  from newroles r left join memberAdds m
  on r.a_objid = m.a_identitytargetobjid;
```

3.2.3 Sample Scenario: Creating the Report

Once the necessary data has been extracted, a report can be created using any of the tools and techniques available from within SAS software. One approach is to integrate this new report into the set of existing metadata security audit reports. To do this the stored process used to generate the metadata security audit reports must be modified. As mentioned earlier, this stored process is called **AuditReports** and is found in the folder **SAS Folders → Shared Data → EBIAPM92 → AUDIT** in the metadata repository. The metadata for this stored process does not need to be modified; however, the SAS source code will need to be modified. By default, the source code is found in the file **{SAS Config Dir}\Lev1\SASApp \SASEnvironment\SASCode\Jobs\bi_audit_reports.sas**.

The structure of this program is straightforward. For each of the reports, a block of SAS code extracts the appropriate data, sets macro parameters to control how the report is rendered and calls a standard report generating macro. Adding the new **“New Roles”** report involves inserting a new block of comparable code into the stored process source code. In addition to allowing the user to select a report, the sample Web interface allows the user to select a period of interest and specify some report formatting options. Using the existing code as a guide, these same capabilities can be supported for the new report.

Some important things to remember:

- The final data set used by the report generating macro should be a temporary data set with the name **Results**
- The user specified period is used to construct an appropriate WHERE clause. This WHERE clause is available as the SAS macro variable **&WHERE_CLAUSE**
- The stored process code uses the following SAS macro variables to pass parameters to the report generating macro:
 - **COLUMNS**: used to pass the list of columns to include in the report; it is used to construct a COLUMNS statement in PROC REPORT.
 - **GROUPING**: used to pass a PROC REPORT statement to be included in report generation; for most of the existing reports, it is used to pass a DEFINE statement in PROC REPORT indicating how the rows should be grouped but it could be used to pass any PROC REPORT statement;
 - **GROUPING2**: used to pass an additional PROC REPORT statement; none of the existing reports currently use this parameter.

The following code block can be added to the program after the logic for the last of the standard reports.

```
%else %if (%index(%quote(&report_name),New Roles) > 0) %then %do;

proc sql;
  create table newroles as
    select a_identityname,
           a_objid
    from artifact.audit_transactions
       where a_recordt='Identity' and a_identitytype='Role'
          and (index(a_recordevent,'Added IdentityType') > 0)
          &where_clause;

  create table memberAdds as
    select a_identityname,
           a_identitytype,
           a_identitytargetobjid,
           a_activeuserid,
           a_datetime
    from artifact.audit_group
       where index(a_recordEvent,"Added") gt 0
          &where_clause;

  create table results as
    select r.a_identityname as newRoleName label='Role',
           m.a_identityname as username    label='Role Holder',
           m.a_identitytype as persontype  label='User or Group',
           m.a_activeuserid as adminID    label='Assigned By',
           m.a_datetime     as datetime   label='Date Role Assigned'
```

```
        from newroles r left join memberAdds m
            on r.a_objid = m.a_identitytargetobjid
        order by newRoleName,
            datetime,
            username;

quit;

%let columns=newRoleName username persontype adminID datetime;
%let grouping=define newRoleName / group 'Role';
%end;
```

3.2.4 Sample Scenario: Surfacing the Report

The last step is to make the new audit report available to the users. The developers are free to use any of the techniques and interfaces that can be leveraged using SAS software. Since the “**New Roles**” report created above was designed to be integrated into the existing audit reports, it will be surfaced to the users through the same sample Web interface. The metadata security audit reports make up the upper right-hand panel in the sample Web interface. The `bi_audit_report.html` file defines this panel. This file, and the other files that make up the sample web application, are shipped in the `{EBIAPM Install Dir}\Html` directory. As part of the original package deployment process these file are moved to a location, which can be surfaced through the organization’s Web servers. These changes must be made in the version of this file that end-users access from the Web server.

Adding the new report is accomplished by adding its name to list of possible reports, as shown in the following excerpt from the file:

```
<td><b>Report Name</b></td><td><Select name="report_name">
<option>Access Control Changes
<option>Administrators
<option>Authentication Error
<option>Group Changes
<option>Login Not Authorized
<option>Userids Added
<option>Userids Removed
<option>New Roles and Role Assignments
</select><br>
</td></tr>
```

Once the html file has been updated, the changes will available the next time the page is loaded, or, if the page is already in the browser, it will be available after a refresh. The screenshot in Figure 2 shows the updated interface and a sample of the new report.

SAS 9.2 Enterprise BI Server and Web Tier Status Report: 13SEP2010:16:35

Monitoring	Servers and Web App	Status	Validation Time	Last Checked
Mid-Tier	SASBIDashboard	Up		30AUG2010:11:10:06:55

SAS 9.2 Metadata Server Audit Reports

Report Name: New Roles
 Date Range:
 Report Format:
 ODS Style:

New Roles on 13SEP2010

Role	Role Holder	User or Group	Assigned By	Date Assigned
GregNewRole	Charlie	Person	sasadm@saspw	10SEP2010:14:25:49.708
	Gloria	Person	sasadm@saspw	10SEP2010:14:25:49.708
	demosasgz	Person	sasadm@saspw	10SEP2010:14:25:49.708
	Gloria	Person	sasadm@saspw	10SEP2010:14:26:18.161
	Harry	Person	sasadm@saspw	10SEP2010:14:26:18.161
	ReportAuthors	IdentityGroup	sasadm@saspw	10SEP2010:14:26:18.161
	demosasgz	Person	sasadm@saspw	10SEP2010:14:26:18.161
	sasdemo1	Person	sasadm@saspw	10SEP2010:14:26:18.161

Server Performance Reports

- Report Description
- Time Statistics for Artifact Usage
- Artifact Usage during Business Hours
- OLAP Cube Usage by User
- Top OLAP Cube Users
- Most Heavily Used Datasets
- Most Heavily Used Directories
- Metadata Client Status: Active / Inactive Userids
- Metadata Client Sessions: Login Details
- Duplicate Metadata references for a directory
- SAS PROC Usage
- Top 10 Report Usage
- Server Comparison of Elapsed versus CPU Time
- SAS Server Usage by SAS User
- Stored Process Average Response Time
- Stored Process Consumption by Elapsed Time
- Inventory of Metadata Artifacts
- Artifact Usage by User
- Top Workspace Users by Sessions
- Top Workspace Users by Memory Usage
- Workspace Server Usage by Hour

Figure 2: Screenshot Showing Availability of the "New Roles" Report and Sample Report Output

4 Conclusion

As part of maintaining a secure Enterprise Business Intelligence deployment, organizations need to monitor the SAS Metadata Server and track changes made to security information within its metadata repository. The package provides tools to track these changes and includes a number of related reports. This document has described how the metadata security audit data the package collects can be further leveraged for additional reporting. Understanding where the data comes from, what it contains and how it is organized, will allow developers to do this more effectively. The sample scenario described in this document was based on an actual customer request and illustrates the process for developers.

5 For More Information and References

SAS Institute Inc., *SAS 9.2 Enterprise Business Intelligence Audit and Performance Measurement for Windows Environments*, Cary, NC: SAS Institute Inc., 2009. A PDF version of this file is included in the installation media for the package.

SAS Institute Inc., *SAS 9.2 Enterprise Business Intelligence Audit and Performance Measurement for UNIX Environments*, Cary, NC: SAS Institute Inc., 2009. A PDF version of this file is included in the installation media for the package.

Additional information (including the installation media) is available from within the SAS R&D Enterprise Management focus area of the SAS support site. <http://support.sas.com/rnd/emi/EbiApm92/index.html>

Appendix A: The AUDIT_TRANSACTION Family of Data Sets

This appendix includes detailed information about the AUDIT_TRANSACTION table and related tables. In addition to information about the table structure, this section includes information about:

- Possible values of important fields
- The relationship between fields
- Identifies which fields will be populated for the different types of metadata security events or activities.

Structure of Data Sets

#	Variable	Length	Label
1	Log_Line	512	Line from Log File
2	A_DateTime	8	DateTime
3	startdt	8	
4	A_Level	5	Diagnostic Level
5	A_ClientID	8	Connection ID
6	A_ActiveUserid	16	Active Userid
7	A_Thread	8	Active Thread
8	Log_File	400	File Processed
9	A_MetaUserid	16	Metadata Userid
10	A_ClientIPAddr	36	Remote Client IP Address
11	A_ClientPort	5	Remote Client Port
12	A_RecordT	24	Audit Record Type
13	A_RecordEvent	64	Audit Record Event
14	A_IdentityType	24	Identity Type
15	A_IdentityName	36	Identity Name
16	A_ObjID	17	(Metadata) Object ID
17	A_ObjType	24	(Metadata) Object Type
18	A_AuthDomain	16	Authentication Domain
19	A_IdentityTargetType	64	Target Identity Type
20	A_IdentityTargetName	64	Target Identity Name
21	A_IdentityTargetObjID	17	Target Object ID
22	A_PermissionName	24	Metadata Permission
23	A_PermissionType	24	Permission Type
24	A_Repository	36	Repository Name
25	A_ACT_Message	64	Audit Message

Table 4: The AUDIT_TRANSACTION Data Set

#	Variable	Length	Label
5	A_ActiveUserid	16	Active Userid
2	A_DateTime	8	DateTime
4	A_Level	5	Diagnostic Level
7	A_RecordT	24	Audit Record Type

6	A_Thread	8	Active Thread
1	Log_Line	512	
3	startdt	8	

Table 5: The AUDIT_ADMUSER Table

#	Variable	Length	Label
5	A_ActiveUserid	16	Active Userid
2	A_DateTime	8	DateTime
10	A_IdentityName	36	Identity Name
14	A_IdentityTargetName	64	Target Identity Name
15	A_IdentityTargetObjID	17	Target Object ID
13	A_IdentityTargetType	64	Target Identity Type
9	A_IdentityType	24	Identity Type
4	A_Level	5	Diagnostic Level
11	A_ObjID	17	Object ID
12	A_ObjType	24	Object Type
8	A_RecordEvent	64	Audit Record Event
7	A_RecordT	24	Audit Record Type
6	A_Thread	8	Active Thread
1	Log_Line	512	
3	startdt	8	

Table 6: The AUDIT_GROUP Data Set

#	Variable	Length	Label
12	A_ACT_Message	64	Audit Message
5	A_ActiveUserid	16	Active Userid
2	A_DateTime	8	DateTime
9	A_IdentityName	36	Identity Name
4	A_Level	5	Diagnostic Level
10	A_ObjID	17	Object ID
11	A_ObjType	24	Object Type
8	A_RecordEvent	64	Audit Record Event
7	A_RecordT	24	Audit Record Type
6	A_Thread	8	Active Thread
1	Log_Line	512	
3	startdt	8	

Table 7: The AUDIT_ACCESSC Data Set

Notes on Selected Fields

A_ActiveUserID – the identity, which established the connection to the metadata server; for message groups dealing with metadata objects, it represents the user who made the change captured in the log file. In other cases, the identity used to establish a connection to the metadata server is an internal utility user identity.

A_ClientID – the internal numeric identifier used by the SAS Metadata Server to keep track of client connections; it is *not* the IP address of the client.

A_DateTime – date and timestamp from log line, a SAS datetime value

A_IdentityType– the type of the Identity object identified in **A_IdentityName**; possible values: *User, Group or Role*

A_Level – diagnostic logging level; possible values: *TRACE, DEBUG, INFO, WARNING, ERROR, FATAL*

A_MetaUserid –the user identity involved; for authentication error messages, it identifies the user ID attempting to connect; for Login and InternalLogin messages: it identifies the user with which the Login (or InternalLogin) is associated; for admin message groups, it contains the user ID with administrator access.

A_RecordEvent – The specific activity or transaction being logged. See Table 8 for a list of possible values.

A_RecordT – the metadata object that has been modified or the category of log message represented. See Table 2 for possible values and their meaning.

Log_Line – the text of the line from Log File

startdt – a second datetime field, shows same timestamp info as **A_DateTime**

List of Valid Record Type and Record Event Values

The following table shows the valid values for the record type (**A_RecordType**) and record event (**A_RecordEvent**) fields. It is possible that some of these values will change as updates to the package are released. The SAS macro **%AUDITPROC** (source code: **{EBIAPM Install Dir}\SASEnvironment\SASMacro\auditproc.sas**) contains the logic used to set these values.

Message Category	A_RecordT Value	Possible A_RecordEvent Values
Access Control	"AccessControl"	"Access Control change " "Not Authorized to change Access Control " "Access Control definition change " "Not Authorized to change Access Control definition " " Deleted Access Control"
Access Control Template	"AccessControlTemplate"	"Added AccessControlTemplate " "Changed AccessControlTemplate " "Removed AccessControlTemplate " "Not Authorized to add AccessControlTemplate " "Not Authorized to remove AccessControlTemplate " "Not Authorized to change AccessControlTemplate "
Administrative User	"AdminUser"	"Admin User" "Unrestricted Admin User" "Trusted User"
Authentication Domain	"AuthenticationDomain"	"Added Authentication Domain Name" "Changed Authentication Domain Name" "Removed Authentication Domain Name" "Not Authorized to add Authentication Domain Name" "Not Authorized to remove Authentication Domain Name" "Not Authorized to change Authentication Domain Name"
Authentication Error	"AuthenticationError"	"Error authenticating user" "Access denied"
Client Connection	"ClientConnection"	"New Client Connection" "Client Connection Closed" "Unknown User Name"

Group Membership	<i>"Group"</i>	<i>"Added Member IdentityType"</i> <i>"Removed Member IdentityType"</i> <i>"Not Authorized to add Member IdentityType"</i> <i>"Not Authorized to remove Member IdentityType"</i>
Identity Objects Note: Use the A_IdentityType field to differentiate between User (A_IdentityType="User"), Group ("IdentityGroup"), or Role ("Role") identities	<i>"Identity"</i>	<i>"Added IdentityType"</i> <i>"Removed IdentityType"</i> <i>"Changed IdentityType"</i> <i>"Not Authorized to add IdentityType"</i> <i>"Not Authorized to delete IdentityType"</i> <i>"Not Authorized to change IdentityType"</i>
Internal Login	<i>"InternalLogin"</i>	<i>"Added Internal Login with UserId"</i> <i>"Changed Internal Login UserId"</i> <i>"Removed Internal Login with UserId"</i> <i>"Not Authorized to add Internal Login UserId"</i> <i>"Not Authorized to remove Login Internal UserId"</i> <i>"Not Authorized to change Login Internal UserId"</i>
Login	<i>"Login"</i>	<i>"Added Login with UserId"</i> <i>"Changed Login UserId"</i> <i>"Removed Login with UserId"</i> <i>"Not Authorized to add Login UserId"</i> <i>"Not Authorized to remove Login UserId"</i> <i>"Not Authorized to change Login UserId"</i>
Permissions	<i>"Permission"</i>	<i>"Added Permission Name"</i> <i>"Changed Permission Name"</i> <i>"Deleted Permission Name"</i> <i>"Not Authorized to add Permission Name"</i> <i>"Not Authorized to delete Permission Name"</i> <i>"Not Authorized to change Permission Name"</i>
Protected Password	<i>"ProtectedPassword"</i>	<i>"Added Password"</i> <i>"Changed Password"</i> <i>"Deleted Password "</i> <i>"Not Authorized to add Password "</i> <i>"Not Authorized to delete Password "</i> <i>"Not Authorized to change Password "</i>
All Other Log Messages	<i>"Metadata"</i>	<i>"Server Event"</i>

Table 8: List of Valid Record Type and Record Event Values

Matrix of Populated Columns by Record Type

Other Columns	Record Type (Value of A_RecordT)												
	AccessControl	AccessControlTemplate	AuthenticationDomain	AuthenticationError	Login	Group	InternalLogin	Identity	Permission	ProtectedPassword	Metadata	AdminUser	ClientConnection
A_RecordEvent	x	x	x	x	x	x	x	x	x	x	x	x	x
A_ActiveUserid	x	x	x	x	x	x	x	x	x	x	x	x	x*
A_ObjID	x	x	x		x	x	x	x	x	x			
A_IdentityName	x	x			x	x	x	x		x			
A_ACT_Message		x		x									
A_AuthDomain			x		x								
A_IdentityTargetObjId					x	x	x			x			
A_IdentityType					x	x	x!	x					
A_MetaUserid				x	x		x!!						
A_ObjType	x	x								x			
A_IdentityTargetName						x							
A_IdentityTargetType						x							
A_PermissionName									x				
A_Permission_Type									x				
A_Repository									x				
A_ClientID				x									x
A_ClientIPAddr				x									x**
A_ClientPort				x									x**
Notes: ! Column only populated for InternalLogin: Added or InternalLogin:Removed records !! Column only populated for InternalLogin:Changed or InternalLogin:Not Authorized records * Column will not be populated for ClientConnection:Close records linked to a failed authentication error. ** Column only populated for ClientConnection:Open records													

Table 9: Populated Columns by Record Type

Appendix B: The AUDIT_ACCESSCONTROLDETAILS Data Set

Structure of Data Set

#	Variable	Length	Label
1	A_DateTime	8	DateTime
2	A_ClientID	8	Connection ID
3	A_ActiveUserid	16	Active Userid
4	A_ObjID	17	(Metadata) Object ID
5	User_Group	8	Identity
6	Administrator	8	State of Administrator Permission (see Table 11)
7	CheckInMetadata	8	State of CheckIn Metadata Permission (see Table 11)
8	Delete	8	State of Delete Permission (see Table 11)
9	Read	8	State of Read Permission (see Table 11)
10	ReadMetadata	8	State of Read Metadata Permission (see Table 11)
11	WriteMetadata	8	State of Write Metadata Permission (see Table 11)
12	WriteMemberMetadata	8	State of WriteMember Metadata Permission (see Table 11)
13	Create	8	State of Create Permission (see Table 11)
14	Execute	8	State of Execute Permission (see Table 11)
15	Create_Table	8	State of Create Table Permission (see Table 11)
16	Drop_Table	8	State of Drop Table Permission (see Table 11)
17	Alter_Table	8	State of Alter Table Permission (see Table 11)
18	Select	8	State of Select Permission (see Table 11)
19	Insert	8	State of Insert Permission (see Table 11)
20	Update	8	State of Update Permission (see Table 11)
21	References	8	References

Table 10: AUDIT_ACCESSCONTROLDETAILS Data Set

Notes on Selected Fields

A_ActiveUserID – the user who made the change captured in the log file

A_ClientID – the internal identifier used by the SAS Metadata Server to keep track of client connections; it is *not* the IP address of the client.

A_DateTime – date and timestamp from log line, a SAS datetime value

A_IdentityType– Type of Identity object; i.e. User, Group or Role

All other fields – the other fields in the table represent the state of a specific metadata permission for a specified identity (**A_ActiveUserID**) at a specific point in time (**A_DateTime**). Possible values are shown in Table 11. A permission may be granted or denied through multiple mechanisms and each field can contain more than one value. Refer to the SAS documentation for an explanation of each of the permissions.

Explanation of Effective Permission Values

Code	Meaning	Description
AG	ACT Grant	This permission has been <i>granted</i> through an Access Control Template
AD	ACT Deny	This permission has been <i>denied</i> through an Access Control Template
EG	Explicit Grant	This permission has been <i>explicitly granted</i> to this user or group identity
ED	Explicit Deny	This permission has been <i>explicitly denied</i> to this user or group identity
NG	Indirect Grant	This permission has been <i>indirectly granted</i> to this user or group identity; permission grant inherited from a parent object.
ND	Indirect Deny	This permission has been <i>indirectly denied</i> to this user or group identity; permission denial inherited from a parent object.

Table 11: List of Permission Values and Their Meaning

Appendix C: Important Files, Programs and Code Modules

Batch/Script Files

Scripts found in folder: {EBIAPM Install Dir}

readSASLogs.bat – driver script for the log processing; the log files are identified, parsed and data is extracted into SAS data sets.

SAS Macro Programs

Source Code in folder: {EBIAPM Install Dir}\SASEnvironment\SASMacro

auditproc.sas – primary log processing logic for audit data; includes log line parsing and creation of records in **AUDIT_TRANSACTIONS** table, including assignment of **A_RecordT** and **A_RecordEvent**.

auditaccessdetail.sas – primary log processing logic for access control log; includes log line parsing and creation of records in **AUDIT_ACCESSCONTROLDETAILS** table.

HTML Files

The files that make up the sample web application are shipped in the *{EBIAPM Install Dir}\html* directory. As part of the deployment process these files are moved to a location to be surfaced through the organization's Web servers. Therefore, the ultimate location for these files is site-specific.

Files found in folder: {EBIAPM Install Dir}\html

bi_audit_reports.html – defines the contents of the Metadata Server Audit Report panel of the sample Web interface.

bienv.html – defines the overall sample Web interface.

Stored Processes

Source Code in folder: {SAS Config Dir}\Lev1\SASapp\SASEnvironment\SASCode\Jobs

Metadata found in: SAS Folders → Shared Data → EBIAPM92 → AUDIT

bi_audit_reports.sas – Generates the most of the existing metadata security audit reports.

bi_accesscontrol_detail.sas – Generates the **Access Control Change Details** report

Appendix D: Sample Message Line and Matching Data Record

The following tables show the log messages generated when the access controls were modified for a metadata object in a test environment. After the excerpt of the log files, the corresponding records created in the two audit tables are also shown.

Access Control Change Transaction

```
2010-07-29T10:28:58,099 INFO [00004042] 176:demoUser@SASBI - Access Control change on ObjectType=Tree,
Name=My Folder, ObjId=A5QTSUMO.AJ00011K.
```

Table 12: Sample Message Line (from Audit_SASMeta_Metadataserver_2010-07-29_2308.log)

A_DateTime	29JUL2010:10:28:58.099
A_ActiveUserid	demoUser@SASBI
A_RecordT	AccessControl
A_RecordEvent	Access Control change
A_IdentityName	My Folder
A_ObjID	A5QTSUMO.AJ00011K
A_ObjType	Tree
Log_File	Audit_SASMeta_Metadataserver_2010-07-29_2308.log
Log_Line	2010-07-29T10:28:58,099 INFO [00004042] 176:demoUser@SASBI - Access Control change on ObjectType=Tree, Name=My Folder, ObjId=A5QTSUMO.AJ00011K
A_Level	INFO
A_ClientID	176
A_ClientIPAddr	
A_ClientPort	
A_IdentityTargetName	
A_IdentityTargetObjID	
A_IdentityTargetType	

A_IdentityType	
A_MetaUserid	
A_PermissionType	
A_Repository	
A_Thread	00004042
startdt	29JUL2010:10:28:58.099
A_PermissionName	
A_ACT_Message	
A_AuthDomain	

Table 13: Matching Record in AUDT_TRANSACTION Table

Access Control Details Information

2010-07-29T10:28:58,099 INFO [00004042] 176:demoUser@SASBI - Access Control change on ObjectType=Tree, Name=My Folder, ObjId=A5QTSUMO.AJ00011K.

2010-07-29T10:28:58,115 TRACE [00004042] 176:demoUser@SASBI - Trace log showing effective permissions protecting object: OMSOBJ:Tree/A5QTSUMO.AJ00011K.

demoUser Person Administer=ND, CheckInMetadata=EG|ND, Delete=ND, Read=EG, ReadMetadata=EG|ND, Write=ND, WriteMetadata=ND, WriteMemberMetadata=EG|ND, Create=ND

PUBLIC IdentityGroup Administer=ND, CheckInMetadata=AD|ND, Delete=ND, Read=ND, ReadMetadata=AD|ND, Write=ND, WriteMetadata=AD|ND, WriteMemberMetadata=ND, Create=ND

SASAdministrators IdentityGroup Administer=NG, CheckInMetadata=AG|ND, Delete=NG, Read=NG, ReadMetadata=AG|ND, Write=NG, WriteMetadata=AG|ND, WriteMemberMetadata=NG, Create=NG

SAS System Services IdentityGroup Administer=ND, CheckInMetadata=ND, Delete=ND, Read=NG, ReadMetadata=AG|ND, Write=ND, WriteMetadata=ND, WriteMemberMetadata=ND, Create=ND

SASUSERS IdentityGroup Administer=ND, CheckInMetadata=ND, Delete=ND, Read=NG, ReadMetadata=ND, Write=ND, WriteMetadata=ND, WriteMemberMetadata=ND, Create=ND

SAS Demo User Person Administer=ND, CheckInMetadata=ND, Delete=NG, Read=NG, ReadMetadata=ND, Write=NG,

WriteMetadata=ND, WriteMemberMetadata=ND, Create=NG

Table 14: Sample Log Messages Showing Access Control Settings (from Access_SASMeta_MetadataServer_2010-07-29_2308.log)

Record #	A_DateTime	A_ClientID	A_ActiveUserId	A_ObjID	User_Group	Administer	CheckIn Metadata	Delete	Read	ReadMetadata	Write	WriteMetadata	Write Member Metadata	Create
7	29JUL2010:10:28:58.099	176	demoUser@SASBI	A5QTSUM0.AJ00011K	demoUser Person	ND	EG ND	ND	EG	EG ND	ND	ND	EG ND	ND
8	29JUL2010:10:28:58.099	176	demoUser@SASBI	A5QTSUM0.AJ00011K	PUBLIC IdentityGroup	ND	AD ND	ND	ND	AD ND	ND	AD ND	ND	ND
9	29JUL2010:10:28:58.099	176	demoUser@SASBI	A5QTSUM0.AJ00011K	SASAdministrators IdentityGroup	NG	AG ND	NG	NG	AG ND	NG	AG ND	NG	NG
10	29JUL2010:10:28:58.099	176	demoUser@SASBI	A5QTSUM0.AJ00011K	SAS System Services IdentityGroup	ND	ND	ND	NG	AG ND	ND	ND	ND	ND
11	29JUL2010:10:28:58.099	176	demoUser@SASBI	A5QTSUM0.AJ00011K	SASUSERS IdentityGroup	ND	ND	ND	NG	ND	ND	ND	ND	ND
12	29JUL2010:10:28:58.099	176	demoUser@SASBI	A5QTSUM0.AJ00011K	SAS Demo User Person	ND	ND	NG	NG	ND	NG	ND	ND	NG
13	29JUL2010:10:28:58.146	176	demoUser@SASBI	A5QTSUM0.AJ00011L	demosasgzs Person	ND	EG ND	ND	EG	EG ND	ND	ND	EG ND	ND
14	29JUL2010:10:28:58.146	176	demoUser@SASBI	A5QTSUM0.AJ00011L	PUBLIC IdentityGroup	ND	AD ND	ND	ND	AD ND	ND	AD ND	ND	ND
15	29JUL2010:10:28:58.146	176	demoUser@SASBI	A5QTSUM0.AJ00011L	SASAdministrators IdentityGroup	NG	AG ND	NG	NG	AG ND	NG	AG ND	NG	NG
16	29JUL2010:10:28:58.146	176	demoUser@SASBI	A5QTSUM0.AJ00011L	SAS System Services IdentityGroup	ND	ND	ND	NG	AG ND	ND	ND	ND	ND
17	29JUL2010:10:28:58.146	176	demoUser@SASBI	A5QTSUM0.AJ00011L	SASUSERS IdentityGroup	ND	ND	ND	NG	ND	ND	ND	ND	ND

18	29JUL2010:10:28:58.146	176	demoUser@SASBI	A5QTSUM0.AJ00011L	SAS Demo User Person	ND	ND	NG	NG	ND	NG	ND	ND	NG
----	------------------------	-----	----------------	-------------------	-------------------------	----	----	----	----	----	----	----	----	----

Table 15: Corresponding Records in AUDT_ACCESSCONTROLDETAILS Table

Note: The following fields were omitted from the table above because they had no values: Execute; Create_Table; Drop_Table; Alter_Table; Select; Insert; Update; References

Copyright © 2010 SAS Institute Inc., Cary, NC, USA. All rights reserved. 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 registered trademarks or trademarks of their respective companies. All Rights Reserved.