

What SAS Administrators Should Know about Libraries, Metadata, and SAS Enterprise Guide

For SAS 9.2 and SAS Enterprise Guide 4.2 and 4.3

BACKGROUND

SAS Enterprise Guide provides access to libraries that are defined in a SAS metadata repository, in addition to those libraries that are defined within a SAS session using the LIBNAME statement. When accessing a SAS library via its definition in metadata, SAS Enterprise Guide usually uses the META library engine. The META engine provides a layer of indirection between the SAS session and the physical data store, and it behaves according to the rules defined within the SAS metadata repository. Because the META engine behaves differently than the "native" library engine (such as BASE for file-based data, or a SAS/ACCESS library for DBMS data), it's important for SAS administrators to understand the different behaviors and how these can be controlled.

This document describes the behaviors of the META engine and its use in SAS Enterprise Guide, and provides guidance to SAS administrators who would like to see the libraries behave in a certain way.

Note: for information that pertains to SAS 9.1.3 and SAS Enterprise Guide 4.1, see: *What SAS® Administrators Should Know about Libraries, Metadata, and SAS® Enterprise Guide 4.1* (<http://support.sas.com/techsup/technote/ts788.pdf>).

ABOUT SAS METADATA LIBRARIES

SAS libraries defined in metadata have two main components:

1. The library definition, which describes how to assign the library. This information includes the path, engine, and any library-specific options that are needed. These parts of the definition correspond to the syntax of the traditional LIBNAME statement that most SAS programmers are familiar with.

An administrator can attach authorization details to a library definition, controlling which users can view/access the contents of the library and change its contents.

2. The table definitions, which describe the tables and columns available within the library. Table definitions are added to metadata in three main ways:
 - Using SAS Management Console, you can select the library and use the Import Tables feature, which allows you to connect to the library, view the physical tables that exist, and import the desired table definitions into metadata.
 - Using the METALIB procedure from within a SAS session; this is a programmatic method of importing table metadata, synchronizing metadata with the physical contents of the library. You can also use the Tools->Update Library Metadata feature in SAS Enterprise Guide to generate PROC METALIB code.
 - Using SAS Data Integration Studio to build a data warehouse. The process of building a data warehouse updates metadata about libraries, tables, and columns.

Administrators can also attach authorization details to table definitions, controlling which users can access specific tables and even specific columns within those tables.

Library and table metadata is a prerequisite for many data-related features in SAS clients and solutions. For example, if you want to provide a business view of data by defining information maps in SAS Information Map Studio, you must first add metadata about the tables and columns that feed into those information maps.

ABOUT THE METADATA LIBRARY ENGINE (META ENGINE)

The META engine is a special SAS library engine that enforces the metadata-centric view of a SAS library. The engine provides a level of indirection between your SAS programs and the physical location of your data, allowing you to reference a library definition that exists in metadata.

For example, this LIBNAME statement:

```
libname hr meta library="Human Resources";
```

looks up the library definition named “Human Resources” within metadata, and provides access to its contents via the HR libref. The physical location of the tables is not evident in this statement. For example, the library definition might refer to SAS data sets in a file path using the BASE engine, or it might refer to a DBMS source using the ORACLE engine.

The tables and columns surfaced in the resulting HR libref reflect the tables and columns defined in metadata and the permissions that you, as an authenticated user, have to view the data.

META ENGINE BEHAVIOR – READ ONLY BY DEFAULT

The META engine, by default, prevents your SAS programs from modifying the library contents in a way that would cause the data to become out of sync with the registered metadata. Attempts to do so will result in an error message similar to the following:

```
ERROR: You cannot create or delete datasets, views or indexes in this mode. Try the
option METAOUT=DATA. Use Proc Metalib to create metadata for datasets.
```

For example, using the example HR library definition from earlier, this SAS program would fail:

```
data hr.newdata;
  set hr.empinfo(where=(gender="F"));
run;
```

ADDING/MODIFYING TABLES VIA THE META ENGINE

The default behavior of the META engine is to provide a purely metadata-centric view of your library. However, you can specify an additional option on the LIBNAME statement to produce a sort of hybrid view of the library. The METAOUT=DATA option allows the META engine to show any additional tables that don't exist in metadata, but do exist in the physical library location. Here is an example statement:

```
libname hr meta library="Human Resources" metaout=data;
```

With the METAOUT=DATA option in place, you can add and modify tables in the library. While the library engine, in this mode, allows access to “non-metadata” tables, the engine still enforces the permissions defined on those tables and columns that do exist in metadata. That is, the METAOUT=DATA option does not open access to tables and columns where READ permission has been set to Deny in metadata.

For example, with the proper permissions in place, this SAS program will add a table to the physical library, but not update the metadata:

```
libname hr meta library="Human Resources" metaout=data;
data hr.newdata;
  set hr.empinfo(where=(gender="F"));
run;
```

Here is an example SAS log:

```
NOTE: Libref HR was successfully assigned as follows:
```

```
Engine:          META
```

```
Physical Name:
```

```
14      libname hr meta library="Human Resources" metaout=data;
15      data hr.newdata;
16      set hr.empinfo(where=(gender="F"));
```

```
NOTE: HR.EMPINFO was found in the SAS Metadata Repository.
```

```
17          run;
```

```
NOTE: There were 104 observations read from the data set HR.EMPINFO.
      WHERE gender='F';
```

```
NOTE: The data set HR.NEWDATA has 104 observations and 17 variables.
```

```
NOTE: DATA statement used (Total process time):
```

```
      real time          0.03 seconds
      cpu time           0.03 seconds
```

SYNCHRONIZING PHYSICAL DATA AND TABLE METADATA

If your SAS program makes changes to the contents of the library in METAOUT=DATA mode, those changes are **not** reflected in the metadata. In order to synchronize the metadata with the physical changes made during your program, you must:

- Use SAS Management Console to import the new/changed tables into the library definition
- Use the METALIB procedure to update metadata definitions with the physical table attributes.
- Use Tools->Update Library Metadata task in SAS Enterprise Guide. This task is a simple series of windows that can generate PROC METALIB code for you.

For example, this SAS program will synchronize the metadata to add the additional table defined earlier:

```
/* sync up metadata with physical library */
/* requires Create permissions for the library */
libname _temp meta library="Human Resources" metaout=data;
proc metalib;
  omr (library="Human Resources");
  update_rule=(delete);
  report;
run;
libname _temp clear;
```

SAS ENTERPRISE GUIDE AND LIBRARY ASSIGNMENTS

Note: All library behaviors described for SAS Enterprise Guide also apply to SAS Add-In for Office.

SAS Enterprise Guide offers improved integration with libraries defined in metadata, including new options for controlling how libraries are assigned.

When you view a SAS server in SAS Enterprise Guide in the server view, you see a list of libraries. This list includes assigned libraries – those that are built-in, pre-assigned, or assigned in an autoexec program. The list also includes *potential* libraries – those defined in metadata that haven't actually been assigned or accessed yet, but that you are authorized to view.

Figure 1 shows an example view of a SAS server with a mix of assigned and potential libraries.

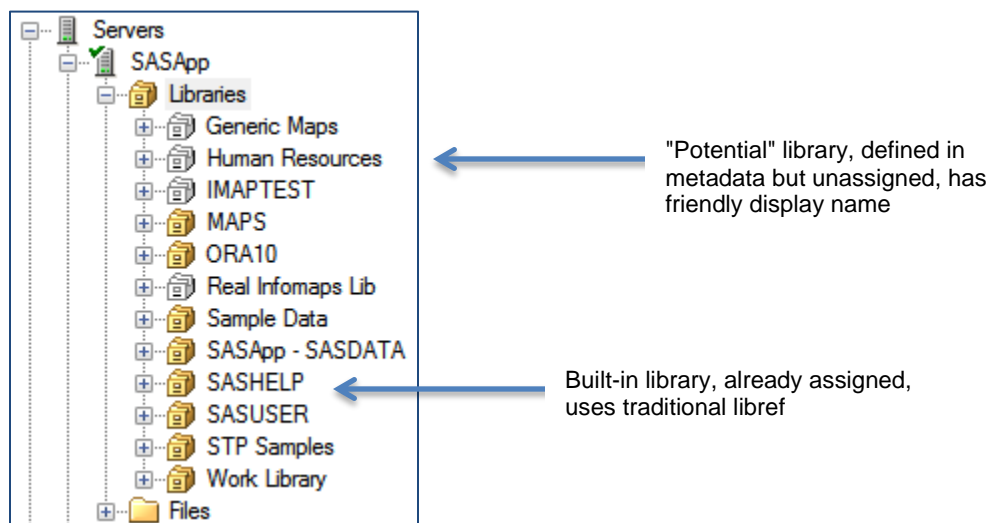


Figure 1. An example view of libraries in the Servers list.

The library icons that are shaded yellow represent the **assigned** libraries. Those that are white represent the **potential** or **unassigned** libraries.

By default, when you expand an unassigned library by clicking on it, or assign it by right-clicking on it and selecting **Assign**, SAS Enterprise Guide assigns the library using the META engine in its default mode. That means that the library contents match the tables and columns registered in metadata, and the views that you see match the permissions assigned to you within metadata. It also means that the assigned library is effectively read-only, since the META engine will not allow you to change add/modify tables.

Figure 2 shows another example view, this time from Tools->SAS Enterprise Guide Explorer. It shows the mix of metadata (such as Description and SAS Folder Location) and the traditional library information (such as Libref to use for your SAS programs, and whether the library is ACCESS=READONLY).

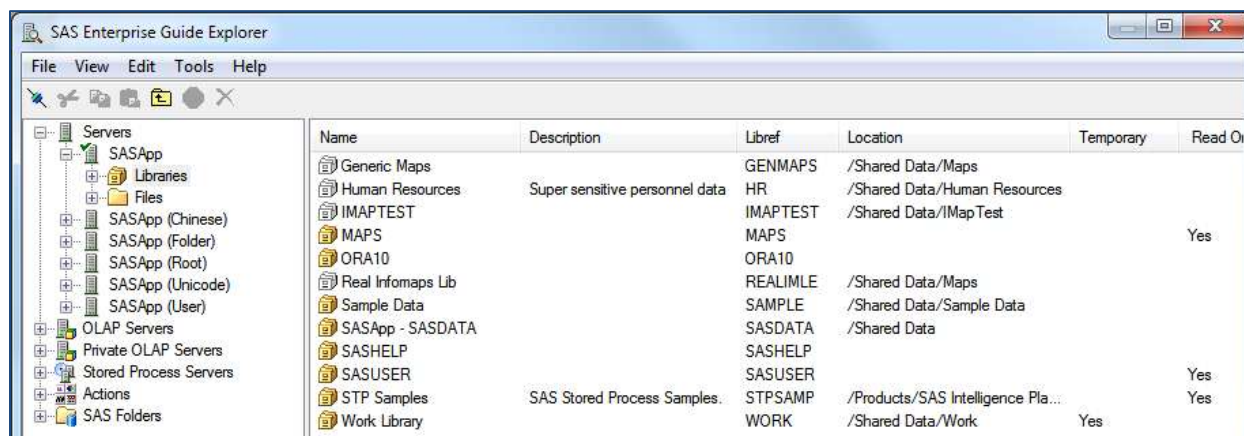


Figure 2. An example view from Tools->SAS Enterprise Guide Explorer.

CONTROLLING LIBRARY ASSIGNMENT BEHAVIOR

It is possible for you to control how SAS Enterprise Guide assigns metadata-defined libraries. You can specify the behavior with Tools->SAS Enterprise Guide Explorer, which offers an administrative view of SAS servers and libraries similar to SAS Management Console.

Note: For more information about managing libraries using SAS Enterprise Guide Explorer, see *Administering SAS Enterprise Guide 4.3*, available from <http://support.sas.com/eguide>.

Figure 3 shows the default library properties for an example metadata library:

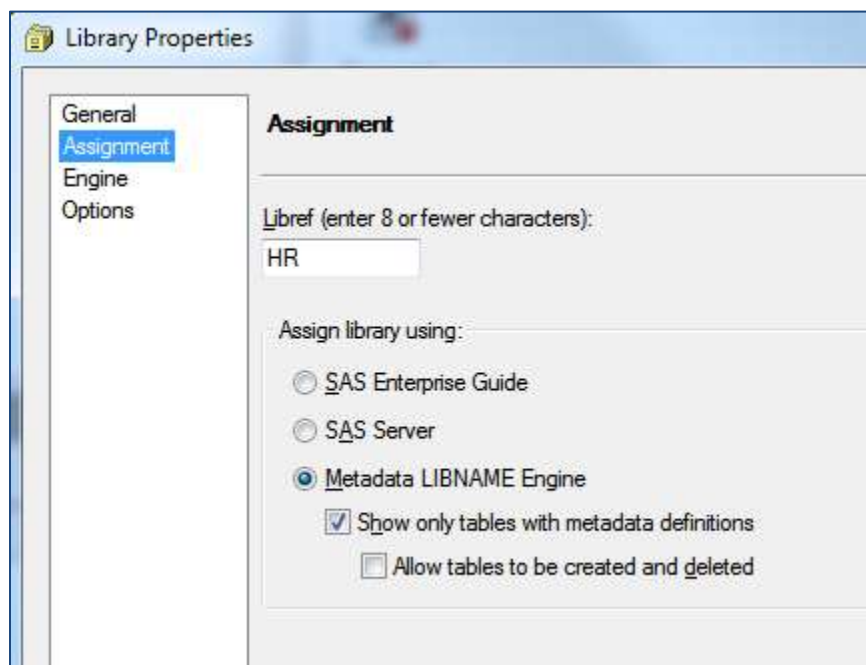


Figure 3. Default properties for a library as shown in SAS Enterprise Guide

Here is an explanation of the “Assign library using” modes.

SAS Enterprise Guide

SAS Enterprise Guide reads the library definition from metadata and assigns the library using the native engine. No table/column level metadata is read/used in this library -- it's as if you submitted a libname statement that used the underlying native engine, bypassing the META engine and any permissions that are specified in metadata.

SAS Server

SAS Enterprise Guide does not assign the library, but treats it as preassigned. This means that the library, even though its definition exists in metadata, is actually assigned in an autoexec or via the METAAUTOINIT mechanism.

Metadata LIBNAME Engine

SAS Enterprise Guide uses the META engine to assign the library. This option offers three main modes:

- Show only tables with metadata definitions (first checkbox selected) -- only tables that have metadata defined actually show in the library. This uses the METAOUT=ALL option on the META engine. In this mode, the library is always Read Only, *unless* you also check the next box, "Allow tables to be created and deleted".
- Show only tables with metadata definitions, and allow updates to those tables (both checkboxes selected). This uses the METAOUT=DATAREG option on the META engine. In this mode you can read, update, and delete the tables and columns that are defined in metadata, but any new tables that you create will not appear until you register them in metadata.
- Show all physical tables (checkboxes not selected) -- this mode shows all physical tables in the library. Metadata READ permissions are still enforced when you try to open data. This uses the METAOUT=DATA option on the META engine. In this mode, it is possible to add/modify/delete tables within the library. That is, the library is not Read Only.

When you make changes to the Assignment page of the library definition in SAS Enterprise Guide Explorer, the changes are reflected in metadata as Extended Attributes on the library. You can view/modify these extended attributes using SAS Management Console, if desired.

To view the extended attributes in SAS Management Console, right-click on the library name and select **Properties**, then click on the Extended Attributes tab. Figure 4 shows an example:

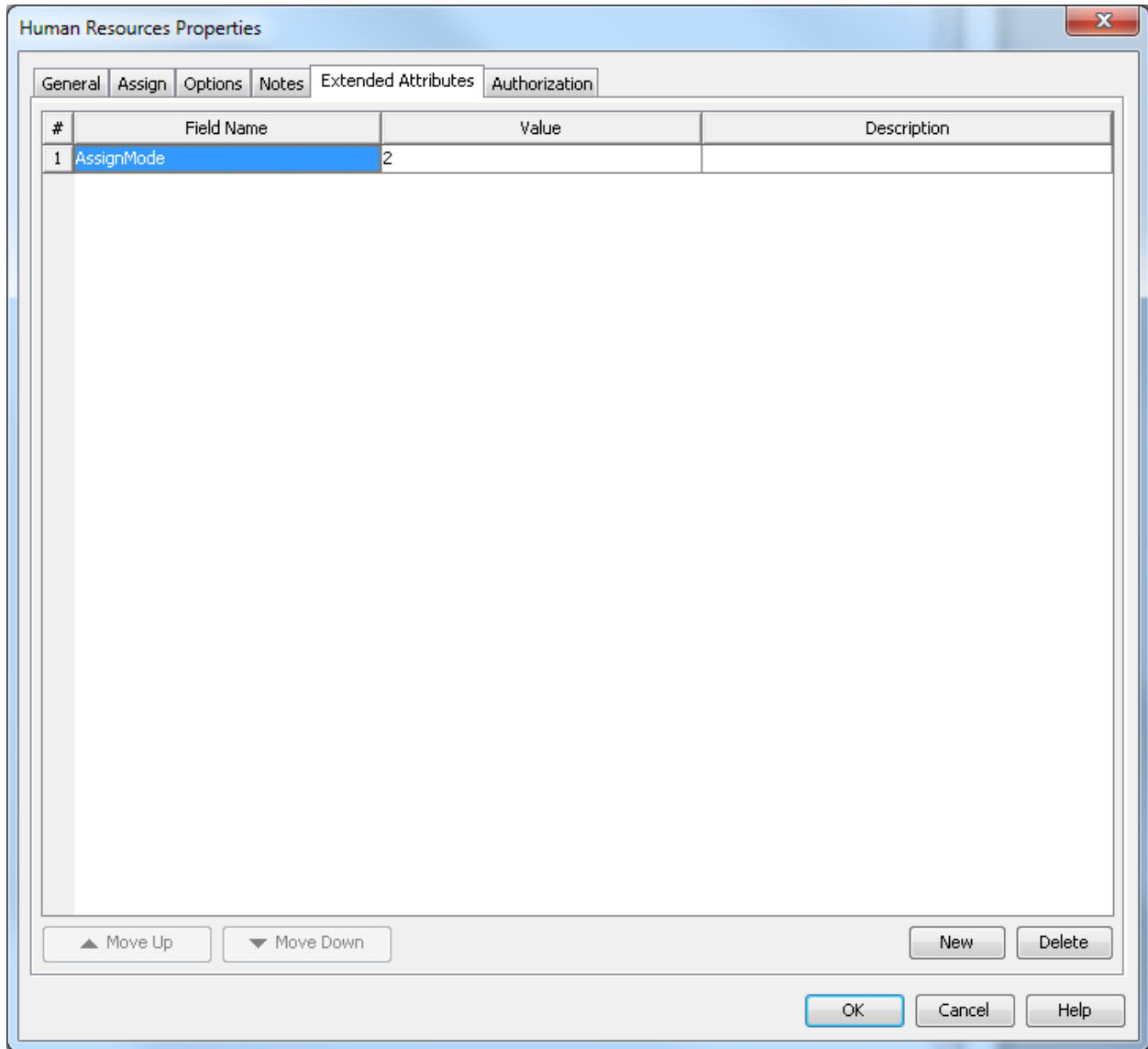


Figure 4. AssignMode properties as shown in SAS Management Console

The **AssignMode** name is the attribute that controls the assignment behavior. Valid values are:

0	Assign using SAS Enterprise Guide
1	Assign using the META engine, METAOUT=ALL (default META engine behavior)
2	Assign using the META engine, METAOUT=DATA
3	Assigned by the SAS Server (preassigned)
4	Assign using the META engine, METAOUT=DATAREG

FREQUENTLY ASKED QUESTIONS ABOUT LIBRARIES IN SAS ENTERPRISE GUIDE

I am trying to save output tables to a library, but I get an error message that says “you cannot create tables in this mode.” What does that mean and how can I fix it?

When accessing a library defined in metadata, SAS Enterprise Guide will use the META engine in “METAOUT=ALL” mode, by default. This means that the engine will enforce restrictions to prevent additions and modifications that would cause library metadata to become out of sync with the physical tables.

There are a couple of options to allow output tables to be added to the library.

1. Use SAS Enterprise Guide Explorer to change the library assignment behavior, unchecking the “Show only tables with metadata definitions” checkbox. This tells SAS Enterprise Guide to use the METAOUT=DATA mode, which allows you to add/modify tables.
2. Use SAS Enterprise Guide Explorer to change the library assignment behavior to “Assign library using SAS Enterprise Guide.” This tells SAS Enterprise Guide to bypass use of the META engine and assign the library using the native engine and options that are defined in metadata. This mode bypasses the metadata view of the library altogether, ignoring permissions that might have been set on individual tables and columns in metadata. Instead, all permissions are dependent on the security set at the system level (for folder/file-based libraries) or DBMS level (for SAS/ACCESS libraries).

Why do I get an “access denied” error message when trying to save output tables to a library assigned with METAOUT=DATA?

It’s possible that the authenticated user does not have write permissions to the physical location that the library maps to. If the library is defined to a file path, make sure that your user account has write permissions to the server folder. Without proper system-level permissions, you might see an error similar to this:

ERROR: User does not have appropriate authorization level for library <library>.

Note that granting or denying read/write permissions in SAS metadata affects only the view of tables accessed with the META engine. The metadata permissions do not override other permissions that have been set at the system level (for example, permissions on a system folder or in a database system). You can use metadata permissions to restrict a user’s view of the library contents. Use system or DBMS permissions to prevent unauthorized access to sensitive information.

My users can see tables that they shouldn’t have access to in a certain library. I changed the metadata permissions to deny them access; why aren’t the permissions working?

There are a couple of things to check here:

1. First, ensure that the library is assigned using the META engine, which enforces the metadata-defined view of the library contents. This is the default behavior in SAS Enterprise Guide, but it can be affected by library settings controlled in SAS Enterprise Guide Explorer. The SAS workspace server can also assign the library using the native engine (circumventing the META engine), when the library is marked as “Pre-assigned” and the METAUTOINIT object spawner option is in effect on the server definition.
2. If the META engine is used in METAOUT=DATA mode, ensure that the tables that you want to prevent access to are tagged with Deny for Read, but not Deny for ReadMetadata. If you select Deny for ReadMetadata, the META engine is unable to determine the intended Read permissions for the table. It might seem counterintuitive, but the META engine requires the ability to ReadMetadata in order to determine the proper Read/Write access control for the table.

WARNING:

If you use the object server parameter METAUTOINIT in combination with the SAS Library advanced option “Library is pre-assigned”, the SAS workspace will assign the library automatically when it initializes, and it will use the native engine (for example, BASE or ORACLE) for the assignment (not the META engine). This will override any settings you specified in SAS Enterprise Guide Explorer for the library Assign Mode. **By default in SAS 9.2, SAS workspace servers are configured to start with METAUTOINIT.**

I have denied ReadMetadata and Read access to a table for my user, but when he views the library in SAS Enterprise Guide, he can not only see the table, but also the records. Why aren’t the permissions working?

If you have assigned the library with the METAOUT=DATA option, remember that SAS Enterprise Guide will display all data sets in the physical location and not just the tables that have been registered in the metadata. Therefore, even though you have ReadMetadata, the table will appear. Also, since you have denied ReadMetadata, SAS Enterprise Guide is not able to view the metadata for that table on of your user, and therefore never sees that the Read permission has been. You must grant ReadMetadata in order for the denied Read to be enforced.

If the objective is to allow the user to update tables he has permissions for, but deny read permissions for those that he does not, consider using METAOUT=DATAREG instead.

I used SAS Enterprise Guide to create some output tables for use in information maps. When I view the library in SAS Information Map Studio, the new tables don't appear there. Why not?

Before tables can be used in metadata-based objects such as information maps, you must first add the table definitions to metadata. You can achieve this by using the Import Tables feature in SAS Management Console, or within a SAS program using the METALIB procedure to synchronize metadata definitions with the physical contents of the library. You can also use Tools->Update Library Metadata, which can generate the PROC METALIB program for you.

How can I make sure that my users see only the metadata-defined view of a data library?

To guarantee that the library is assigned using the META engine, you can add a LIBNAME statement to your SAS workspace autoexec file.

For example, a LIBNAME statement like this one:

```
libname hr meta library="Human Resources";
```

will ensure that all users of that SAS workspace server see only the metadata-defined view of the "Human Resources" library, with their particular permissions applied. This library will be immediately assigned within the SAS session (meaning it will show as a yellow icon in the server view, not white).

Important: make sure that the libref used in the LIBNAME statement ("HR" in the above example) matches the libref defined in the metadata for the library. If the librefs don't match, your users will see two views of the library: one assigned with the libref in your autoexec, and the other as a potential library, not yet assigned, using the library name defined in metadata.

When I view the properties of a library in SAS Enterprise Guide, I see the native engine rather than META? Why? How can I determine for certain that the META engine is used?

When you right-click on a library in the server view and select Properties, SAS Enterprise Guide shows you details about the library as defined in metadata. Figure 5 shows an example view:

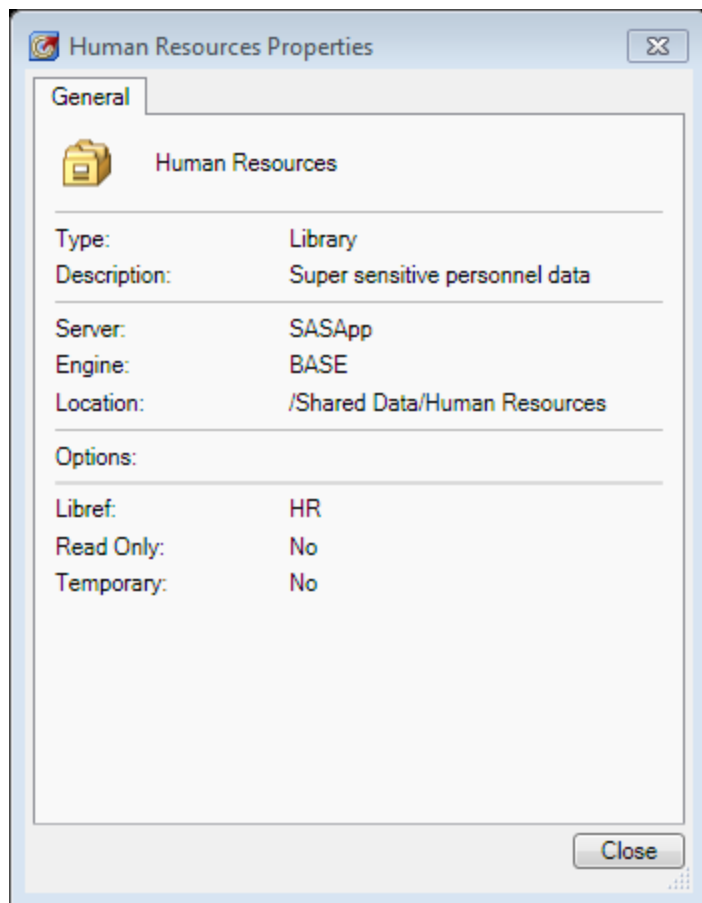


Figure 5. Library properties for a metadata library as viewed in SAS Enterprise Guide

Even though you see the BASE engine listed here, the default behavior is to use the META engine to assign the library when you access it. Notice that there is no physical path listed for this library, only its Location within the SAS Folders.

To verify that the META engine is used, right-click on the library and select Assign. Then, from a SAS code window, submit the following statement:

```
libname _all_ list;
```

The log will show the details about how the library is defined. For example:

```
NOTE: Libref=   HR
      Scope=   IOM ROOT COMP ENV
      Engine=  META
      Physical Name= C:\SAS\Data\HR
      Filename= C:\SAS\Data\HR
```

From this log you can see that the META engine, not the BASE engine, is controlling access to the library. Therefore you can expect the metadata-defined view of the library to be in place.

I created a stored process that references a data library that I accessed within Enterprise Guide. When I run the stored process, it fails to write data to the library. However, when I run the code within Enterprise Guide, it works. Why?

If you changed the library definition within Enterprise Guide to use METAOUT=DATA (AssignMode=2), then Enterprise Guide tasks will be able to write tables to the library. However, when running in an environment outside of Enterprise Guide's control, such as within a stored process, the default behavior of the META engine (METAOUT=ALL) may apply. This difference can occur even when you are running the stored process from within

SAS Enterprise Guide.

When you create a stored process that must access or write data tables in a metadata-controlled library, you cannot rely on the “Assign mode” behavior specified within SAS Enterprise Guide Explorer. The Assignment options affect the behavior within SAS Enterprise Guide and SAS Add-In for Microsoft Office, but they don’t affect the behavior within the SAS Stored Process server or other SAS products.

How can I prevent an end user (not an administrator) from adding or updating SAS library definitions in metadata?

Use SAS Management Console to deny Write Metadata (WM) permission on the SAS Application Server context (for example “SASApp”), for the end users you want to restrict. You can apply this Deny WM permission to individual users or groups. While this alone does not prevent users from creating new library definitions, it does prevent them from being able to associate library definitions with the SAS Application Server. In practice, this prevents them from creating/updating libraries that can be used from within SAS Enterprise Guide.

You can also use SAS Management Console to deny WM permission to individual SAS library definitions, which will prevent end users from updating the table metadata within a library.

OTHER RESOURCES

Administering SAS® Enterprise Guide 4.3, available from <http://support.sas.com/eguide>

SAS® 9.2 Intelligence Platform: Data Administration Guide, available from <http://support.sas.com/documentation/cdl/en/bidsag/61236/PDF/default/bidsag.pdf>

SAS® 9.2 Intelligence Platform: Desktop Application Administration Guide, available from <http://support.sas.com/documentation/cdl/en/bidaag/61231/PDF/default/bidaag.pdf>