Paper 420-2013

What SAS® Administrators Should Know About Security and SAS® Enterprise Guide®

Casey Smith, SAS Institute Inc., Cary, NC

ABSTRACT

SAS® Enterprise Guide® is a flexible and powerful tool in the hands of your users. However, as every superhero knows, with power comes responsibility. As an administrator, you want to ensure various groups of users have access to the specific resources they need to be as productive as possible, while at the same time protecting company assets and minimizing risk. This paper explores various security considerations from the SAS Enterprise Guide perspective, such as authentication, authorization, user administration, access management, encryption, and role-based availability of application features.

INTRODUCTION

Security is often a thankless responsibility. Awards aren't handed out for prevention. Security is noticed only when inconvenient or breached. If successful, no one notices, and you can pat yourself on the back. Nevertheless, security in today's connected world, full of constantly evolving threats, is more important than ever.

Many security aspects in SAS Enterprise Guide are not specific to this application. SAS Enterprise Guide is one of many components that participate in the robust security model provided by the SAS Intelligence Platform. Security in the SAS Intelligence Platform is a large topic, only some of which we cover in this paper. I highly recommend reading *SAS Intelligence Platform: Security Administration Guide* for more complete information about this topic. This paper looks at security from a SAS Enterprise Guide viewpoint and covers the security aspects that are most relevant to SAS Enterprise Guide.

THE ECOSYSTEM

Knowing the relationship between components in a system enables you to better assess the security requirements. SAS Enterprise Guide is a Microsoft Windows client application that communicates with various SAS servers and third-party data servers to provide querying, reporting, and advanced analytics.

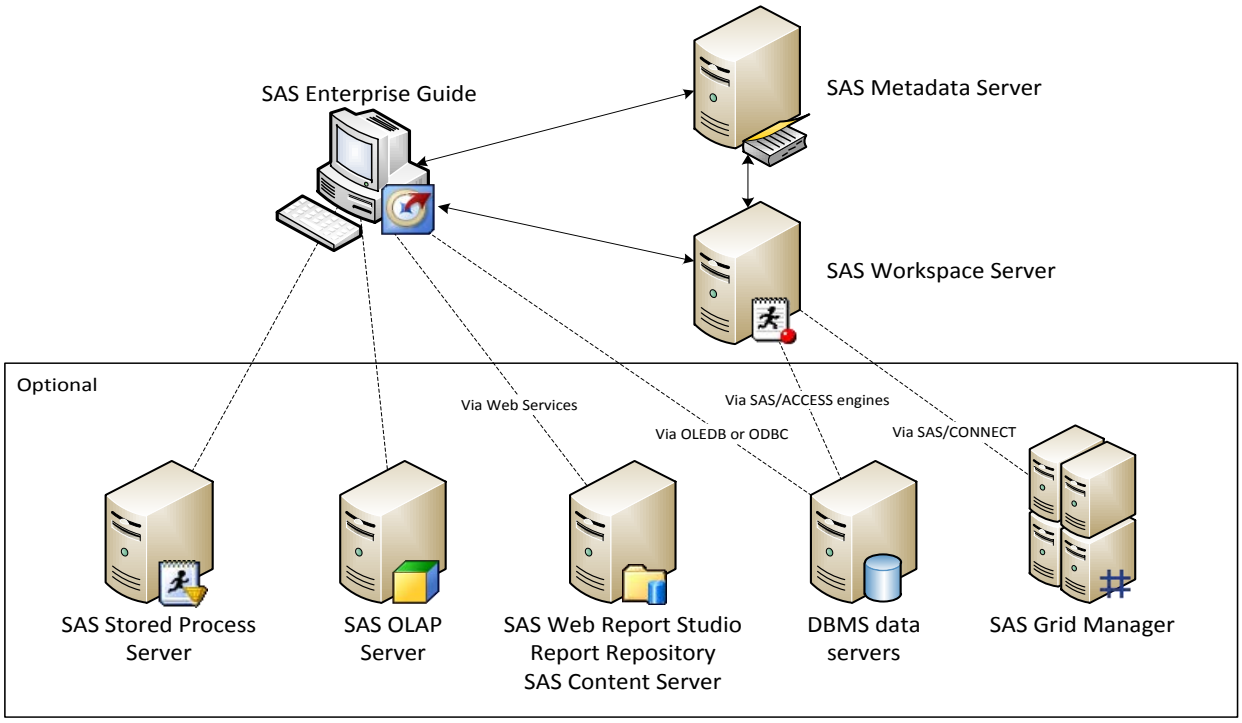

Figure 1. The SAS Enterprise Guide Client/Server Environment

CLIENT/SERVER COMMUNICATION

SAS Enterprise Guide interacts with SAS servers via the SAS Integrated Object Model (IOM), a part of SAS® Integration Technologies, and the Microsoft Component Object Model (COM). A key feature of SAS Integration Technologies is the IOM Bridge for COM component, which allows Microsoft Windows applications such as SAS Enterprise Guide to transparently communicate with SAS servers on different platforms (such as UNIX and z/OS, in addition to Microsoft Windows).

Network Protocols

At a lower level, the IOM Bridge for COM component communicates using the proprietary IOM Bridge protocol. The IOM Bridge protocol operates in the application layer of the TCP/IP stack. Because servers in the SAS Intelligence Platform communicate with clients and other servers by using TCP/IP, each server listens on a particular port or ports for incoming requests. Maintaining open ports, particularly outside the protection of a firewall, has associated risks from individuals and applications that might attempt to exploit them.

Here is a list of default ports for SAS servers and spawners commonly used by SAS Enterprise Guide:

SAS Server	Port
SAS Metadata Server	8561
SAS Object Spawner	8581
SAS Workspace Server	8591
SAS Stored Process Server	8601
SAS OLAP Server	5451

Figure 2. Default Ports for SAS Servers and Spawners Commonly Used by SAS Enterprise Guide

The connections established on these ports are the lifelines between SAS Enterprise Guide and the SAS servers. When a connection is severed, any state information in that server session is lost. For example, suppose you had temporary data in the Work library of a workspace server session and then a network outage occurred. You would see a message from SAS Enterprise Guide indicating you had lost your connection to the server. You could establish a new connection once the network was available. However, your temporary data would no longer be retrievable.

Over-the-Wire Encryption

Client/server communications over a network, especially when they include passwords and other sensitive data, present an obvious security risk. It is fairly easy with a tool such as Wireshark (a popular network protocol analyzer) to intercept and view network traffic. Fortunately, SAS provides options for securing over-the-wire communications.

SAS servers and SAS Enterprise Guide always have the ability to use a SAS proprietary weak encryption algorithm, aptly called SAS Proprietary. SAS Proprietary is included with Base SAS® software. Strong encryption support requires SAS/SECURE™ software to be licensed and installed on both the server and client. (Starting in 9.4, SAS/SECURE will also be included with BASE SAS.) SAS/SECURE software enables SAS Integration Technologies to use encryption algorithms that are available through the Microsoft Cryptographic Application Programming Interface (CryptoAPI). To communicate with a SAS server configured for strong encryption, each SAS Enterprise Guide client must have the SAS/SECURE client installed. The SAS/SECURE client installs the tcpdcapi.dll file that is necessary for the IOM Bridge for COM component to use the CryptoAPI algorithms to communicate with the IOM server. The file is installed to the shared files location on the client.

SAS Enterprise Guide does not specify encryption settings; it conforms to the requirements of the server. You specify the encryption algorithm (how traffic is encrypted) and encryption level (which content is encrypted) when installing the metadata server. Those settings initially apply to all SAS servers, but can be changed independently. To change the over-the-wire encryption settings, you have to update the server configuration files and update the server metadata definitions as follows.

To update the server configuration files:

1. On the metadata server host operating system, navigate to the following path or the equivalent for your system: **`SAS/Config/Lev1/SASMeta/MetadataServer/`**
2. To change the encryption algorithm, add the desired NETENCRALG setting to the sasv9_usermods.cfg file.
3. To change the encryption level, copy the OBJECTSERVERPARMS line from the sasv9.cfg file to the sasv9_usermods.cfg file and edit the CEL value in the sasv9_usermods.cfg version.

For example, to encrypt all traffic with AES, add these lines:

```
-netencralg "AES"
-objectserverparms "cel=everything  {other-parameters}"
```

To update the server metadata definitions:

1. In SAS Management Console, on the **Plug-ins** tab, expand the **Server Manager** plug-in.
2. Expand the desired application server (for example, SASApp) and logical server (for example, SASApp – Logical Workspace Server).
3. Select the desired server definition (for example, SASApp – Workspace Server) underneath the logical server.
4. In the **Connections** pane, right-click the connection and select **Properties**.
5. On the **Options** tab, click **Advanced Options**.
6. On the **Encryption** tab, specify the desired encryption algorithm and encryption level.

Note: All algorithms are listed regardless of whether you have SAS/SECURE. Do not select an algorithm other than SAS Proprietary in the dialog box, nor for the NETENCRALG option in the configuration file, unless you have SAS/SECURE licensed and installed.

Figure 3. Server Encryption Algorithm and Encryption Level

Be mindful of the trade-off between security and performance. By default, only transmitted credentials are encrypted. Securing all communications and securing with strong encryption both reduce performance as they require extra processing power for encrypting and decrypting on each end.

WINDOW INTO THE SAS WORLD

Like most other SAS client applications, the primary point of integration of SAS Enterprise Guide with the rest of the SAS environment is through the SAS Metadata Server. The SAS Metadata Server is the central hub that defines and coordinates the resources in your environment, including users, servers, libraries, data sources, permissions, and more.

In SAS Enterprise Guide, a connection profile defines the information needed to establish a connection to a SAS Metadata Server. You access the Connections dialog box from the connection icon and hyperlink in the bottom right corner of the application.

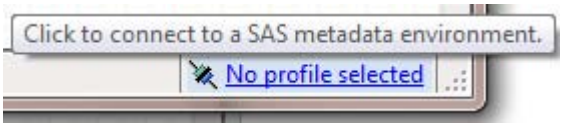

Figure 4. Active Connection Profile Link

A connection profile contains a name, an optional description, the SAS Metadata Server host machine information, and the desired authentication method.

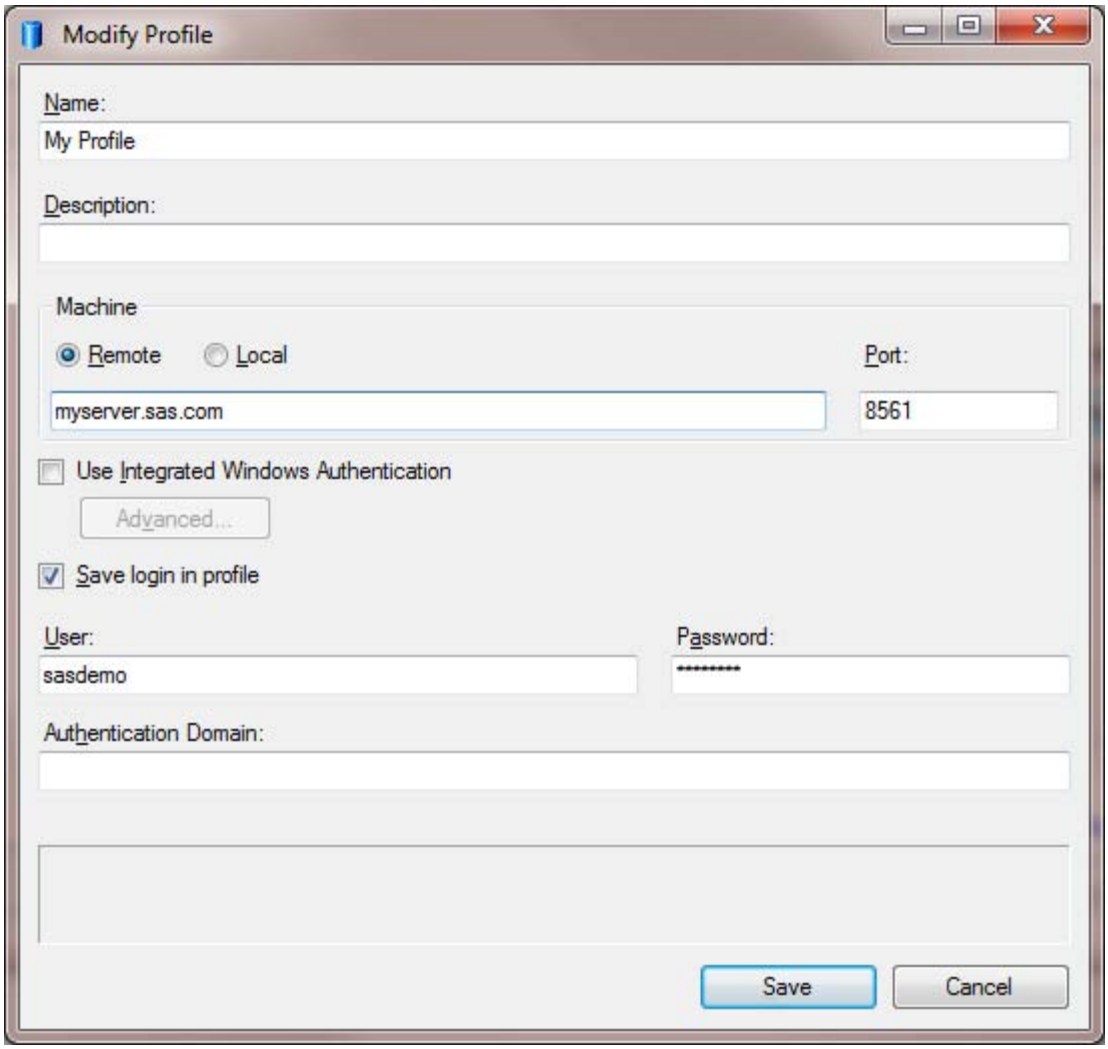

Figure 5. Modify Connection Profile Dialog Box

Connection profiles are stored in an XML file called ConfigurationVnn.xml (where nn is the SAS Enterprise Guide version number) in the user's application data area (for example, %appdata%\SAS\MetadataServerProfiles\ConfigurationV51.xml). Note that this file is shared by several applications. It is used by SAS Enterprise Guide, the SAS® Add-In for Microsoft Office, and all JMP® products.

Also notice in the following XML that connection profiles might or might not store a user name and password depending on whether **Save login in profile** is checked. A policy in the SAS Metadata Server can control at a system-wide level whether credentials can be stored in client profiles. We'll discuss this in a little more detail farther down.

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration LastWrittenBy="Engine Configuration Manager">
  <Profiles>
    <Profile>
      <Name>bigband (IWA)</Name>
      <Description />
      <Type>OMS</Type>
      <HostName>bigband</HostName>
      <Port>8561</Port>
      <UseIWA>True</UseIWA>
      <SecurityPackage>Negotiate</SecurityPackage>
      <ServicePrincipalName></ServicePrincipalName>
      <SecurityPackageList>Kerberos,NTLM</SecurityPackageList>
```

```
    </Profile>
    <Profile>
      <Name>bigband (sasdemo)</Name>
      <Description />
      <Type>OMS</Type>
      <HostName>bigband</HostName>
      <Port>8561</Port>
      <UseIWA>False</UseIWA>
      <SaveLogin>True</SaveLogin>
      <User>sasdemo</User>
      <Password>{sas002}6FE4B62647E0E59721B8CAD44A12345C</Password>
      <DefaultServer>SASApp</DefaultServer>
    </Profile>
  <Settings>
    <ActiveProfile>bigband (IWA)</ActiveProfile>
    <GenerateLocalServer>False</GenerateLocalServer>
  </Settings>
</Configuration>
```

Figure 6. Contents of Connection Profiles XML File

Now that we have reviewed the environment and how we communicate and plug into the environment, let us examine how the connection profile information is actually used to establish a connection.

AUTHENTICATION

A primary tenet of security is being able to establish and confirm a user's identity. In order to make access distinctions and track user activity, a security system must know who is making each request. The question is more than just "Who are you?", but also "Are you who you say you are?"

To support authentication, a certain level of user administration is usually required. The primary user administration task is to define SAS user identities in the SAS metadata for each of your users and link their external account IDs to their SAS identity. When that is done and the user authenticates using one of the linked external accounts, they can access the metadata using the permissions associated with the SAS identity.

Before communication between SAS clients and servers can begin, the user must be authenticated. The SAS Intelligence Platform offers a number of authentication mechanisms, each with different advantages and disadvantages. Many environments require a mix of authentication mechanisms to achieve desired results. SAS Enterprise Guide facilitates authentication to the metadata server in one of two ways—by providing credentials or by using Integrated Windows Authentication.

AUTHENTICATING WITH CREDENTIALS

Providing credentials in the form of a user name and password is something every computer user is probably too familiar with. In the Modify Connection Profile dialog box in Figure 5, notice that you have the ability to specify a user name and password. When specified, those credentials are passed to the SAS Metadata Server and are authenticated using the authentication provider configured on the SAS Metadata Server. The default authentication mechanism used by the SAS servers is credentials-based host authentication, meaning the credentials supplied are authenticated on the host machine on which the SAS server is running.

Benefit

No configuration is required. It enables users to use the same credentials that they use in your general computing environment.

Limitations

- You require your users to provide an initial user ID and password.
- Credentials are transmitted over the network.
- When connecting to a workspace server on Windows, the **Log on as a batch job** Windows privilege is required.

Figure 7 shows how credentials are routed for authentication, verified by the host, and matched to a SAS identity.

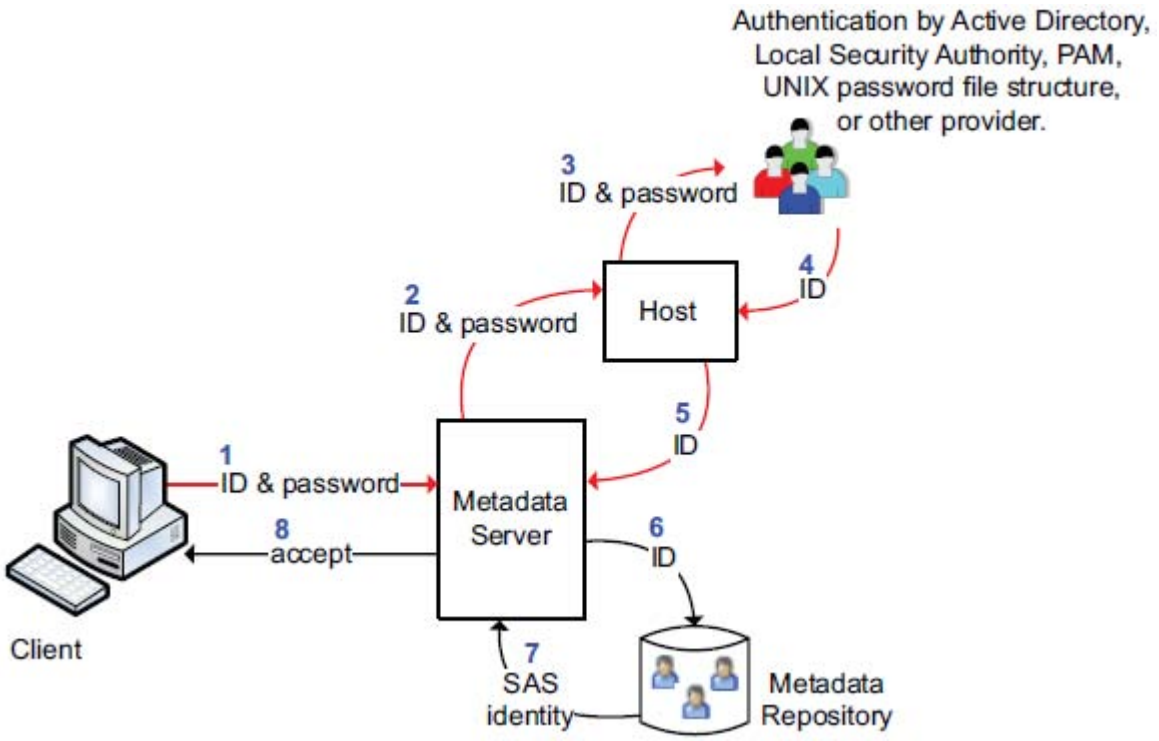

Figure 7. Host Authentication (Credentials-Based)[1]

Client-Side Storage of Passwords

As a convenience, to prevent users from having to supply credentials every time they log on to a SAS Metadata Server, SAS Enterprise Guide can save metadata credentials in your connection profile. (See Figure 5 and Figure 6.)

Notice in the XML in Figure 6 that the password begins with "{sas002}". This indicates the password is encrypted using the SAS Proprietary encryption algorithm. Even though the password is encrypted, it is subject to replay attacks. Therefore, it is important to restrict access to the connection profiles XML file. The fact the file lives in the user's application data area, and the Microsoft Windows host access restrictions that go along with that, is usually sufficient protection.

Some administrators might prefer to restrict clients from saving credentials in their profile. An administrator can configure this by setting the SASSEC_LOCAL_PW_SAVE= option to "N" (or "0" or "F") in the metadata server's omaconfig.xml file. This server-side setting affects all the SAS client applications connecting to that metadata server. It is not specific to SAS Enterprise Guide.

Note: Changes to the SASSEC_LOCAL_PW_SAVE= option take effect after the metadata server is restarted. The **Save login in profile** check box will still be available in the profile dialog box in SAS Enterprise Guide, because the metadata server setting is not discovered until a connection is made. Once a connection is established, if SASSEC_LOCAL_PW_SAVE= option is off, the **Save login in profile** check box will be unchecked and the credentials will not be saved beyond the current SAS Enterprise Guide session. Likewise, any credentials already stored in a client profile are not removed until the next successful connection for that profile. Then, after SAS Enterprise Guide is restarted (which clears the credentials cached in the session) you will always be initially prompted for credentials when connecting with that profile.

AUTHENTICATING WITH INTEGRATED WINDOWS AUTHENTICATION

Because you already have to log on to your Windows desktop to use SAS Enterprise Guide, it would be nice if authenticating servers from SAS Enterprise Guide would just piggyback on your prior successful Windows authentication. Integrated Windows Authentication (IWA) enables this and provides support for single sign-on (SSO). Single sign-on enables users to access multiple servers seamlessly. IWA is another form of host authentication, but

[1] SAS Institute Inc., *SAS 9.3 Intelligence Platform: Security Administration Guide*. (Cary, NC: SAS Institute Inc., 2011), 97.

does not rely on credentials. Rather, it uses an authenticated token based on your Windows logon.

Benefits

- Bypasses the initial logon attempt.
- No user credentials are transmitted.
- Users do not need the **Log on as a batch job** Windows privilege.

Limitations

- All participating clients and servers must authenticate against the same Windows domain (or against domains that trust one another) – only supported on SAS servers on Windows and UNIX.
- Primarily used for connections to the metadata server and the workspace server.
- If you use IWA for the metadata server, there are no cached credentials from an initial logon. Therefore, it is a good idea to configure IWA for the workspace server as well. Otherwise, the user will be prompted for credentials when connecting to the workspace server.

Note: Support for IWA on UNIX was added in SAS 9.3 and requires some special configuration, including requiring third-party software.

Figure 8 shows how a connection is accepted using IWA. An authenticated token is requested and passed between the client, target server, and the Windows authentication authority. Once the token is verified, the user ID is matched to a SAS identity and the connection is accepted.

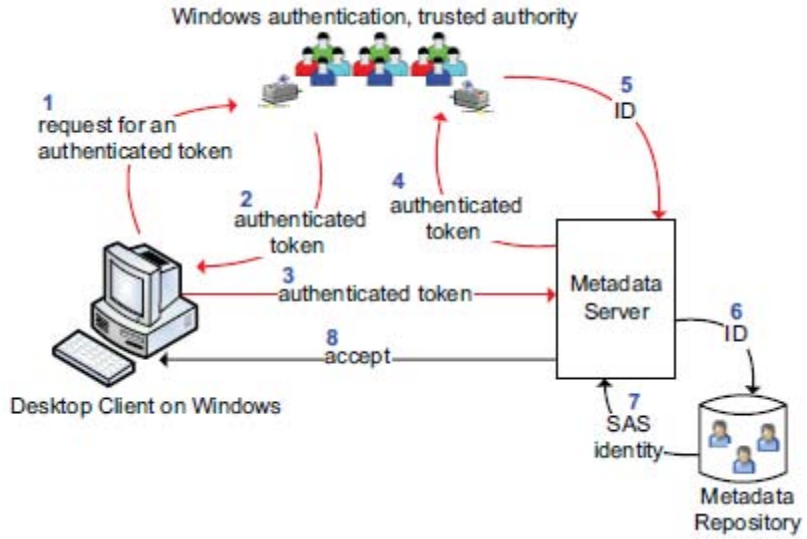

Figure 8. Integrated Windows Authentication[2]

Configuring to Use Integrated Windows Authentication

Checking **Use Integrated Windows Authentication** in your SAS Enterprise Guide profile (see Figure 5) configures SAS Enterprise Guide to use IWA to connect to the metadata server. Checking the option also enables the **Advanced** button and disables the credentials fields. The **Advanced** button shows advanced IWA security package settings. (See Figure 9.) These advanced IWA settings rarely need to be changed, except in very specific configurations.

[2] SAS Institute Inc., *SAS 9.3 Intelligence Platform: Security Administration Guide*. (Cary, NC: SAS Institute Inc., 2011), 98.

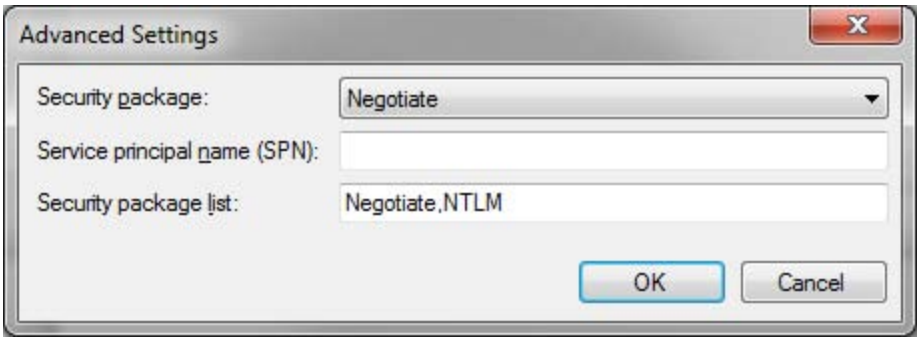

Figure 9. Advanced Integrated Windows Authentication Settings

There are similar settings that can be specified in SAS Management Console to configure workspace servers to use IWA. To instruct a workspace server to use IWA, on the **Options** tab of the logical workspace server **Properties**, select **Host** authentication service and specify **Negotiate** (or Kerberos or NTLM if you know which specific one to use) as the security package.

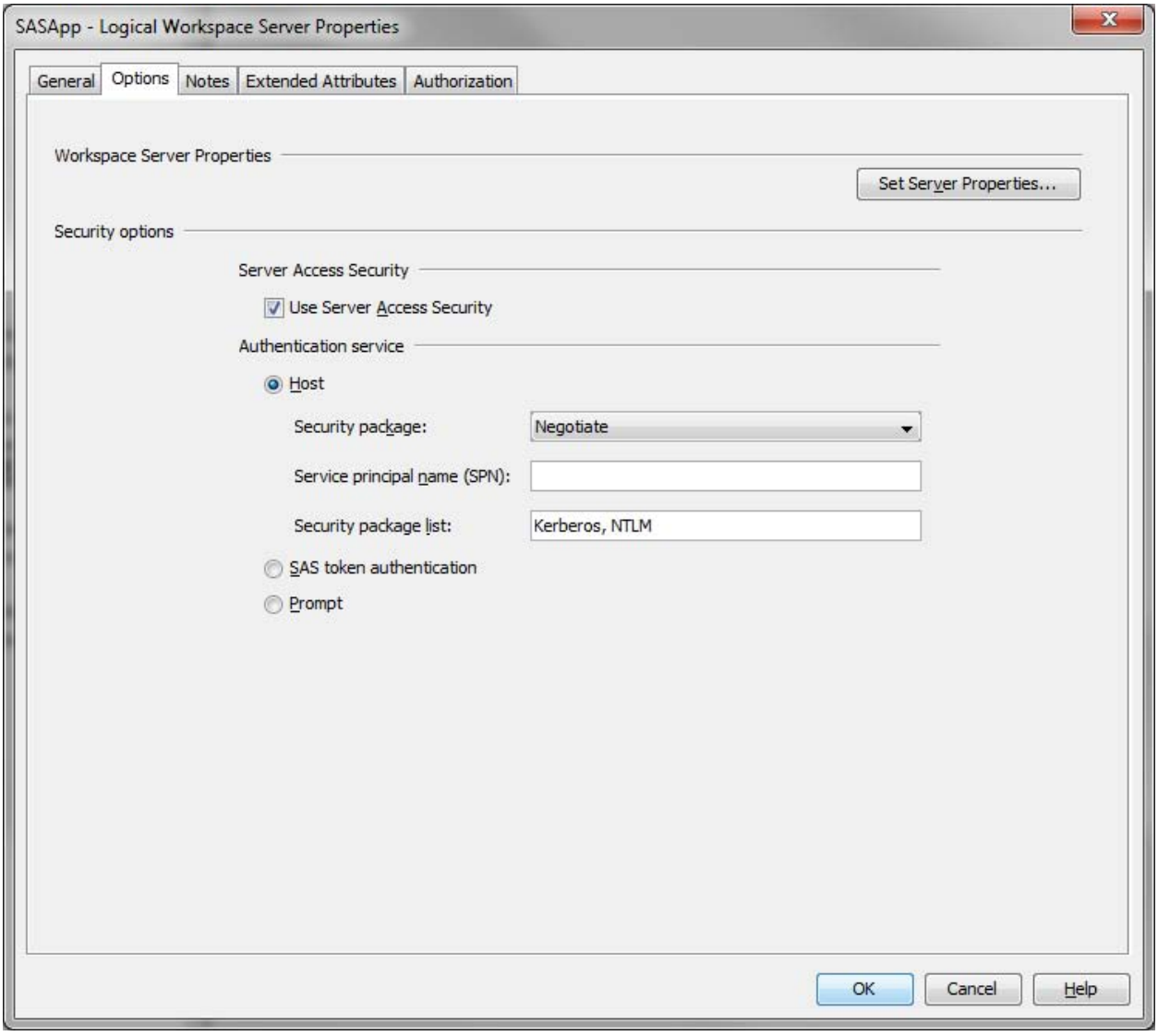

Figure 10. SAS Workspace Server Security Options in SAS Management Console

There are other authentication mechanisms that can be used in SAS Enterprise Guide, but are less apparent. For example, SAS Enterprise Guide can connect to most SAS servers using SAS Token Authentication, which provides single sign-on support. SAS Token Authentication requires an existing metadata server connection. Therefore, it cannot be used to connect to a metadata server.

CONNECTED

The result of a successful authentication is the acceptance of a connection, in this case to a SAS Metadata Server. Then, the SAS Enterprise Guide connection link shows the name of the active connection profile. Positioning your mouse pointer over the connection link displays in a tooltip the SAS Metadata Server machine information, the user ID (if credentials) or method (if IWA) used to connect, and the user identity that was resolved in metadata.

Figure 11. Active Connection Tooltip with Integrated Windows Authentication

Figure 12. Active Connection Tooltip with Credentials

AUTHORIZATION

Once you have successfully authenticated and established a connection to your SAS environment from SAS Enterprise Guide, a proprietary, metadata-based authorization layer will manage your access to most of your SAS resources. The metadata layer supplements protections from the host environment and other systems.

As with authentication, the authorization model is provided by the SAS Intelligence Platform, is not specific to SAS Enterprise Guide, and is a large topic. We will scratch the surface only.

SAS Enterprise Guide rarely explicitly checks metadata authorizations prior to attempting an operation. Rather, SAS Enterprise Guide uses trial and error. It attempts an operation and if not authorized, handles the error.

Access management is usually performed interactively in SAS Management Console. However, authorization settings can also be defined or queried programmatically.

Likely, the most common authorization tasks in SAS Enterprise Guide are controlling access to data and servers for specific users or groups of users.

CONTROLLING ACCESS TO DATA

Data is often one of the most important digital assets to protect. End users with varying responsibilities have varying data access needs.

Library Access

SAS accesses data through libraries. SAS Enterprise Guide lists libraries that are defined in metadata as well as any libraries that are currently assigned on the server. Libraries denoted with a yellow icon are currently assigned, and libraries with an uncolored icon are defined in metadata, but not currently assigned. (See Figure 13.) Therefore, a library with an initial yellow icon is a pre-assigned library (perhaps by a LIBNAME statement in an autoexec file, or the pre-assigned option is checked in the library's metadata definition and the METAAUTOINIT option is turned on).

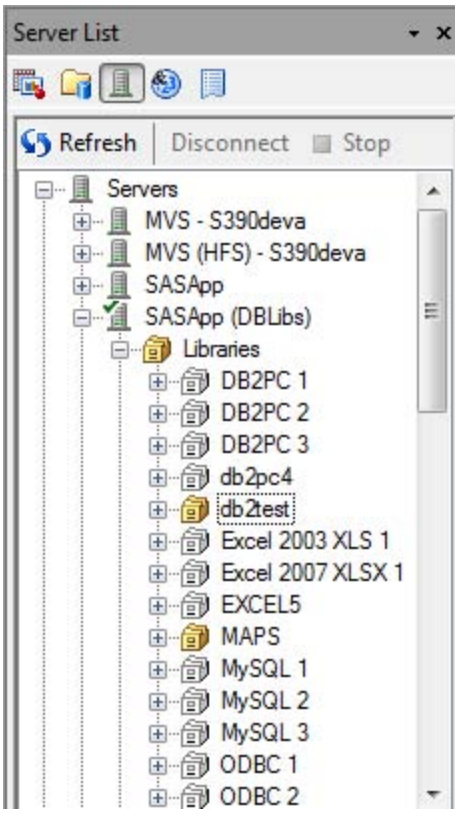

Figure 13. Libraries in Server List

Note: Defining a large number of pre-assigned libraries, particularly those that connect to remote database systems, can negatively impact the amount of time it takes to establish a new workspace server session.

You should define libraries, register data tables, and set authorizations in metadata to match your users to their appropriate level of data access.

Controlling Library Assignment Behavior

SAS Enterprise Guide can configure metadata libraries to be assigned using different mechanisms (see Figure 14), each with different characteristics and trade-offs.

To change a library assignment mode in SAS Enterprise Guide:

1. Select **Tools►SAS Enterprise Guide Explorer.**
2. From within SAS Enterprise Guide Explorer, right-click the desired server while it is disconnected and select **Libraries.**
3. Select the desired library and click **Modify.**
4. Select the **Assignment** page.

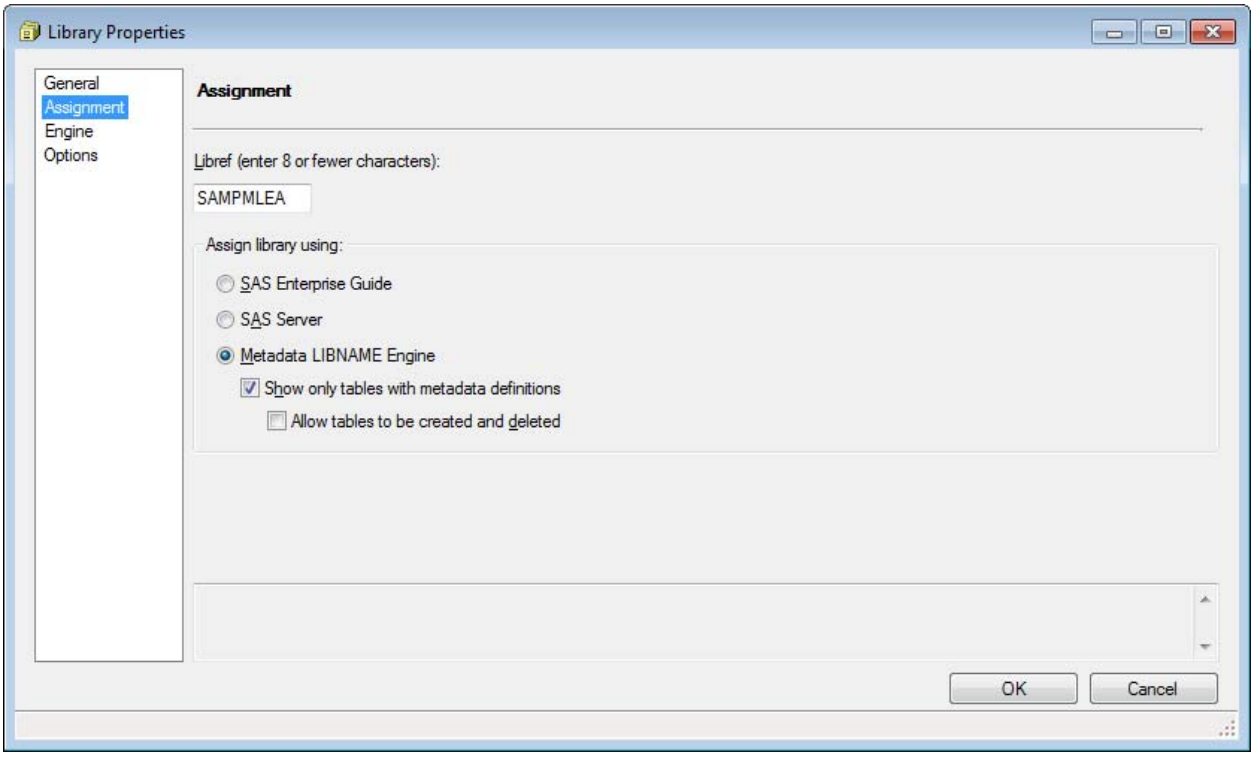

Figure 14. Library Assignment Options

Here is an explanation of the assignment modes.[3]

SAS Enterprise Guide

SAS Enterprise Guide reads the library definition from metadata and assigns the library using the native engine. No table or column level metadata is read or used in this library—it is 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 pre-assigned. This means that the library is actually assigned in an autoexec file or via the METAAUTOINIT mechanism, even though its definition exists in metadata.

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 check box selected)—Only tables that have metadata defined appear 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 check boxes 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** (check boxes 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, and delete tables within the library. That is, the library is not Read-Only.

[3] SAS Institute Inc., "What SAS Administrators Should Know about Libraries, Metadata, and SAS Enterprise Guide". (Cary, NC: SAS Institute Inc., 2011), 5.

Changes to the Assignment page of the library definition in SAS Enterprise Guide Explorer are reflected in metadata as Extended Attributes on the library. You can view and modify these extended attributes using SAS Management Console.
The **AssignMode** attribute controls the assignment behavior. Here are valid values:

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 (pre-assigned)
4	Assign using the META engine, METAOUT=DATAREG

Figure 15. AssignMode Options in SAS Management Console

Restricting Access to Physical Data through SAS Information Maps

As I noted in a previous paper (Smith 2012), SAS Information Maps contain business metadata that is applied to data sources. They enable users to work with a more user-friendly representation of the data, instead of working directly with the raw data sources. It is not uncommon for administrators to want to force their users to be able to see only the view offered through an information map, and not be able to access the source data.

The query executed by the INFOMAPS engine to return the information map data is done on your behalf on a standard workspace server, using your credentials. Therefore, for the information map to be able to retrieve the data, you have to be able to retrieve the data, which means you cannot restrict access to the physical data in this default configuration.

However, you can restrict access to the physical data by configuring server-side pooling. The SAS Pooled Workspace Server runs with the service account sassrv, so if you give sassrv host access to the data and deny host access to your users, the INFOMAPS engine will still be able to access the physical data through the SAS Pooled Workspace Server, but your end users will not be able to. Be aware this mediated access approach has a performance side effect, because the INFOMAPS engine will have to use remote access to access the data. When opening an information map in this configuration, an error in the log indicates that direct access was not possible and that it is reverting to remote access, which should then access the information map data just fine, although potentially slower. So, don't be alarmed by the error.

CONTROLLING ACCESS TO SERVERS

In the metadata layer, the following permissions are always enforced:

- the ReadMetadata permission (RM), which controls the ability to see an object
- the WriteMetadata permission (WM), which controls the ability to update or delete an object

Therefore, to prevent users from using or even seeing a server in SAS Enterprise Guide, you simply deny them the ReadMetadata permission on the server object in metadata. If you have different users or groups of users with different responsibilities, you can create multiple server definitions with different capabilities and only allow the ReadMetadata permission for the intended users of each server.

SECURITY FEATURES AND CONSIDERATIONS SPECIFIC TO SAS ENTERPRISE GUIDE

Though mentioned on several occasions that most security aspects that pertain to SAS Enterprise Guide are part of the larger SAS Intelligence Platform and not specific to the application, there are some security considerations that are specific to SAS Enterprise Guide.

PASSWORD-PROTECTING A PROJECT

SAS Enterprise Guide enables you to password-protect a project. When you open a password-protected project, you are prompted for the project password. The password must be supplied to be able to open the project.

To password-protect a SAS Enterprise Guide project:

1. Select **File►Project Properties.**
2. Select the **Security** page.
3. Provide a password in the **Password to open** text box and click **OK**.
4. Re-type the password to verify and click **OK**.

A SAS Enterprise Guide project file is really a Zip archive with a .egp file extension. You can confirm this by changing the extension to .zip, and then open the project file in a compression utility, such as WinZip. Password-protection of a SAS Enterprise Guide project is implemented by password-protecting the Zip archive. Be aware there are a number of Zip password recovery tools that might allow you to unlock a password-protected SAS Enterprise Guide project without knowing the password.

CREDENTIALS PERSISTENCE

When working with SAS password-protected data sets, a valid password must be supplied to read, alter, or update the data set. When a data set password is needed to access the data, SAS Enterprise Guide prompts you for the password. SAS Enterprise Guide provides a **Credentials Persistence** option on the **Security** tab in **Tools►Options** (see Figure 16) to control if and for how long the data set password is cached, so you can balance security and convenience. The **Clear** button enables you to empty any passwords in the cache.

The option also applies to project passwords. For example, if set to **Persist during EG session** (the default), you would be prompted for the project password for a password-protected project only the first time you opened it during that SAS Enterprise Guide session. So, if you then closed the project and reopened it in the same SAS Enterprise Guide session, you would not be prompted for the project password a second time.

Note: The **Credentials Persistence** option does not apply to server credentials or library credentials. It only applies to data set passwords and project passwords.

PASSWORD MASKING

Plain-text passwords, even when encoded or encrypted, are susceptible to replay attacks. SAS Enterprise Guide provides features to limit their exposure in SAS code and logs.

In SAS Code

The point-and-click interfaces in SAS Enterprise Guide generate SAS code. When password-protected data sets are used, the generated SAS code that will eventually be executed by a workspace server must contain any necessary data set passwords. However, by default, SAS Enterprise Guide masks passwords with tokens in generated SAS code to limit exposure. The password tokens are replaced with the actual passwords under the covers immediately before the code is submitted. The actual passwords are retrieved from the password cache or supplied by user prompt.

You can even generate and use password tokens in place of passwords in your own SAS code, which is resolved when submitted in SAS Enterprise Guide.

The format of the password token for password-protected data sets is as follows:

```
/*{credentials type="dataset" name="<dataSetName>_<serverName>_<libraryName>"
read|write|alter}*/
```

Here is some example code that contains a password token in place of an actual password:

```
PROC SQL;
    CREATE VIEW WORK.SORTTempTableSorted AS
          SELECT T.x
    FROM WORK.MYDATA(PW=/*{credentials type="dataset" name="MYDATA_SASApp_SASWork"
write}*/ ) as T
;
QUIT;
```

There might be scenarios in which you want generated code to contain actual passwords instead of tokens (for example, to be able to run the code outside of SAS Enterprise Guide). SAS Enterprise Guide provides a **Do not put plain text passwords in generated code** option (checked by default; see Figure 16), which can be unchecked so the actual passwords can be inserted instead of password tokens.

In SAS Logs

SAS servers mask password values in password fields with Xs in the SAS log. For example:

```
26         proc contents data=mydata(pw=XXXXXX);
27         run;
```

Output 1. SAS Log with Standard Passwords Masked

However, SAS servers do not recognize SAS Enterprise Guide password tokens, so if a password token is used outside of a standard password context, the SAS server sees the password as plain text and doesn't know to mask it. Because SAS Enterprise Guide knows the password tokens resolve to passwords, the application attempts to mask the cases that the server misses.

For example, submitting the following SAS code in SAS Enterprise Guide would result in the SAS log output by default:

```
%put "/*{credentials type="dataset" name="MYDATA_SASApp_SASWork" write}*/";
```

```
23          %put "                XXXXXX             ";
"               XXXXXX               "
```

Output 2. SAS Log with Unrecognized Passwords Masked

There is a **Mask credentials in SAS log when possible** option (checked by default; see Figure 16), which can be unchecked to prevent SAS Enterprise Guide from attempting to mask these fringe cases.

Continuing the example above, the log output with the option off would be as follows:

```
29          %put "mypass";
"mypass"
```

Output 3. SAS Log with Unrecognized Passwords Not Masked

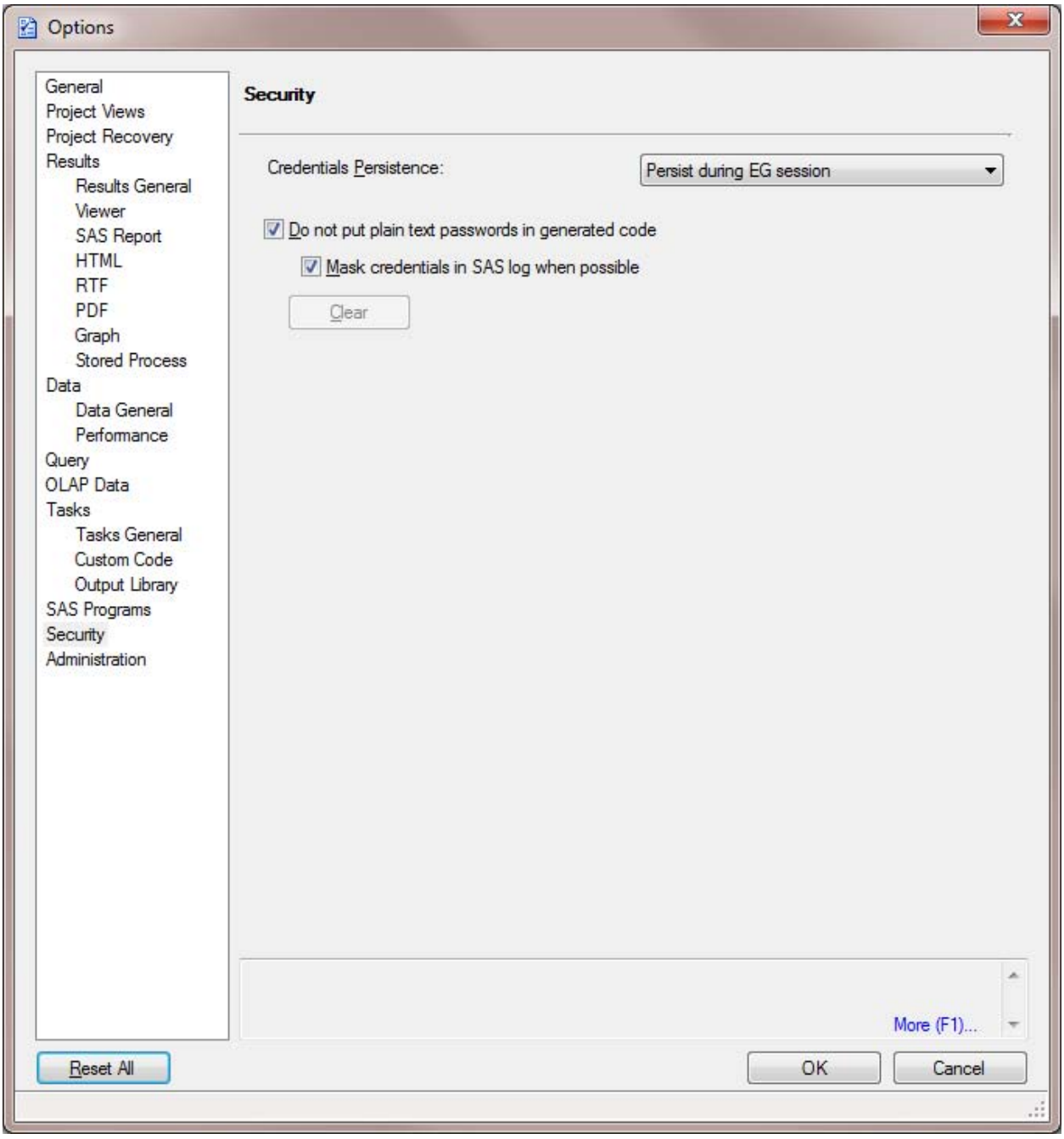

Figure 16. SAS Enterprise Guide Security Options

PROJECT SCHEDULING

SAS Enterprise Guide allows projects to be scheduled and run non-interactively on the client, which has some security requirements and considerations.

Requires Elevated Permissions

SAS Enterprise Guide accomplishes scheduling by creating a VBScript file, which runs the project via the SAS Enterprise Guide automation model. So, for scheduling to work, you must have appropriate permissions for running VBScript files on your machine. Scheduling is managed by the Windows Task Scheduler, which simply executes the VBScript file on the desired schedule or trigger. Therefore, to schedule SAS Enterprise Guide projects, you must also have the appropriate Windows permissions for creating and modifying Windows Scheduled Tasks. By default, the

ability to create and modify Windows Scheduled Tasks is typically reserved for administrators and a few special groups. These permissions are not typically available to limited user accounts, so if your users do not have elevated rights and you want them to be able to schedule projects, you must grant them the rights to create and modify Windows Scheduled Tasks.

Supplying Necessary Credentials

When a SAS Enterprise Guide project is run, certain resources such as workspace servers might require credentials. When running a project interactively from within SAS Enterprise Guide, the user is prompted for credentials when necessary. However, when running a project via automation (as is done when a scheduled project is run), the intent is often for it to be executed non-interactively and without displaying prompts. SAS Enterprise Guide provides a mechanism for making credentials available to projects that are run via automation. To accomplish this, you can specify the credentials needed in the file **%programdata%\SAS\SharedSettings\5.1\credentials.xml** in the following example XML format:

```
<xml FileVersion="5.1">
<credentials>
<server name="<server name>" userid="<userid>" password="<password>"/>
<server name="SASApp" userid="sasdemo" password="pw"/>
<server name="UNIX - Sunbeam" userid="qadcomm" password="{sas002}c2fbYD9uMQ==">/>
</credentials>
</xml>
```

Figure 17. XML Format of Credentials.xml File

Note: To avoid putting plain-text passwords in the credentials.xml file, use the PWENCODE procedure to encrypt them, similar to what is shown here:

```
proc pwencode in='my-password';
run;
```

When a project run via automation fails because of missing credentials, the file **%programdata%\SAS\SharedSettings\5.1\credentials.failed.xml** is created and contains the resources for which credentials were needed but not available. Therefore, an easy way to create the credentials.xml file is to rename the credentials.failed.xml to credentials.xml and supply the user ID and password for the resources listed in the XML.

Note: The credentials.xml and credentials.failed.xml files do not apply to library credentials, data set passwords, or project passwords. They only apply to workspace and metadata server credentials.

WHAT IS *NOT* SECURE

The only thing worse than a lack of security is a false sense of security. There are a couple of areas in SAS Enterprise Guide that might look or feel like security, but should not be relied upon for security. It is important to be aware of their limitations.

METADATA DATA ACCESS PERMISSIONS

As I noted in a previous paper (Smith 2012), setting data access permissions in SAS metadata can give you a false sense of data security. One of the most important things to understand related to data access and SAS Enterprise Guide is that SAS Enterprise Guide does not force users to work in a metadata-aware context. By providing the ability to directly submit code, a user can submit a LIBNAME statement (or use the Assign Project Library wizard, or any of a number of other ways) to get direct access to the physical data, bypassing any metadata layer permissions that might have been applied to defined libraries and registered data sources.

For example, suppose you define a library and restrict permissions in metadata to certain data sources for a particular user. When the user attempts to access data through that library, any data access restrictions in metadata are enforced. However, if the user knows where the physical data lives, the user can assign another library to access the data directly. You can easily find the physical location of the restricted library by assigning it in SAS Enterprise Guide and then submitting code similar to the following:

```
proc sql;
select * from sashelp.vlibnam;
quit;
```

Library Name	Engine Name	Pathname	Library Concatenation Level	Default File Format	Read-only?	Sequential?	System Information Description	System Information Name	System Information Value	Temp Access?
SAMPMLEA	META	e:\Sample Data	0	7	no	no	Host dependent information	Filename	e:\Sample Data	no

Output 4. Query Results Showing Physical Path of Library

Notice in Output 4, the query results show the physical path used by the library. Once the user knows the physical path, the user can simply submit a LIBNAME statement and access the data directly through the Base engine library:

```
15         libname mylib 'e:\sample data';
NOTE: Libname MYLIB refers to the same physical library as SAMPMLEA.
NOTE: Libref MYLIB was successfully assigned as follows:
      Engine:        V9
      Physical Name: e:\Sample Data
```

Or instead of accessing through another library, if the user knows the physical location of the data and has Read permission on the host, the user can read the file as a binary data stream and write it back out to another location, such as in the following SAS code example:

```
filename in "c:\tmp\secret.sas7bdat";
filename out "%sysfunc(getoption(work))\notsecret.sas7bdat";
data _null_;
  length fin 8 fout 8;
  r = '20'x;
  fin = fopen('in','I',1,'B');
  fout = fopen('out','O',1,'B');
  do while(fread(fin)= 0);
    rc = fget(fin,r,1);
    rc = fput(fout, r);
    rc =fwrite(fout);
  end;
  rc = fclose(fin);
  rc = fclose(fout);
run;
```

As long as users have the necessary host operating system permissions, they will be able to directly access and read a SAS table. So, the effect of this is that any real data security must be put in place at the physical host operating system or database layer. At least that *was* the case, until metadata-bound libraries were introduced in the second maintenance release of SAS 9.3.

Solution: Metadata-Bound Libraries

Metadata-bound libraries enable administrators to always enforce metadata-layer permission requirements before providing access to SAS data. Therefore, they are a good choice for providing seamless, secure access to SAS data. They offer more robust protection than other metadata-based data access control approaches.

A metadata-bound library is a physical library that is tied to a corresponding metadata object. Physical tables within a metadata-bound library have header information that points to a secured table object in metadata. The pointer creates a security binding between the physical table and the metadata object that ensures universal enforcement by SAS of metadata-layer permissions for the physical table.

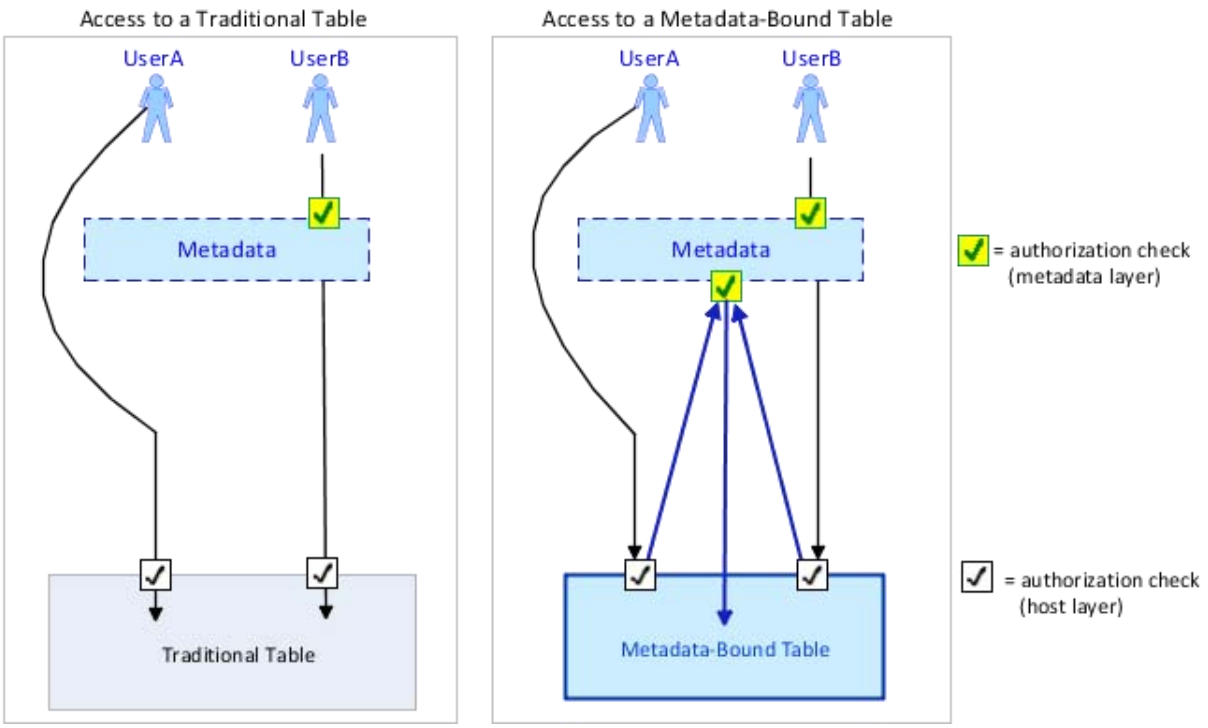

Figure 18. Authorization Checks: Traditional Table versus Metadata-Bound Table[4]

If you attempt to bypass the metadata layer as in the traditional table example in **Error! Reference source not found.**, you can discover the physical path of a metadata-bound library and submit a LIBNAME statement to assign an additional library to the location. However, metadata-layer permissions are still enforced when you try to access the metadata-bound data through this newly assigned library. The same applies if you perform a binary copy of the data file. In both cases, access is still secure because enforcement originates from the physical table.

If you are concerned about security of your physical SAS data (especially if you are using SAS data set passwords), you should seriously consider using metadata-bound libraries once all of your SAS systems have been upgraded to the second maintenance release of 9.3.

For more information, see *SAS® 9.3 Guide to Metadata-Bound Libraries*.

ROLE CAPABILITIES

As an administrator, you have the ability to control access to certain features in SAS Enterprise Guide through the use of roles. Roles are defined in metadata and configured using the User Manager plug-in in SAS Management Console. A role contains a collection of capabilities that define which specific features of the product are available to the members of that role. Roles enable you to customize the application interface for different individuals or groups, based on their responsibilities.

[4] SAS Institute Inc., *SAS 9.3 Guide to Metadata-Bound Libraries.* (Cary, NC: SAS Institute Inc., 2012), 3.

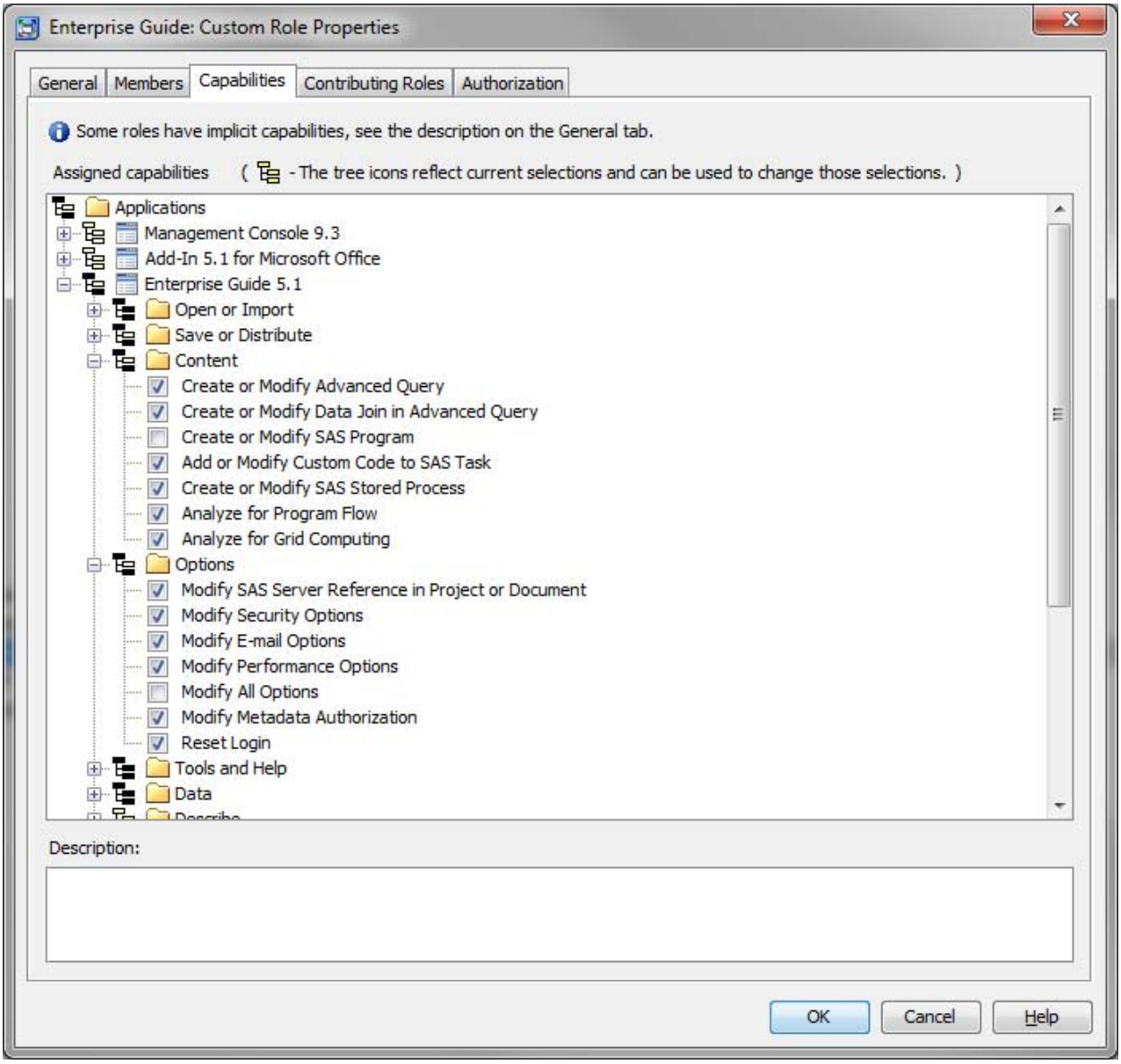

Figure 19. Enabling or Disabling Role Capabilities in SAS Management Console

Notice in Figure 20, when you log on to SAS Enterprise Guide as a user who is a member of a role that has at least one capability disabled, the Functions: Restricted link appears on the status bar next to the active connection link. Clicking the **Functions** link displays the capabilities that you do and do not have access to.

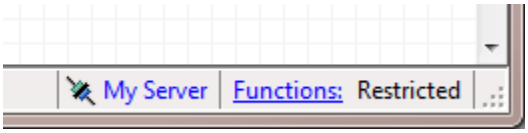

Figure 20. Restricted Functions Link on Status Bar in SAS Enterprise Guide

It is very important to note that disabling capabilities is not a substitute for security! Rather, roles simply provide a way to customize the application interface per user responsibilities.

For example, if you disable the Modify All Options capability, a user will not be able to change any of the options in the Options dialog box in SAS Enterprise Guide. However, this will not prevent the user from directly editing the EGOptions.xml file to make changes.

Likewise, if you disabled the Create or Modify SAS Program capability, SAS programs in an existing SAS Enterprise Guide project would be opened Read-Only. However, a user could easily edit the program in an external text editor, even if the program is embedded in the project.

When security is a requirement, it should be done on the server or host layer. Do not rely on role capabilities, as they are intended for feature customization, not security.

CUSTOM TASKS

Make sure you only use custom tasks from trusted sources. Custom tasks are a powerful feature that allows you to extend the functionality of SAS Enterprise Guide. However, custom tasks run with sufficient access to resources to cause harm if created with malicious intent or carelessness.

CONCLUSION

Almost all of the enterprise-level security features of SAS Enterprise Guide are inherited from the SAS Intelligence Platform. Security from the platform perspective should often be the primary focus. However, it is important to have an understanding of the overall environment, the core security features, and how SAS Enterprise Guide participates. It is also worth having awareness of the security features and considerations such as password-protecting project files, credentials caching, password masking, and project scheduling that are specific to SAS Enterprise Guide. Finally, recognize features that are not intended to offer robust security in SAS Enterprise Guide, such as normal metadata data access controls (though metadata-bound libraries provide an excellent solution), role capabilities, and custom tasks. Be safe!

REFERENCES

- SAS Institute Inc. 2011. *SAS 9.3 Intelligence Platform: Security Administration Guide*. Cary, NC: SAS Institute Inc.
  Available at http://support.sas.com/documentation/cdl/en/bisecag/63082/PDF/default/bisecag.pdf.
- SAS Institute Inc. 2012. *SAS 9.3 Intelligence Platform: Data Administration Guide, Second Edition*. Cary, NC: SAS Institute Inc.
  Available at http://support.sas.com/documentation/cdl/en/bidsag/62767/PDF/default/bidsag.pdf.
- SAS Institute Inc. 2011. *SAS 9.3 Intelligence Platform: Desktop Application Administration Guide*. Cary, NC: SAS Institute Inc.
  Available at http://support.sas.com/documentation/cdl/en/bidaag/63126/PDF/default/bidaag.pdf.
- SAS Institute Inc. 2011. *SAS 9.3 Integration Technologies: Windows Client Developer's Guide*. Cary, NC: SAS Institute Inc.
  Available at http://support.sas.com/documentation/cdl/en/itechwcdg/62763/PDF/default/itechwcdg.pdf.
- SAS Institute Inc. 2012. *SAS 9.3 Guide to Metadata-Bound Libraries.* Cary, NC: SAS Institute Inc.
  Available at http://support.sas.com/documentation/cdl/en/seclibag/65159/PDF/default/seclibag.pdf.
- SAS Institute Inc. 2011. "What SAS Administrators Should Know about Libraries, Metadata, and SAS Enterprise Guide". Cary, NC: SAS Institute Inc.
  Available at http://support.sas.com/documentation/onlinedoc/guide/EG43MetaLibraries.pdf.
- Smith, Casey. 2012. "Best Practices for Administering SAS Enterprise Guide." *Proceedings of the SAS Global Forum 2012 Conference*. Cary, NC. SAS Institute Inc.
  Available at http://support.sas.com/resources/papers/proceedings12/297-2012.pdf.

CONTACT INFORMATION

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

Casey Smith
SAS Campus Drive
SAS Institute Inc.
Casey.Smith@sas.com