CHAPTER

1

Understanding Authentication

Scope of This Document

This document explains the security model for the SAS Intelligence Platform and provides instructions for performing security-related administrative tasks. The emphasis is on suite-wide aspects of the security functionality that SAS provides. Some interactions with other security layers (such as operating system permissions, WebDAV access controls, and third-party database security) are noted. Detailed information about features and requirements that are unique to a particular application is provided in the administrative documentation for that application.

This document assumes that you are familiar with the concepts and terminology that are introduced in *SAS Intelligence Platform: Overview*. For a list of all of the documents that SAS publishes to support administration of the SAS Intelligence Platform, see **support.sas.com/administration**.

Accessibility Features in the SAS Intelligence Platform Products

For information about accessibility for any of the products mentioned in this book, see the documentation for that product. If you have questions or concerns about the accessibility of SAS products, send e-mail to accessibility@sas.com.

Authentication Overview

Introduction to Authentication

Authentication is an identity verification process that attempts to determine whether users (or other entities) are who they say they are. Authentication is a prerequisite for authorization, because a user's identity is the basis for authorization decisions about which actions the user is permitted to perform with which resources.

In the SAS Intelligence Platform, a user's identity is verified first when the user logs on to an application and again as the user requests access to other systems. For example, when a user logs on to SAS Data Integration Studio, the user authenticates to the SAS Metadata Server. When the user makes a request from SAS Data Integration Studio to run a job against an Oracle table, the user must authenticate to the SAS Workspace Server that processes the request and to the Oracle server that manages the table.

How Identities Are Verified

In most cases, SAS servers rely on their host operating systems to verify identities. This process is called host authentication. For example, before allowing a user to run a stored process, a stored process server asks its host computer to authenticate the user. The host computer compares a provided user ID and password to a list of valid accounts in the operating system (or in a back-end authentication database that the operating

system is using). If the provided user ID and password correspond to a valid account, the authentication is successful.

Note: As an alternative to relying on the host operating system, the metadata server and the OLAP server can make direct use of Lightweight Directory Access Protocol (LDAP) or Microsoft Active Directory to verify identities. However, the preferred way to use an alternative authentication provider is as a back-end user store behind host authentication, because direct use of LDAP and Active Directory Direct can significantly increase the need to store user IDs and passwords in the metadata repository. △

In some cases, SAS servers trust verification that has been performed by other components. The SAS Intelligence Platform supports the following trust relationships:

- The metadata server trusts the identity verification that the SAS OLAP Server performs.
- By default, the metadata server trusts the identity verification that a connecting SAS process performs.
- If SAS Web applications are configured to use Web server authentication, the metadata server trusts the identity verification that a Web server performs.

Related Topics:

"Using LDAP or Active Directory" on page 82

"Using Web Authentication" on page 82

"The Authentication Process" on page 11

Single Sign-On

Single sign-on enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. The SAS Intelligence Platform provides these single sign-on features:

- Most applications can cache the credentials that a user submits to log on.
- All applications can retrieve credentials that have been stored in the metadata repository.
- Web applications can share user and session contexts.

Related Topics:

"Reuse of Credentials That Are Cached from an Interactive Log On" on page 17

"Retrieval of Credentials from the Metadata Repository" on page 18

"Shared User Context (Among Web Applications)" on page 20

Identity Management

In addition to managing user accounts in external systems, administrators must create and maintain some user information in the metadata repository. You can minimize the amount of identity information that you need to replicate in the metadata by choosing your authentication providers carefully and making appropriate use of shared accounts.

The SAS Intelligence Platform provides the following tools for management of identity information in the metadata:

- Administrators can use SAS Management Console to define and manage metadata identity information.
- Administrators can use batch processes to extract identity information from sources such as LDAP or UNIX /etc/passwd files and create corresponding identity

information in the metadata repository. Batch processes can also be used to periodically update the identity information. Batch processes cannot be used to manage passwords.

- Users can use the SAS Personal Login Manager desktop application to manage their own account information.

Related Topics:

"Choosing Authentication Providers" on page 52

"How to Use Shared Accounts" on page 80

Appendix 2, "Bulk-Load Processes for Identity Management," on page 183

Authentication Terminology

Introduction to Authentication Terminology

The following terms are important to understanding how authentication works in the SAS Intelligence Platform:

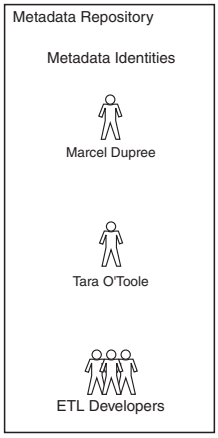
authentication provider	a technology that servers or applications can use to verify that users are who they say they are. Operating systems, LDAP, Active Directory, and third-party database system authentication mechanisms are examples of authentication providers.
metadata identity	a metadata object that represents an individual user or a group of users on a SAS Metadata Server. Each metadata identity must be unique within a metadata server.
login	a metadata object that is owned by a metadata identity. Each login contains the user ID (and, sometimes, the password) for an account that has been established with an authentication provider. Each login corresponds to a particular user account with a particular authentication provider. For example, if you have a UNIX account with a user ID of **`tara`** and a password of **`tara1234`**, then you can store that account information in the metadata as a login.
authentication domain	a metadata object that links logins to the servers for which the logins are valid. Each authentication domain should be associated with one or more servers and with the logins that provide access to those servers. All of the computing resources within an authentication domain use the same authentication provider. You can choose to use the same groupings and names for your authentication domains as you do for your host domains or network domains, but you are not required to do so.

The following topics explain the role of metadata identities, logins, and authentication domains in the authentication model.

How Metadata Identities Are Used

Metadata identities are used as the basis for making authorization decisions and responding to requests for credentials. The following figure depicts several metadata identities within a SAS Metadata Repository.

Figure 1.1 Metadata Identities



The metadata server discovers a user's metadata identity by performing these steps:

1 The metadata server searches the metadata repository for a login that contains a user ID that matches the user ID with which the user was authenticated.

   In this process, the metadata server attempts to match the fully qualified user ID. For example, if a user logs on to a server that is using Windows host authentication, and the user's Windows user ID is **`marcel`** in a Windows domain named **`WinNT`**, then the metadata server searches the repository for a login that includes the user ID **`WinNT\marcel`**. For this reason, it is important to carefully specify the user ID in each login that you create.

   - When you create a login for a network Windows user account, specify the user ID in the form *Windows-domain-name**userID* or in the form *userID*@*Windows-domain-name*.
   - When you create a login for a local Windows user account, specify the user ID in the form *machine-name**userID* or in the form *userID*@*machine-name*.
   - When you create a login for an LDAP user account, specify the user ID in the form *userID*@*authentication-provider*.
   - When you create a login for a Microsoft Active Directory user account, specify the user ID in the form *Windows-domain-name**userID* or in the form *userID*@*Windows-domain-name*.
   - When you create a login for a UNIX or z/OS operating system user account, specify the user ID in the form *userID*.

2 The metadata server determines which metadata identity owns the login that contains the matching user ID. For example, if the metadata server finds that a login that contains the user ID **`WinNT\marcel`** is stored with the user definition for Marcel Dupree, then the metadata server knows that Marcel Dupree is the metadata identity of the user who logged on.

   If there is no matching user ID in the repository, then the user's access corresponds to the access of the PUBLIC group, with these exceptions:

   - If the user is using SAS Web Report Studio (with the surrogate user configuration) or the SAS Information Delivery Portal, then the user's access corresponds to the access that has been defined for the surrogate user. By default this is the SAS Guest User.
   - If the user has special status as an *unrestricted user* or an *administrative user* of the metadata server, then the user can perform certain tasks even if he or she does not have an individual metadata identity.

How Logins Are Used

Logins are primarily used in two ways:

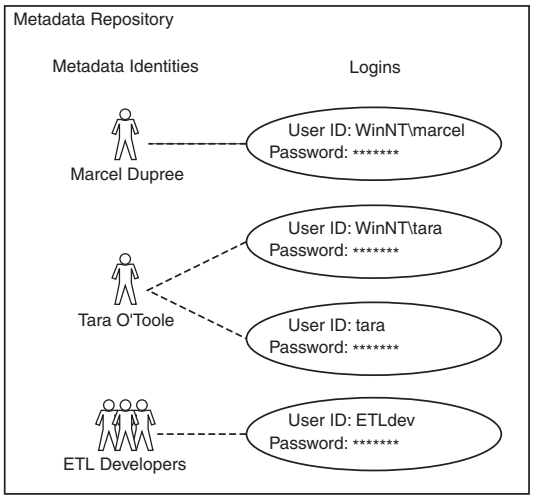
- The metadata server uses logins to determine a user's metadata identity. When a login is used to determine a user's metadata identity, the login is functioning as an *inbound login* (the login is inbound to the metadata server). As explained in the preceding topic, the metadata server does not examine passwords or consider authentication domains in this process.
- Applications use logins to acquire credentials as part of a single sign-on approach to authentication. An application can retrieve a login from the metadata server and send those credentials to another system that needs to verify a user's identity. When a login is used to provide access to a server other than the metadata server, the login is functioning as an *outbound login* (the login is outbound from the metadata server to another system). An outbound login must include a user ID and password that are appropriate for the server or host to which the login provides access. An outbound login must be associated with an authentication domain.

Logins can also be used in these ways:

- The object spawner uses logins to obtain credentials for launching servers that run under designated accounts. When you configure a stored process server or a pooled workspace server, you specify an account under which the server will run. For example, during installation the stored process server is configured to run under the sassrv account. In order to launch that stored process server, the object spawner needs the credentials for the sassrv account. The object spawner obtains those credentials from a login that is owned by the SAS General Servers group. For more information, see "Accessing a Pooled SAS Workspace Server" on page 27.
- The Xythos WebFile Server uses logins to discover and set up users for access control in the WebDAV authorization layer. Xythos builds its list of users by retrieving from the metadata server all of the logins that are associated with the authentication domain of the Xythos WebFile Server. In the default configuration, that server is associated with the DefaultAuth authentication domain, so a user must have a login that is associated with the DefaultAuth authentication domain in order to be a valid user of the Xythos WebFile Server. In alternate configurations, a user must have a login for some other authentication domain in order to be a valid user of the Xythos WebFile Server. For more information, see "Accessing a Xythos WebFile Server" on page 31.

Each login is owned by only one metadata identity. Each metadata identity can own multiple logins. The following figure depicts the relationships between logins and metadata identities in a metadata repository.

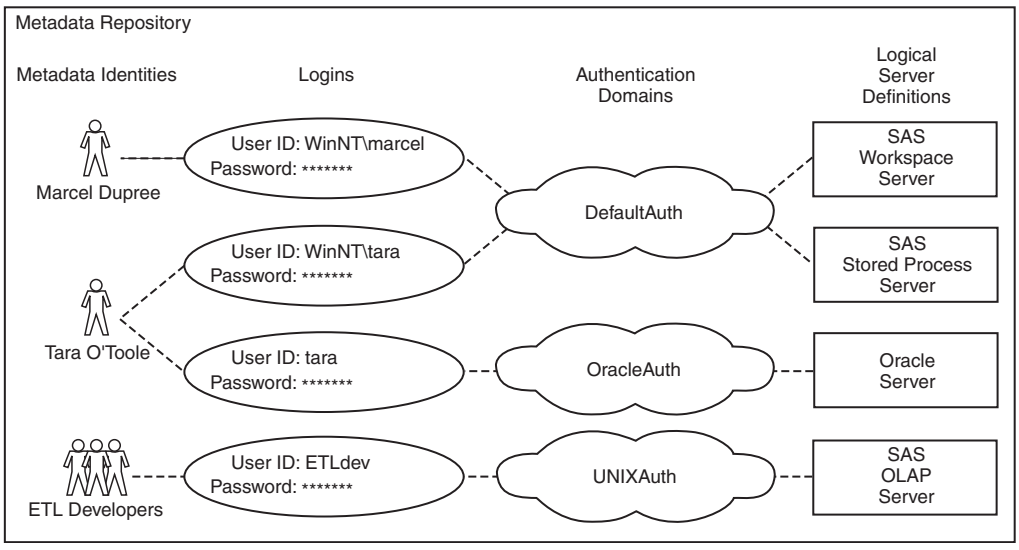
Figure 1.2 Metadata Identities and Logins



How Authentication Domains Are Used

Authentication domains are used to support single sign-on from an application to other systems. Each authentication domain corresponds to a logical grouping of servers and logins within a metadata repository. The following figure depicts the relationships between servers, authentication domains, and logins.

Figure 1.3 Metadata Identities, Logins, and Authentication Domains



When an application searches the metadata for a login that provides access to a particular server, the application uses authentication domains to determine which logins contain credentials that are appropriate for that server. For example, if Tara makes a request that requires access to the Oracle server, then the Oracle server will have to verify Tara's identity. The application that Tara is using must provide Tara's Oracle user ID and password to the Oracle server. The application will complete these steps:

1 Determine that the Oracle server definition is associated with the OracleAuth authentication domain.

2 Ask the metadata server for a login that is both associated with the OracleAuth authentication domain and owned by Tara's metadata identity (or by a group to which Tara's identity belongs).

In the preceding figure, Tara's second login meets these criteria. If this login includes Tara's password for the Oracle server, then Tara will be able to access that server. If Marcel makes a similar request, he will be denied access to the Oracle server because Marcel does not have a login for the OracleAuth authentication domain. For additional examples, see "Authentication Scenarios" on page 21.

Uniqueness Requirements for Names and User IDs

Within a SAS Metadata Server, the following uniqueness requirements apply to metadata identity names and stored credentials:

- You cannot create a user definition that has the same name as an existing user definition.
- You cannot create a group definition that has the same name as an existing group definition.
- You cannot assign the same user ID to two different metadata identities. All of the logins that include a particular user ID must be owned by the same metadata identity. This enables the metadata server to resolve each user ID to a single metadata identity.
  - This requirement is case-insensitive. For example, you cannot assign a login with a user ID of *smith* to one user and a login with a user ID of *SMITH* to another user.
  - This requirement applies to the fully qualified form of the user ID. For example, you can assign a login with a user ID of *winDEV\brown* to one user and a login with a user ID of *winPROD\brown* to another user. In this example, *winDEV* and *winPROD* are Windows domain names, which are incorporated into the fully qualified form of a user ID.
  - This requirement cannot be mitigated by associating the logins with different SAS authentication domains. For example, if one user has a login with a user ID of *smith* that is associated with a SAS authentication domain named *DefaultAuth*, you cannot give any other user a login with the user ID *smith*, even if you plan to associate the login to a different SAS authentication domain.

  Note: To enable multiple users to share an account, store the credentials for that account in a login as part of a group definition. Then add the users who will share the account as members of that group definition. △
- If you give a user two logins that contain the same user ID, the logins must be associated with different authentication domains. Within an authentication domain, each user ID must be unique. For example, if you give the person *Tara O'Toole* two logins that both have a user ID of *tara*, then you cannot associate both of those logins with the *OraAuth* authentication domain.

  Note: Like the previous requirement, this requirement is case-insensitive and is applied to the fully qualified form of the user ID. △

The Authentication Process

Overview of the Authentication Process

The authentication process can be thought of as occurring in two phases:

1. In the *initial authentication* phase, a user logs on with a SAS Intelligence Platform client or opens a metadata profile. The user ID and password that the user submits are sent to an authentication provider to verify the user's identity. After the user ID and password are verified, the metadata server determines the user's metadata identity.
2. In the *additional authentication* phase, a user makes a request that requires access to an additional system such as a workspace server, stored process server, or database server. The application that the user is using provides the user's credentials to the additional server. This enables the additional server to verify the user's identity against its authentication provider.

These phases are described in detail in the following sections.

Initial Authentication

Overview of Initial Authentication

Initial authentication is the verification of a user's identity based on information that the user provides when the user logs on with a SAS Intelligence Platform client. Initial authentication requires that the user have an account with the authentication provider that verifies the user ID and password that is submitted. The account can be any of the following:

- a local user account in the operating system of the computer on which the authenticating server is running
- a network user account that provides access to the operating system of the computer on which the authenticating server is running
- an LDAP or Active Directory user account (if the authenticating server is using one of these alternative authentication providers)
- a user account with any authentication provider that the Web application server uses (for applications that are configured to use Web authentication)

The initial authentication process varies depending on the software component that the user is using. The following table describes how each software component verifies identities.

Table 1.1 Initial Authentication

Type of Software Component	Identity Verification Process
Desktop applications Web applications that are using metadata server authentication	The metadata server's authentication provider verifies that the user ID and password that the user submits correspond to an existing account. For a depiction of this process, see "Initial Authentication on a Metadata Server" on page 12.
Web applications that are using Web authentication	The Web application server's authentication provider verifies that the user ID and password that the user submits correspond to an existing account. The Web application then uses *trusted user** functionality to enable the user to access the metadata server. The user does not need an account with the metadata server's authentication provider. For a depiction of this process, see "Initial Authentication on a Web Application Server" on page 13.
Components that connect directly to a SAS OLAP Server (such as the SAS OLAP Data Provider)	The SAS OLAP Server's authentication provider verifies that the user ID and password that the user submits correspond to an existing account. The SAS OLAP Server then uses *trusted user** functionality to enable the user to access the metadata server. The user does not need to have an account with the metadata server's authentication provider. For a depiction of this process, see "Initial Authentication on a SAS OLAP Server" on page 15.

* *Trusted user* functionality supports a multi-tier server environment in which user identities are authenticated by a server other than the metadata server.

After the user ID and password that the user submits are verified by the appropriate authentication provider, the proof-of-identity is complete. None of the user information that is stored in the metadata repository is used to prove the user's identity.

Next, the metadata server must discover the user's metadata identity for these reasons:

- In order to provide authorization decisions and credential management, the metadata server needs to know who the user is.
- Some applications have an *additional* requirement beyond proof-of-identity and will not allow users to log on unless they have a metadata identity. For example, a user must have a metadata identity in order to log on to SAS Information Map Studio or to access the SAS Information Delivery Portal beyond the public kiosk. SAS Web Report Studio can also be configured to require each user to have a metadata identity.

In order to discover your metadata identity, the metadata server examines the user IDs that are stored in the metadata repository. Passwords that are stored in the metadata repository are not examined at any point during initial authentication.

Initial Authentication on a Metadata Server

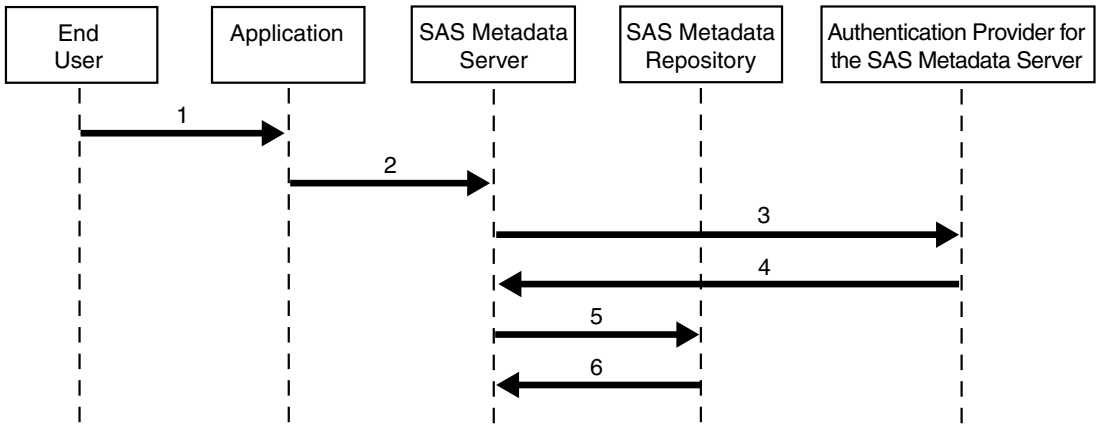
The metadata server handles initial authentication when a user logs on with the following types of applications:

- a desktop application such as SAS Management Console, SAS Data Integration Studio, SAS OLAP Cube Studio, or SAS Information Map Studio
- a Web application (such as SAS Web Report Studio or the SAS Information Delivery Portal) that is configured to authenticate users on the metadata server

The following figure depicts these activities:

- verification of credentials that a user submits to an application that authenticates users on a metadata server
- determination of the user's metadata identity

Figure 1.4 Initial Authentication on a Metadata Server



In this figure, the numbered arrows correspond to the following activities:

1. The user submits a user ID and password to a SAS application (by logging on or by opening a metadata profile).
2. The application sends the user ID and password to the metadata server.
3. The metadata server passes the user ID and password to its authentication provider for verification. For example, if the authentication provider is the host operating system, then the metadata server passes the user ID and password to the operating system of the machine on which the metadata server is running.
4. The authentication provider verifies that the user ID and password combination corresponds to an existing user account. For example, if the authentication provider is the host operating system, then the user ID and password combination must correspond to a local or network user account that has been established in the operating system. After verification, the authentication provider tells the metadata server that the user ID and password are valid and sends the user ID back to the metadata server.
5. The metadata server looks for the user ID in the logins that are stored in the metadata repository.

   Note: The metadata server attempts to match the user ID in its fully qualified form, as described in "How Metadata Identities Are Used" on page 6. △
6. The metadata server determines which metadata identity owns a login that contains the matching user ID.

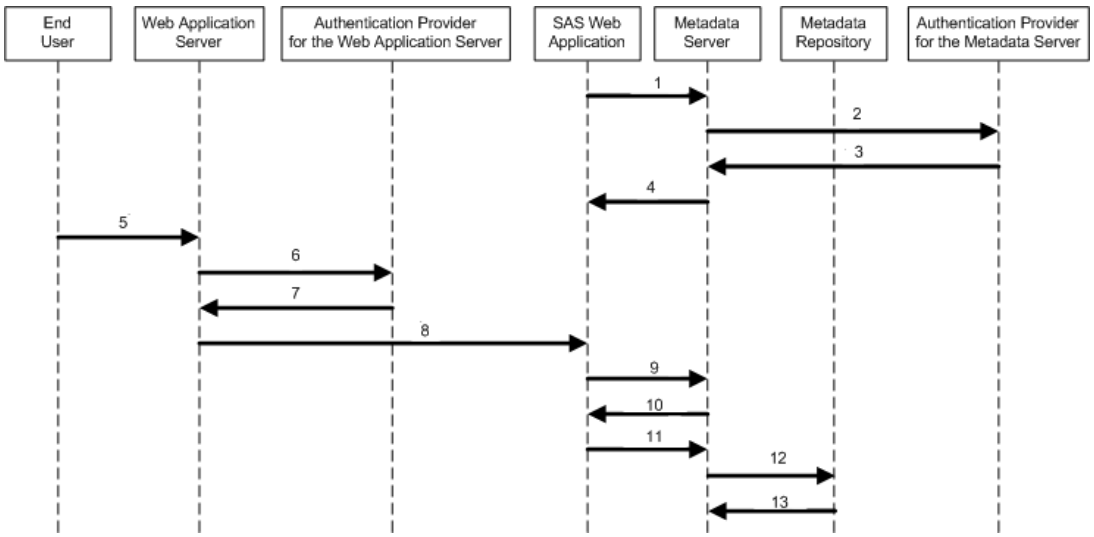
Initial Authentication on a Web Application Server

When a Web application such as SAS Web Report Studio or the SAS Information Delivery Portal is configured to use Web authentication, the authentication provider of the Web application server must verify the credentials that users submit.

The following figure depicts these activities:

- establishment of a trusted connection between a Web application and the metadata server
- verification of credentials that a user submits to a Web application that is configured to authenticate users on a Web application server
- determination of the user's metadata identity

Figure 1.5 Initial Authentication on a Web Application Server



In this figure, the numbered arrows correspond to the following activities:

1 When the SAS Web application initializes, it sends the user ID and password for the *trusted user* (sastrust) to the metadata server to request a trusted connection.

2 The metadata server passes the sastrust user ID and password to its authentication provider.

3 The metadata server's authentication provider verifies that the sastrust user ID and password correspond to an existing account.

4 The metadata server tells the Web application that the trusted connection is accepted. This connection will be used in step 9.

5 After navigating to a URL for a SAS Web application, a user submits a user ID and password in response to a prompt from the Web application server.

6 The Web application server passes the user's ID and password to its authentication provider.

7 The Web application server's authentication provider verifies that the user's ID and password correspond to an existing account.

8 The Web application server sends the authenticated user ID to the SAS Web application.

9 The SAS Web application uses the (previously established) trusted connection to the metadata server to request a one-time-use password for the user.

10 The metadata server generates a one-time-use password for the user and sends that password to the SAS Web application. The metadata server trusts that the user's credentials have already been verified.

11 The SAS Web application uses the user's ID and the generated password to establish a connection to the metadata server for the user.

12 The metadata server looks for the user ID in the logins that are stored in the metadata repository.

Note: The metadata server attempts to match the user ID in its fully qualified form, as described in "How Metadata Identities Are Used" on page 6. △

13 The metadata server determines which metadata identity owns the login that contains the matching user ID.

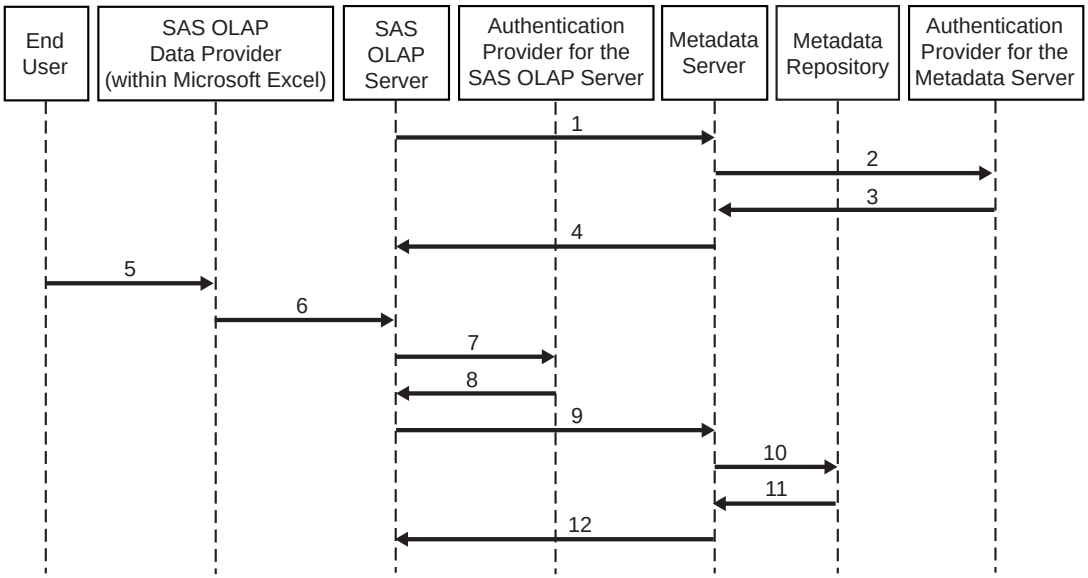
Initial Authentication on a SAS OLAP Server

The SAS OLAP Server handles initial authentication when a user accesses a component that connects directly to a SAS OLAP Server. For example, when a user accesses SAS OLAP data from Microsoft Excel, the SAS OLAP Data Provider passes the user's credentials to the SAS OLAP Server for initial authentication.

The following figure depicts these activities:

- establishment of a trusted connection between a SAS OLAP Server and the metadata server
- verification of credentials that a user submits to a component that connects directly to a SAS OLAP Server
- determination of the user's metadata identity

Figure 1.6 Initial Authentication on a SAS OLAP Server



In this figure, the numbered arrows correspond to the following activities:

1 When the SAS OLAP Server initializes, it sends the user ID and password for the *trusted user* (sastrust) to the metadata server to request a trusted connection.

2 The metadata server passes the sastrust user ID and password to its authentication provider.

3 The metadata server's authentication provider verifies that the sastrust user ID and password correspond to an existing account.

4 The metadata server tells the SAS OLAP Server that the trusted connection is accepted. This connection will be used in step 9.

5 After requesting access to SAS OLAP data from within Microsoft Excel, a user submits a user ID and password in response to a prompt from the SAS OLAP Data Provider.

6 The SAS OLAP Data Provider passes the user's ID and password to the SAS OLAP Server.

7 The SAS OLAP Server passes the user's ID and password to its authentication provider for verification.

8 The SAS OLAP Server's authentication provider verifies that the user's ID and password correspond to an existing account.

9 The SAS OLAP Server uses the (previously established) trusted user connection to request a credential handle for the user. In this process, the user's authenticated ID is passed to the metadata server. The metadata server trusts that the SAS OLAP Server has already verified the user's credentials.

Note: The SAS OLAP Server uses credential handles to specify which user is making a particular request over the trusted connection. △

10 The metadata server looks for the user ID in the logins that are stored in the metadata repository.

Note: The metadata server attempts to match the user ID in its fully qualified form, as described in "How Metadata Identities Are Used" on page 6. △

11 The metadata server determines which metadata identity owns the login that contains the matching user ID.

12 The metadata server sends a credential handle for the user to the SAS OLAP Server.

Trusted Peer Session Connections

During installation, the SAS Configuration Wizard incorporates the **`trustsaspeer`** option in the start command for the metadata server. This causes the metadata server to accept trusted peer connections from SAS processes that request access using a certain proprietary protocol. In a trusted peer connection, the metadata server trusts the authentication that another SAS process (a SAS session, workspace server, or stored process server) has already performed. Trusted peer session connections are used by applications that need to generate code or run batch jobs without explicitly providing credentials to the metadata server.

Additional Authentication

Overview of Additional Authentication

Additional authentication is the use of credentials by other systems after initial authentication. For example, when a user accesses an application such as SAS Web Report Studio to view a report that contains live data, the application might have to provide the user's credentials to a SAS Stored Process Server to enable that server to verify the user's identity.

The SAS Intelligence Platform uses a single sign-on model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. The following sections describe the ways that applications can obtain credentials for the purpose of providing those credentials to the servers that need to verify users' identity.

Reuse of Credentials That Are Cached from an Interactive Log On

Most SAS applications cache the credentials that users provide when they log on. The cached credentials can be reused for authentication to other servers. For example, if a user logs on to SAS Information Delivery Portal and is initially authenticated on a metadata server that is using UNIX host authentication, then the user's cached UNIX credentials can be reused for authentication to a stored process server that is running on UNIX. Of course, the user's cached credentials will be valid for only one authentication provider. For example, the user's cached UNIX credentials cannot be reused for authentication to a stored process server that is running on Windows, or for authentication to a third-party database server that is using a proprietary authentication mechanism. Cached credentials are valid only for servers within only one authentication domain. In most cases, applications use cached credentials for servers that are in the DefaultAuth authentication domain. The following table provides details about how applications determine when to use cached credentials.

Table 1.2 Use of Cached Credentials

Application	Authentication Domain for Which Cached Credentials are Used
SAS Add-In for Microsoft Office	The authentication domain that is specified in the AuthenticationDomain Name= field in the **`CSIDL_APPDATA\SAS\Metadata Server\oms_serverinfo.xml`** file.
SAS Data Integration Studio	The authentication domain that the user specified in the metadata profile.[2]
SAS Enterprise Guide	Any authentication domain. When a user requests access to a workspace server, SAS Enterprise Guide 3.1 attempts to reuse the credentials that the user provided when the user logged on.
SAS Enterprise Miner	The authentication domain that is specified in the default_auth_domain= field in the **`SASAPCore\conf\server.config`** file.[1]
SAS Information Delivery Portal	The authentication domain that is specified in the $SERVICES_OMI_DOMAIN$= field in the **`PortalConfigure\install.properties`** file.[1]
SAS Information Map Studio	The authentication domain that the user specified in the metadata profile.
SAS Marketing Automation	The authentication domain that is specified in the **`login.config`** file (or its equivalent) for the Web container that SAS Marketing Automation is using.
SAS OLAP Cube Studio	The authentication domain that the user specified in the metadata profile.[2]

Application	Authentication Domain for Which Cached Credentials are Used
SAS Web Report Studio	The authentication domain that is specified in the $LOGON_DOMAIN$= field in the **wrs.config** file.[1]
SAS Web OLAP Viewer	The authentication domain that is specified in the $SERVICES_OMI_DOMAIN$= field in the **\SASWebOlapViewerforJava\3.1\Configure\install.properties** file.[1]

1 The authentication domain that is specified in the application properties file must also match the authentication domain that is specified in the associated **login.config** file (or its equivalent). If the two values do not match, the authentication fails.

2 If you do not specify an authentication domain in your metadata profile, then this application does not use your cached credentials.

Retrieval of Credentials from the Metadata Repository

In most deployments, there are some authentication events for which cached credentials cannot be used. For example, if a user's cached credentials are for a UNIX system, those credentials will not enable the user to access a stored process server that is running on Windows. For these authentication events, the user (or a group to which the user belongs) must have a login in the metadata that contains credentials that are appropriate for the target server.

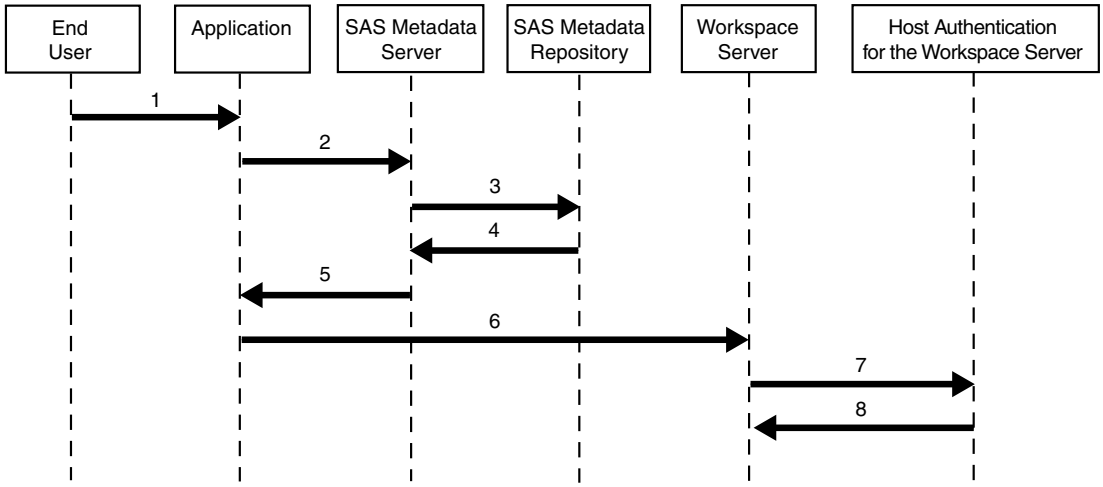
Note: In most deployments, you will need to store some passwords in the metadata. However, you can minimize the need for this by giving careful consideration to your selection of authentication providers and using shared accounts where appropriate. For details, see "Choosing Authentication Providers" on page 52 and "How to Use Shared Accounts" on page 80. △

When an application needs to provide a user's credentials to another server (and cached credentials cannot be used), the application asks the metadata server to search the metadata repository for a login that contains credentials that can be used to access the target server. The login must be owned by the user's metadata identity (or by a user group to which the user's metadata identity belongs). If the application finds an appropriate login, the application passes that user ID and password to the server that the user needs to access. The target server then uses those credentials to verify the user's identity against its authentication provider.

The following figure depicts this process. The example assumes these conditions:

- The user is represented in the repository by a metadata identity that owns a login that contains credentials for accessing the SAS Workspace Server.
- The user has already completed initial authentication.
- The target server is a workspace server that is not configured for pooling.

Figure 1.7 Additional Authentication



Note: In order to provide a generalized depiction that is applicable to a wide variety of target servers, the figure omits the object spawner (which is used to launch workspace servers and stored process servers). For the purposes of completeness, implementation details relating to the object spawner are noted in the following process description. Additional examples and depictions of server-specific aspects of this process are provided in "A Closer Look: Accessing SAS Servers" on page 25 and "A Closer Look: Accessing Third-Party Servers" on page 30. △

In the figure, the numbered arrows correspond to the following activities:

1 The user makes a request that requires access to a SAS Workspace Server.

2 The application recognizes that the request requires access to a workspace server, so the application goes to the metadata server to get credentials that will give the user access to a workspace server.

3 The metadata server looks for the requested credentials in the metadata repository. The credentials must meet both of these criteria:

- The credentials are stored in a login that is owned by the requesting user's metadata identity (or by a group to which that identity belongs).
- The credentials are stored in a login that is associated with the authentication domain in which the workspace server is registered.

4 The metadata server locates the appropriate credentials in the metadata repository and retrieves those credentials from the metadata repository.

5 The metadata server sends the credentials to the requesting application.

6 The application sends the credentials to the target server.

Note: Because the target server is an unpooled workspace server, the application actually sends the credentials to the object spawner that will launch the workspace server (rather than to the workspace server itself). △

7 The target server passes the credentials to its authentication provider for verification.

Note: In this example, it is actually the object spawner (rather than the workspace server) that passes the credentials to its authentication provider for verification. The authentication provider for a workspace server is always the host operating system. △

8 The authentication provider tells the target server that the credentials are valid. The target server then accepts the connection.

Note: In this example, the host operating system tells the object spawner (rather than the workspace server) that the credentials are valid. The object spawner then launches a workspace server for the requesting user. △

Shared User Context (Among Web Applications)

The previous topics describe single sign-on from an application to different servers. SAS Web applications can also support single sign-on from one application to another. When Web applications share user context and session information, users can launch one Web application from within another Web application without having to log on to the second application. For example, because the SAS Web Report Viewer and the SAS Information Delivery Portal use the same remotely deployed session service, a user can access the SAS Web Report Viewer from the portal application without logging in again. In this example, the SAS Web Report Viewer shares the session and user context that was initiated when the user logged on to the SAS Information Delivery Portal.

Interactive Prompting for SAS Server Credentials

If credentials cannot be otherwise obtained, some SAS applications prompt users for their credentials for accessing SAS servers. For example, SAS Data Integration Studio prompts users for their user ID and password for a workspace server if those credentials are not stored in the metadata repository. Interactive prompting is available only for accessing SAS servers. For authentication to a third-party database server, credentials must be stored in the metadata repository, as described in "Example: Managing Authentication to a Database Server" on page 81.

Summary: Credential Management Features by Client

The following table shows which credential management features work with each SAS client.

Table 1.3 Accessing Additional Resources after Initial Authentication

Client	Supported Credential Management Features			
	Retrieval of Stored Credentials	**Reuse of Cached Credentials**	**Interactive Prompting**	**Sharing User Context**
SAS Add-In for Microsoft Office	x	x	x	
SAS Data Integration Studio	x	x	x	
SAS Enterprise Guide	x	x[1]	x	
SAS Enterprise Miner	x	x	x	
SAS Information Delivery Portal	x	x		x
SAS Information Map Studio	x	x		

Client	Supported Credential Management Features			
	Retrieval of Stored Credentials	**Reuse of Cached Credentials**	**Interactive Prompting**	**Sharing User Context**
SAS OLAP Cube Studio	x	x	x	
SAS Web Report Studio	x	x		x
SAS Web OLAP Viewer	x	x		x
SAS Stored Process Application	x	x		x

1 SAS Enterprise Guide 3.1 attempts to reuse cached credentials to provide access to workspace servers (but not to provide access to stored process servers).

Authentication Scenarios

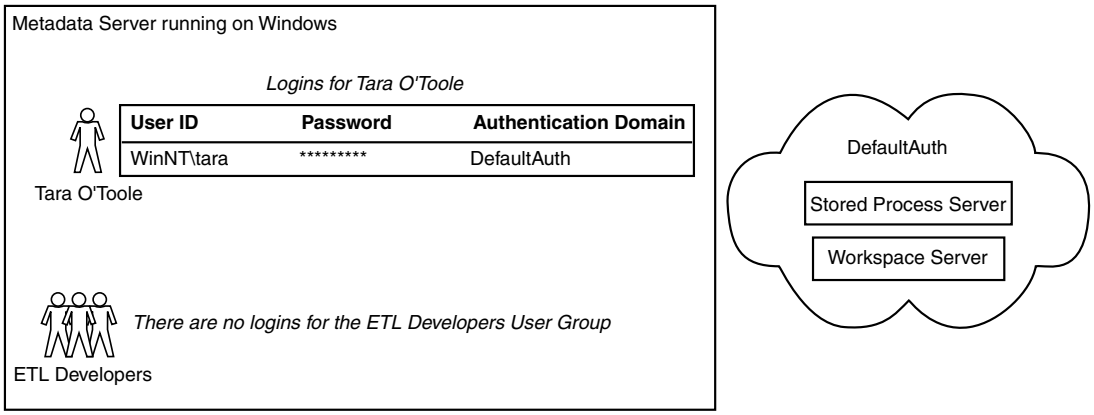
Introduction to Authentication Scenarios

This section explains the relationships between servers, authentication domains, and logins in a variety of deployment scenarios. In each scenario, the logins that are stored in the metadata for an individual user (Tara O'Toole) and a particular user group (ETL Developers) are identified.

Single Platform Environments

In a homogeneous environment, you might need only one authentication domain. The following figure depicts a deployment in which all of the logical servers and all of the logins for all metadata identities are associated with an authentication domain that is named DefaultAuth.

Figure 1.8 Homogeneous Environment, One Authentication Domain



In this figure, the metadata identity that represents Tara owns only one login, which functions as both an inbound and an outbound login. Because the servers are running under Windows, the user ID in the login is fully qualified with the name of the Windows domain (WinNT). Because Tara's password is stored in the login, Tara will be able to access the workspace server and stored process server without being prompted for her credentials.

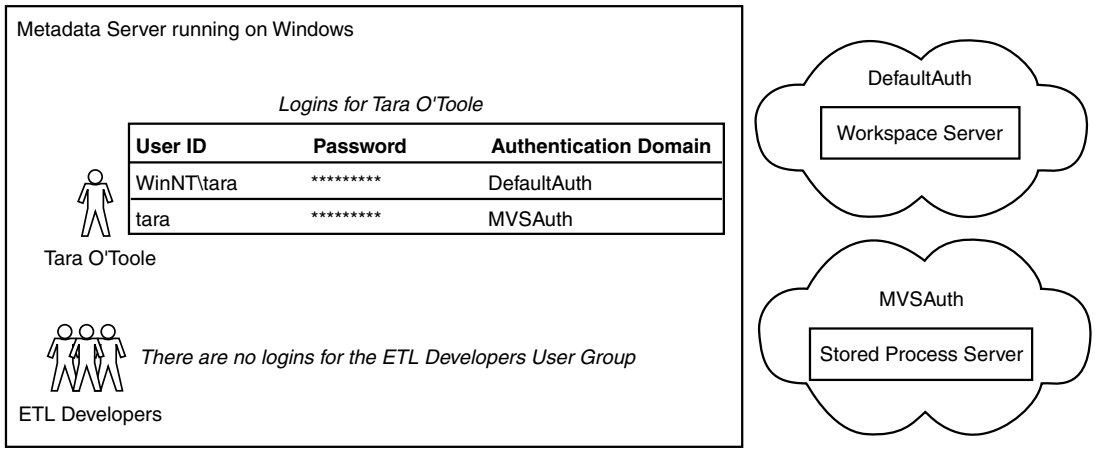
Note: If all of the applications that Tara uses can cache credentials, then Tara's login does not have to include a password. △

In this deployment, no logins have been defined for the ETL Developers user group. This user group exists to simplify administration of access controls.

Mixed Platform Environments

In a multi-host environment, you will usually need more than one authentication domain. For example, if you modify the previous deployment by moving the stored process server to z/OS, then you will need an additional authentication domain, because your users access servers on z/OS using different credentials than they use on Windows. In the metadata, you need to link the stored process server to the logins that contain credentials for accessing that server. You create this link by associating both the server and the logins with a new authentication domain. The following figure depicts this modification to the previous deployment.

Figure 1.9 Mixed Environment, Two Authentication Domains



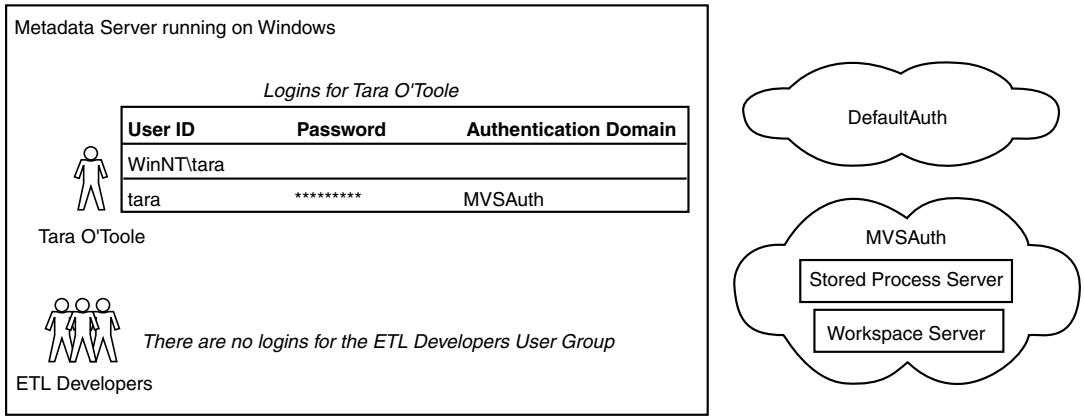
In this figure, a new authentication domain named MVSAuth has been defined, and the stored process server has been registered in that authentication domain. Two logins have been defined for Tara:

- The first login is for the DefaultAuth authentication domain. This login is used by the metadata server to determine Tara's identity and by the workspace server during additional authentication.
- The second login is for the MVSAuth authentication domain. This login enables Tara to access the stored process server during additional authentication.

Note: If the applications that Tara uses cache her credentials, then Tara can access the workspace server using credentials that are cached from initial authentication. In this scenario, Tara's first login would not have to include a password. △

The next figure depicts the deployment after you move the workspace server to z/OS. Now only the metadata server is running under Windows. All of the other servers are running under z/OS and are registered in the MVSAuth authentication domain.

Figure 1.10 Mixed Environment, One Authentication Domain



In this figure, the DefaultAuth authentication domain still exists, but it is not associated with any servers or logins. Tara still owns two logins, but it is no longer essential to include a password or an authentication domain in the first login. Tara's

first login is now only used to determine her metadata identity; it is not used for any other purposes.

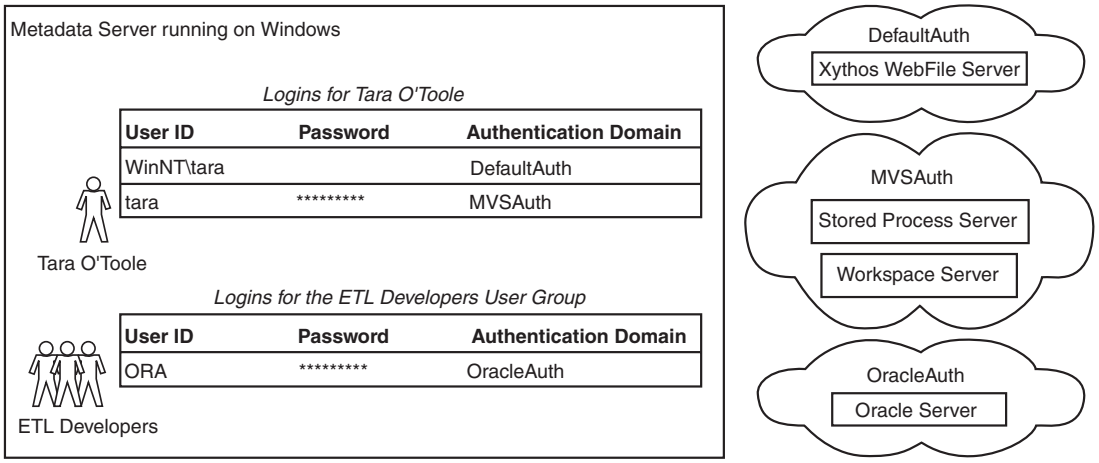
Diverse Environments

In a diverse environment, you might need more authentication domains. In this example, you add two servers to the previous deployment:

- an Oracle server that uses database authentication. When you add this server, you must add another authentication domain, because your users access the Oracle server with different credentials than they use to access the other servers. In the metadata, you must link the Oracle server to the logins that contain credentials for accessing that server. You create this link by associating both the Oracle server and the logins with a new authentication domain.
- a Xythos WebFile Server that delegates authentication to the SAS Metadata Server.* When you add the Xythos server, you do not need to add a new authentication domain, because your users will use their metadata server credentials to access the Xythos server. However, for each user who will access resources on the Xythos server, you must specify the DefaultAuth authentication domain on the login that the metadata server uses to determine that user's identity. This enables the user to authenticate to the Xythos server.

The following figure depicts the revised deployment.

Figure 1.11 Diverse Environment, Multiple Authentication Domains



In the figure, a new authentication domain named OracleAuth has been defined, and an Oracle server has been registered in that authentication domain. The Xythos WFS server has been added to the DefaultAuth authentication domain.

The metadata identity that represents Tara O'Toole owns two logins:

- The first login is used by the metadata server to determine Tara's identity. This login is now also used to enable the metadata server to authenticate Tara on behalf of the Xythos server. This use requires the first login to be assigned to the DefaultAuth authentication domain.

* This is the default configuration for authentication to a Xythos WebFile Server. More information and configuration details are provided in "Accessing a Xythos WebFile Server" on page 31.

- The second login provides access to the stored process and workspace servers that are registered in the MVSAuth authentication domain. This login functions as an outbound login (it is outbound from the metadata server), so this login includes a password to support a single sign-on approach to additional authentication.

Note: A different set of logins might be required if your metadata server uses an alternative authentication provider or your deployment includes pooled servers. △

Tara does not directly own a login that provides access to the server in the OracleAuth authentication domain, so she can access that server only if she is a member of a user group that owns an appropriate login. In this example, Tara is a member of the ETL Developers user group, so she can use that group's shared login to get to the Oracle server in the OracleAuth authentication domain. If you give the ETL Developers group a login for the OracleAuth authentication domain, you should not also give Tara a login for the OracleAuth authentication domain. If more than one login for a particular authentication domain is available to Tara, then a requesting application might not be able to determine which set of credentials to use.

Note: In order to access the Oracle server from SAS Data Integration Studio, Tara must be able to access both the workspace server *and* the Oracle server. △

A Closer Look: Accessing SAS Servers

Introduction to SAS Server Access Examples

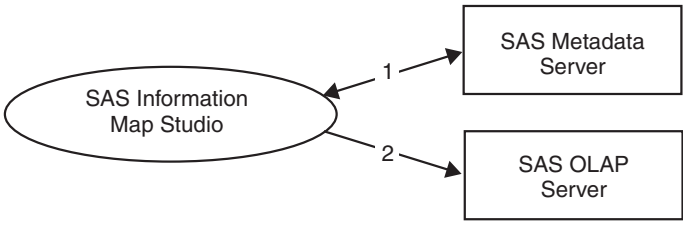
This section contains specific examples of additional authentication from various applications to SAS OLAP Servers, SAS Workspace Servers, and SAS Stored Process Servers. The examples assume these conditions:

- The deployment includes the standard, required accounts that are described in the pre-installation checklist.
- The user has completed initial authentication.
- The user has a metadata identity.
- The logins that the user needs for additional authentication are defined in the metadata repository.
- The accounts that the user needs have been established with the appropriate authentication providers.
- Each SAS OLAP Server, SAS Workspace Server, and SAS Stored Process Server is registered in the metadata and is associated with an appropriate authentication domain.

Accessing a SAS OLAP Server

This example describes the additional authentication process from SAS Information Map Studio to a SAS OLAP Server. The process is initiated when a user makes a request to access cubes from SAS Information Map Studio. The process is depicted in the following figure.

Figure 1.12 Additional Authentication to a SAS OLAP Server



The prerequisites for accessing a SAS OLAP Server are that the metadata server and the OLAP server must be running (the metadata server must be started before the OLAP server).

The numbers in the diagram correspond to these activities:

1. SAS Information Map Studio goes to the metadata server to get the user's credentials for the SAS OLAP Server. As the requesting client, the user must have ReadMetadata permission to the SAS OLAP Server definition. The user (or a group to which the user belongs) must have a login for the authentication domain that is associated with the SAS OLAP Server definition. The user ID and password in that login must correspond to an account that has been established with the SAS OLAP Server's authentication provider.

   Note: If the application can use the user's cached credentials to access the SAS OLAP Server, then this step is omitted. △

2. SAS Information Map Studio provides the user's credentials to the SAS OLAP Server. The SAS OLAP Server then authenticates the user against its authentication provider.

Accessing a SAS Workspace Server

This example describes the additional authentication process from SAS Web Report Studio to a SAS Workspace Server. The process is initiated when a user makes a request that requires access to a workspace server from SAS Web Report Studio.

Note: In this example, the SAS Workspace Server is not part of a pool. The next example describes the process for accessing a pooled workspace server. △

These are the prerequisites for accessing a workspace server:

- The metadata server must be running.
- The object spawner must be running and must have been started after the metadata server was started.
- When it initializes, the object spawner must be able to get information about the workspace server from the metadata server. To get this information, the object spawner connects to the metadata server as the SAS Trusted User (which corresponds to the sastrust account on the metadata server).*

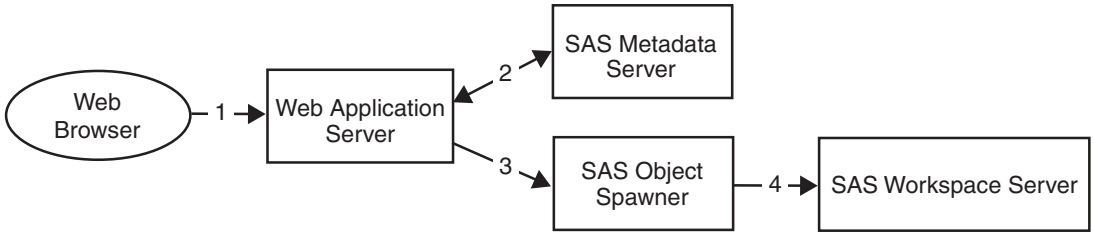
  By default, the SAS Trusted User can see the workspace server definition because sastrust is a member of the SAS System Services user group, which has ReadMetadata access to the repository. As you set access controls, you must

* As explained in "Minimizing the Availability of Accounts" on page 76, this connection does not make use of any *trusted user* functionality.

ensure that the SAS System Services group does not lose its ReadMetadata access to the workspace server definition.

The following diagram depicts the process that is initiated by the user's request.

Figure 1.13 Additional Authentication to a SAS Workspace Server



The numbers in the figure correspond to these steps:

1. The user's Web browser sends the request to the SAS Web Report Studio application.
2. The application goes to the metadata server to get the user's credentials for the workspace server. The application must find a login that is associated with the workspace server's authentication domain and is owned by the user (or by a user group to which the user belongs).

   Note: If the application can use the user's cached credentials to access the workspace server, then this step is omitted. △
3. The application asks the object spawner to launch a workspace server, using the user's credentials.
4. The object spawner uses the credentials that were obtained from the metadata server to authenticate the user (using host authentication). The object spawner then launches a workspace server for the user.

Accessing a Pooled SAS Workspace Server

This example describes the additional authentication process from SAS Web Report Studio to a pooled SAS Workspace Server.

When you set up pooling, you assign one login to each puddle within the pooled logical workspace server. Each puddle login corresponds to an account that has been established in the host environment of the workspace server. When an application asks the object spawner to launch an additional physical workspace server into the pool, the application must provide the user ID and password for one of the puddle logins. Before the object spawner launches the physical workspace server, the object spawner checks those credentials against the host operating system.

If a user makes a request that requires access to the pooled workspace server, the request does not trigger any further authentication. For this reason, a user does not have to have a host account in order to access a pooled workspace server. A user does, however, have to be a member of at least one user group that is associated with at least one puddle in the pool of workspace servers (as this example explains).

These are the prerequisites for accessing a pooled workspace server:

- The metadata server must be running.
- The object spawner must be running and must have been started after the metadata server was started.

- When it initializes, the object spawner must be able to get information about the workspace server from the metadata server. To get this information, the object spawner connects to the metadata server as the SAS Trusted User. As explained in the previous example, the SAS Trusted User can see the workspace server definition because sastrust is a member of the SAS System Services user group, which has ReadMetadata access to the repository.
- The requesting application must be able to obtain all of the puddle logins from the metadata repository. This enables the requesting application to provide the object spawner with the credentials that the object spawner will use to launch the workspace servers. The account that the requesting application uses to retrieve the puddle logins from the metadata repository is called the pool administrator.
- The logical workspace server must be configured for pooling.

The following figures depict the process that is initiated by a user's request.

Figure 1.14 Accessing an Existing Pooled SAS Workspace Server

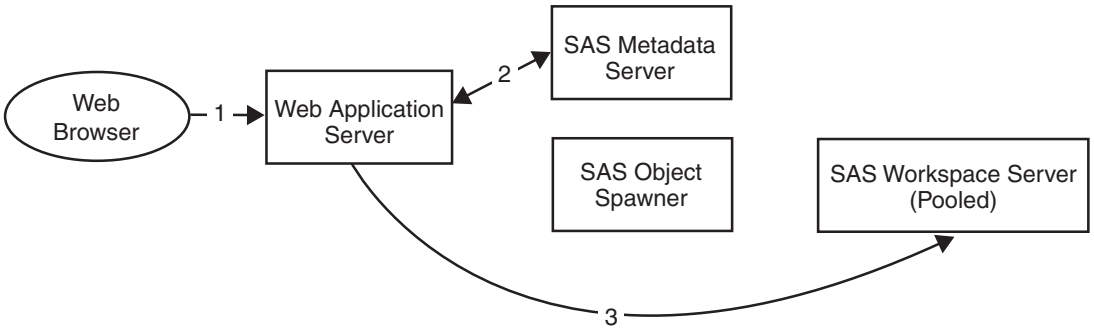
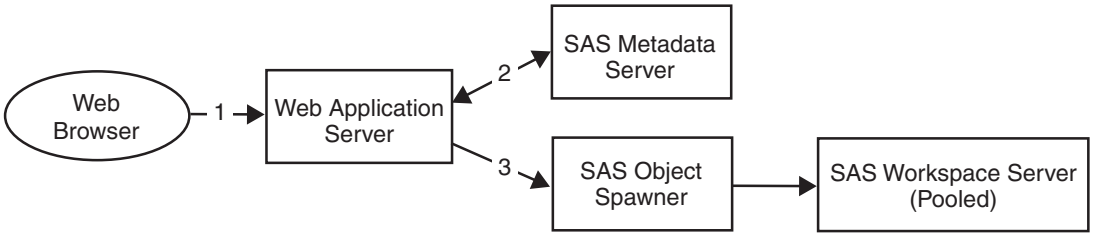


Figure 1.15 Accessing a Newly Launched Pooled SAS Workspace Server



The numbers in the figures correspond to these steps:

1. The user's Web browser sends the request to the SAS Web Report Studio application.
2. SAS Web Report Studio checks the user's group membership information in the metadata repository in order to determine which puddles the user is allowed to use.

   Note: In the metadata, each puddle is assigned to one user group. If a user does not belong to any user groups that are assigned to a puddle, then the user will not be able to connect to a workspace server. △
3. SAS Web Report Studio performs one of the following actions:
   - If there is an available workspace server in a puddle that the user is allowed to use, then SAS Web Report Studio sends the request to that workspace server.

- If there are no available workspace servers in any of the puddles that the user is allowed to use, then SAS Web Report Studio asks the object spawner to launch a new workspace server in an appropriate puddle.

Accessing a SAS Stored Process Server

This example describes the additional authentication process from the SAS Add-In for Microsoft Office to a SAS Stored Process Server. The process is initiated when a user makes a request that requires access to a stored process server from the SAS Add-In for Microsoft Office.

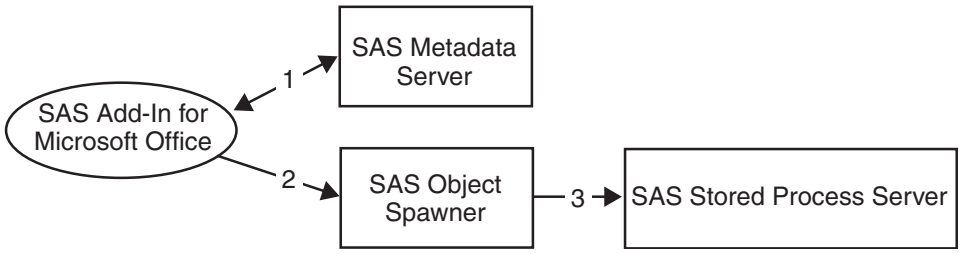
These are the prerequisites for accessing a stored process server:

- The metadata server must be running.
- The object spawner must be running and must have been started after the metadata server was started.
- When it initializes, the object spawner must be able to get information about the stored process server from the metadata server. To get this information, the object spawner connects to the metadata server as the SAS Trusted User (sastrust). This user must be able to see the stored process server definition *and* to use the sassrv login (under which the stored process server runs).
  1. By default, the SAS Trusted User can see the stored process server definition because sastrust is a member of the SAS System Services user group, which has ReadMetadata permission for the repository. As you set access controls, you must ensure that the SAS System Services group does not lose its ReadMetadata access to the stored process server definition. To learn how to manage access to server definitions, see "Access Requirements for Server Definitions" on page 127.
  2. The SAS Trusted User can use the sassrv login because sastrust is a member of the SAS General Servers user group, which owns the sassrv login.

     Note: Only members of the SAS General Servers group can use the sassrv login. An *unrestricted user* such as the SAS Administrator (which corresponds to the sasadm account on the metadata server) cannot obtain any passwords, so you should not use the sasadm account in place of the sastrust account. △

The following figure depicts the process that is initiated by a user's request.

Figure 1.16 Additional Authentication to a SAS Stored Process Server



The numbers in the figure correspond to these steps:

1. SAS Add-In for Microsoft Office goes to the metadata server to get the user's credentials for the stored process server. The application must find a login that is associated with the stored process server's authentication domain and is owned by the user (or by a user group to which the user belongs).

Note: If the application can use the user's cached credentials to access the stored process server, then this step is omitted. △

2 The application asks the object spawner for a stored process server.

3 The object spawner either uses an existing stored process server or launches a new one. The stored process server uses host authentication to verify the user's identity and then runs under the sassrv account.

A Closer Look: Accessing Third-Party Servers

Introduction to Third-Party Server Access Examples

This section contains specific examples of additional authentication from various applications to third-party database servers and WebDAV servers. The examples assume these conditions:

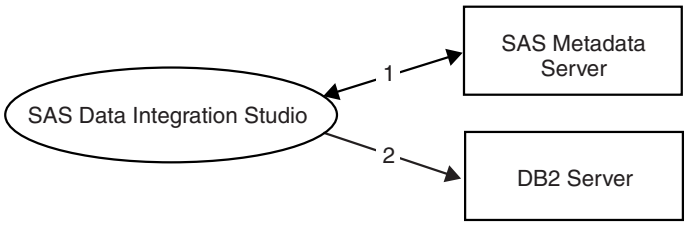
- The deployment uses the standard, required accounts that are described in the pre-installation checklist.
- The user has completed initial authentication
- The user has a metadata identity.
- The logins that the user needs for additional authentication are defined in the metadata repository.
- The accounts that the user needs have been established with the appropriate authentication providers.
- Each third-party server is registered in the metadata and is associated with an appropriate authentication domain.

Accessing a DB2 Database

This example describes the additional authentication process from SAS Data Integration Studio to a DB2 database, using the SAS/ACCESS Interface to DB2.

The process is initiated when a user makes a request to access DB2 data from SAS Data Integration Studio. The following figure depicts the process.

Figure 1.17 Additional Authentication to a DB2 Database



The numbers in the figure correspond to the following activities:

1 SAS Data Integration Studio goes to the metadata server to get the user's credentials for the DB2 system. As the requesting client, the user must have

ReadMetadata access to the DB2 server definition. The user (or a group to which the user belongs) must have a login for the authentication domain that is associated with the DB2 server definition. The user ID and password in that login must correspond to an account that has been established with the DB2 server.

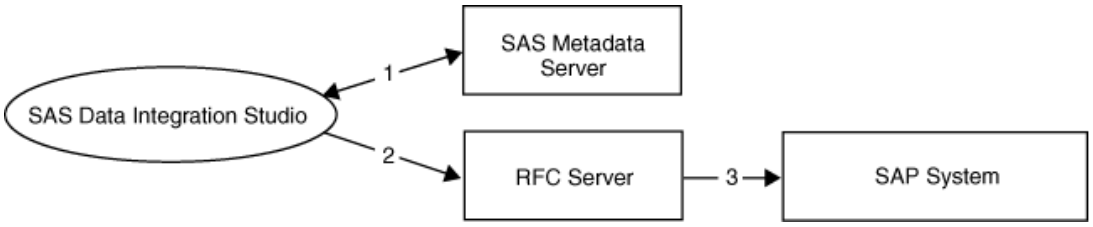
2 SAS Data Integration Studio provides the user's DB2 credentials to the DB2 server. The DB2 server verifies that those credentials correspond to an existing DB2 account.

Accessing an SAP System

This example describes the additional authentication process from SAS Data Integration Studio to an SAP system, using the SAS Data Surveyor for SAP.

The process is initiated when a user makes a request to access SAP data from SAS Data Integration Studio. The following figure depicts the process.

Figure 1.18 Additional Authentication to an SAP System



The numbers in the figure correspond to the following activities:

1 SAS Data Integration Studio goes to the metadata server to get the user's credentials for the SAP system. As the requesting client, the user must have ReadMetadata access to the SAP server definition. The user (or a group to which the user belongs) must have a login for the authentication domain that is associated with the SAP server definition. The user ID and password in that login must correspond to an account that has been established with the SAP system.

2 SAS Data Integration Studio provides the user's SAP credentials to the Remote Function Call (RFC) server.

3 The RFC server passes the SAP credentials to the SAP system, which verifies that those credentials correspond to an existing account on the SAP system.

Accessing a Xythos WebFile Server

The process for accessing a Xythos WebFile Server differs from the process for accessing other servers in some important ways. For this reason, before presenting an example of how this process works, this topic explains how user credentials for Xythos are acquired and verified.

The first step in the authentication process is for the application (the SAS Information Delivery Portal in this example) to acquire the requesting user's credentials.

- In the default configuration, the SAS Information Delivery Portal acquires the requesting user's credentials for the Xythos server by caching the credentials that the user supplied when logging on.
- In an alternate configuration, the SAS Information Delivery Portal retrieves the requesting user's credentials for the Xythos server from the metadata repository.

The SAS Information Delivery Portal determines the authentication domain for the Xythos server by checking a configuration file, rather than by examining metadata that describes that server.

The second step in the authentication process is for the target server (the Xythos WebFile Server) to verify the acquired credentials against its authentication provider.

- In the default configuration, the SAS User Customization for Xythos WFS uses the SAS Metadata Server as its authentication provider (this enables you to avoid maintaining an additional store of user information in the Xythos WebFile Server). In this process, the metadata server uses its authentication provider to verify the acquired credentials.

  After the authentication provider of the metadata server verifies the requesting user's credentials, the SAS User Customization for Xythos WFS must locate a login that is owned by the requesting user and associated with the authentication domain of the Xythos server. If no such login exists, then the user cannot connect to the Xythos server. In the default configuration, the Xythos server is associated with the DefaultAuth authentication domain.

- In an alternate configuration, the SAS User Customization for Xythos WFS first retrieves (from the metadata server) the requesting user's login for the authentication domain with which the Xythos server is associated. The SAS User Customization for Xythos WFS then authenticates the requesting user by determining whether the password in the retrieved login is the same as the password that was provided by the connecting client (the SAS Information Delivery Portal in this example). This process requires that the requesting user's login for the authentication domain of the Xythos server includes a password.

The following example explains the configuration details and illustrates the additional authentication process from the SAS Information Delivery Portal to a Xythos WebFile Server. The example assumes that you are using the default configuration, which includes these settings:

- In your SAS Information Delivery Portal configuration file, the authentication domain of the WebDAV server is the same as the cached credentials authentication domain. The **`install.properties`** file in the **`PortalConfigure`** directory includes these lines:

  ```
  $DAV_DOMAIN$=DefaultAuth
  $SERVICES_OMI_DOMAIN$=DefaultAuth
  ```

- In your SAS User Customization for Xythos WFS configuration file, the authentication domain of both the Xythos server and the metadata server is DefaultAuth. The **`saswfs.properties`** file in the **`wfs-4.0.48`** directory includes these lines:

  ```
  com.sas.wfs.domain.dav=DefaultAuth
  com.sas.wfs.domain.metadata=DefaultAuth
  ```

- The Xythos WebFile Server is *not* configured to force DIGEST HTTP authentication.

  Note: By default, the SAS User Customization for Xythos WFS configures Xythos to use BASIC authentication. This is the preferred configuration. △

These configuration files are created for you based on values that you supply during installation of the SAS Information Delivery Portal and the SAS User Customization for Xythos WFS. The following tables document how the values that you supply during installation correspond to the variables in the configuration files.

Table 1.4 Installation of the SAS Information Delivery Portal

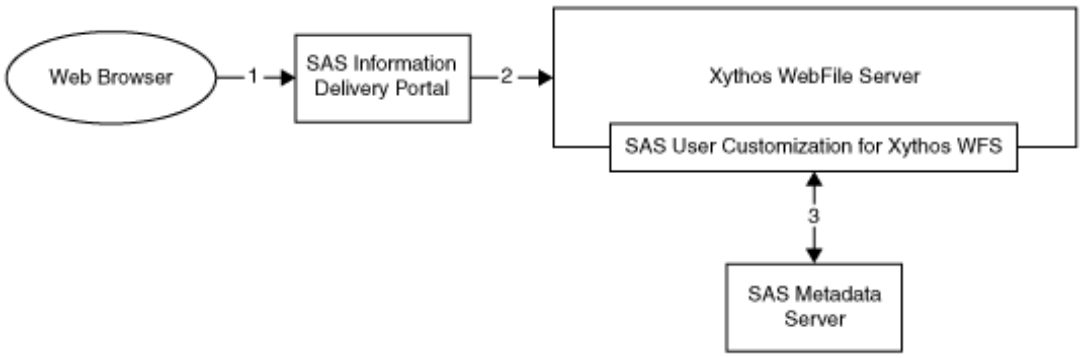
Value That You Supply during Installation	Line That Is Generated in the PortalConfigure\install.properties file
Enter the authentication domain for the SAS Metadata Server > *DefaultAuth*	$SERVICES_OMI_DOMAIN$=DefaultAuth
Enter the authentication domain for the WebDAV Server >*DefaultAuth*	DAV_DOMAIN=DefaultAuth

Table 1.5 Installation of the SAS User Customization for Xythos WFS

Value That You Supply during Installation	Line That Is Generated in the wfs-4.0.48\saswfs.properties file
Enter the authentication domain for the SAS Metadata Server > *DefaultAuth*	com.sas.wfs.domain.metadata=DefaultAuth
Enter the authentication domain for the Xythos WFS WebDAV Server >*DefaultAuth*	com.sas.wfs.domain.dav=DefaultAuth

The following figure depicts the process that is initiated when a user makes a request to access a resource that is stored on the Xythos WebFile Server.

Figure 1.19 Additional Authentication to a Xythos WebFile Server



The numbers in the figure correspond to the following activities:

1 From a Web browser, the user makes a request to the SAS Information Delivery Portal for a resource that is stored in a WebDAV area on a Xythos WebFile Server.

2 The SAS Information Delivery Portal sends the user's credentials to Xythos for authentication. In this example, the default configuration is used, so cached credentials are used.

Note: If the **`PortalConfigure\install.properties`** file does not assign both the Xythos server and the metadata server to the same authentication domain, then cached credentials are not used. Instead, the SAS Information Delivery Portal gets the user's credentials for the Xythos server from the metadata server. In this process, the SAS Information Delivery Portal searches the metadata repository for credentials that are associated with the authentication domain that is specified in the DAV_DOMAIN= setting in the SAS Information Delivery Portal's **`install.properties`** file. △

3 The SAS User Customization for Xythos WFS sends the user's credentials to the metadata server for authentication. In this example, the default configuration is used, so the metadata server's authentication provider verifies the user's identity. After the credentials are verified, the SAS User Customization for Xythos WFS verifies that the user has a login for the authentication domain of the Xythos server. In this example, the default configuration is used, so the Xythos server is associated with the DefaultAuth authentication domain.

Note: If the **`wfs-4.0.48\saswfs.properties`** file does not assign the Xythos server and the metadata server to the same authentication domain, or if the Xythos server is configured to force DIGEST authentication, then the user's identity is not verified by the metadata server's authentication provider. Instead, the credentials are retrieved from the metadata repository and verified against the credentials that are provided by the SAS Information Delivery Portal. This requires that the user's login for the authentication domain of the Xythos server includes a password. △