DATAFLUX
data management

DataFlux® Authentication Server 3.2

Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2013. *DataFlux® Authentication Server 3.2: Administrator's Guide*. Cary, NC: SAS Institute Inc.

DataFlux® Authentication Server 3.2: Administrator's Guide

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, June 2013

Table of Contents

What's New in DataFlux Authentication Server 3.2

Overview

The DataFlux Authentication Server 3.2: Administrator's Guide contains the following changes and enhancements:

- DataFlux Secure installed by default.
- Revised License
- New website for access to documentation.

DataFlux Secure Installed by Default

The DataFlux Secure software is now installed by default when you install the DataFlux Authentication Server. In previous releases, DataFlux Secure was purchased separately.

DataFlux Secure is installed in a disabled state, so additional configuration is required only if you want to use a level of encryption other than the default SASPROPRIETARY. If you upgrade to 256-bit AES encryption, you need to make the same upgrade to the DataFlux clients and servers that will connect to your Authentication Server.

Revised License

A single SAS SETINIT is now the sole license for the DataFlux Authentication Server. Formerly, a primary and a secondary license were available, one from SAS and one from DataFlux. The license change affects the License configuration option set. In that option set, the sole value for the Provider option is SAS. The Secondary option is now ignored. The License option set is specified in the Authentication Server configuration file as_serv_aspsql.xml.

New Website for Documentation

All of the documentation for DataFlux products is now provided by SAS Customer Suppport, at the following locations:

- Documentation
- Install Center
- System Requirements

Information on installation and system requirements is no longer provided in this document.

Recommended Reading

This document might reference other publications including:

DataFlux Data Management Studio User's Guide
DataFlux Data Management Studio Installation and Configuration Guide
DataFlux Data Management Server Administrator's Guide
DataFlux Secure Administrator's Guide
DataFlux Web Studio User's Guide
DataFlux Web Studio Installation and Configuration Guide
SAS Federation Server Administrator's Guide
SAS Drivers for Federation Server User's Guide

See also the Help for the preceding products and for the SAS Federation Server Manager.

For a complete list of SAS publications, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Publishing Sales Representative:

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513-2414
Telephone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Overview of the DataFlux Authentication Server

Purpose

The DataFlux Authentication Server provides a central point of authentication management across multiple domains and multiple operating environments. Features include:

Centralized Authentication - the server accesses native authentication mechanisms, such as Windows Active Directory or LDAP, to verify the identity of the users who create jobs, run jobs, or access data collections.

Centralized Management of Users and Groups - the Authentication Server manages user and group definitions that form the basis for authorization for the following servers and clients:

- Data Management Server
- SAS Federation Server
- Web Studio Server
- Data Management Studio
- Web Studio
- SAS Federation Server Manager

Single or Reduced Sign-On - the Authentication Server enables authenticated users to connect across domains to servers and native databases without submitting additional credentials.

Platform Architecture

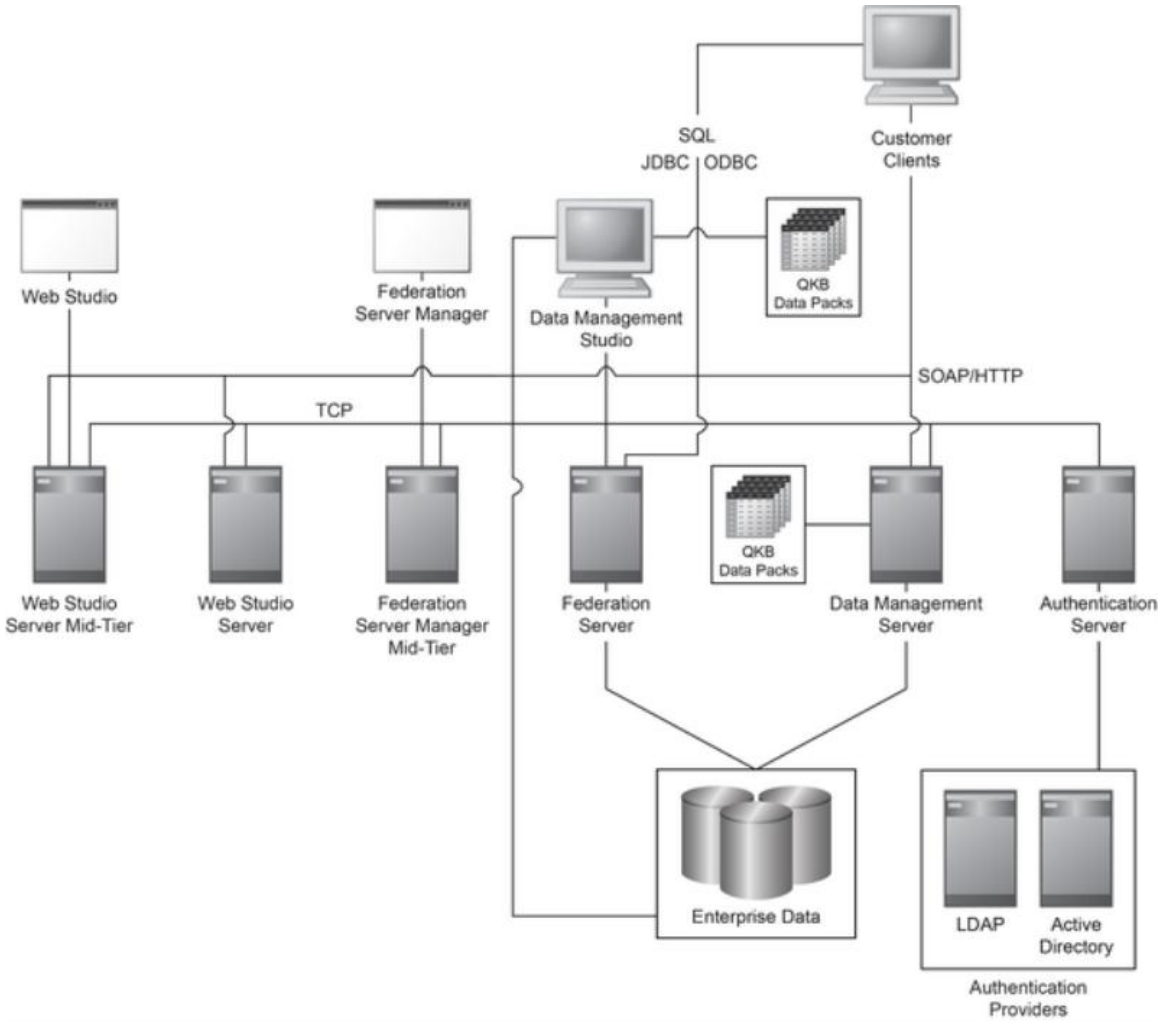
Each Authentication Server can authenticate in as many as three separate domains. The authentication process works with your existing authentication providers to validate submitted logins. Supported authentication providers include LDAP, Active Directory, and the host authentication providers that are supplied in the Windows, UNIX, or Linux operating environments.

The administrative interface to the Authentication Server is provided in the Authentication Riser in Data Management Studio.

The Authentication Server maintains a transactional database for users, groups, domains, logins, and shared logins. A shared login enables multiple users to share a single account on one of your relational databases. The transactional database is stored on the local host of the Authentication Server. You can manage the transactional database using the ASBATCH utility.

An alternative to the transactional database is to maintain users, groups, domains, logins, and shared logins in Oracle. Using Oracle, you can configure multiple Authentication Servers to share a single set of system tables.

The following diagram shows how the Authentication Server connects to clients and servers.



DataFlux Secure: Encryption, SSL, and FIPS Compliance

Starting in the 3.2 release, the Authentication Server is installed with the DataFlux Secure software by default. The additional security features are initially disabled. The encryption algorithm SASProprietary is enabled by default. To enable enhanced security, you configure the Authentication Server as directed in the *DataFlux Secure: Administrator's Guide*.

The DataFlux Secure software provides the following features:

- 256-bit AES encryption for network traffic and password storage.

- Single Sockets Layer protection for communication with SSL-enabled authentication providers.
- Optional compliance with Federal Information Processing Standard FIPS 140-2.

How it Works

Connect to a SAS Federation Server

You connect to a SAS Federation Server from DataFlux Data Management Studio. The connection enables you to access federated data, run jobs, and open connections to your relational databases. The process illustrates how the Authentication Server helps you establish a connection to a Federation Server:

1. In the Administration riser in Data Management Studio, an administrator creates your user definition, then you add one or more logins to that user definition.
2. The administrator adds your user definition to groups and shared logins.
3. Also in Data Management Studio, you open a connection to the Authentication Server. The Authentication Server authenticates your login. The Authentication Server sends a handle to Data Management Studio so that it can read the logins in your user definition.
4. You request a connection to the Federation Server.
5. Data Management Studio retrieves from the Authentication Server your login for the Federation Server domain.
6. Studio connects to the Federation Server using the login from the Authentication Server.
7. The Federation Server connects to Authentication Server using your login.
8. The Authentication Server authenticates the login and returns a handle to the Federation Server. At this point, you can open connections to databases or gain access to resources on the Federation Server.

If you connect to a Federation Server without first connecting to an Authentication Server, you are asked to supply credentials to the Federation Server.

Connect to a Database

After you connect to a SAS Federation Server, you can open a connection to database using the following process:

1. In Studio, you request a connection to a database.
2. The Federation Server uses your handle to retrieve from the Authentication Server a login to the database.

   Depending on the DSN configuration for that database, the login that is retrieved can be a personal login (from your user definition, for that domain),

or a shared login (for which your user definition has been designated as a consumer).

3. The Federation Server uses the login from the Authentication Server to connect to the database, without asking you to enter another login.

Configuring the DataFlux Authentication Server

Upgrade Notes

If you are upgrading an existing Authentication Server to a new release, be sure to follow these general steps:

1. Install the new server alongside your existing server.
2. Migrate your existing users, groups, domains, logins, and shared logins to the new release.
3. lPromote your server configuration, including authentication providers.
4. Test your new server.
5. Uninstall your old server.

> **Note:** The uninstall process for the Authentication Server removes all installed files, including the file ASDB.TDB, which contains the system tables for the default transactional database.

Migrate Authentication Server Data from 2.1.*x* to 3.*x*

Follow these steps to ensure that your Authentication Server 3.*x* has the same data and configuration as your Authentication Server from version 2.1.*x*

1. Install Authentication Server 3.*x* alongside your existing Authentication Server 2.1.*x*.

2. If your 2.1.*x* Authentication Server uses Oracle to manage users, groups, domain, logins and shared logins, then no specific migration tasks are required. Your 3.*x* Authentication Server can use the 2.1.*x* system tables.

   To configure your 3.*x* server to use Oracle rather than the transactional database that is configured by the installation process, see Configure Oracle to Store Users, Groups, Domains, Logins, and Shared Logins.

   Note: Before you configure your 3.*x* Authentication Server to use Oracle, make sure that you first backup your existing (2.1.*x*) Oracle data.

   Go to step 5.

3. In the Windows operating environment, click the shortcut **Migrate 2.1.x Database**. The shortcut was created by the 3.*x* installation process. Clicking the shortcut displays the Migration Utility Wizard, which will prompt you for the location of your 2.1.*x* transactional database. The wizard copies your 2.1.*x* system tables to the 3.*x* directory and then updates the system tables as required.

4. In the UNIX or Linux operating environment, execute the following command:

   ```
   install-path/bin/dasadmin migrate
   ```

   The migration process copies your 2.1.*x* system tables to your 3.*x* directory, displays any migration errors, and concludes with:

   ```
   Authentication Server DB Migration completed successfully.
   ```

5. Compare the 2.1.*x* and 3.*x* configuration files named as_serv_aspsql.xml. The path to the 3*x* file is as follows:

   ```
   install-path\etc\as_serv_aspsql.xml
   ```

   Update the 3.*x* configuration file as needed to apply your 2.1.*x* configuration options. Pay particular attention to the options that apply to authentication.

   For information on configuration options in the configuration files, see Appendix: Configuration File Reference.

   Close your 2.1.*x* configuration file without saving any changes.

6. Update your 3.*x* configuration file to meet any new requirements. For example, you might want to update the location of the Authentication Server's log file.

7. Save and close your 3.*x* configuration file.

8. Start your Authentication Server 3.*x*.

Promote Authentication Server Content to a New Release

Follow these steps to promote your Authentication Server transactional database and server configuration from 3.1 to 3.2:

1. Stop your Authentication Server 3.1 and 3.2.
2. Copy the 3.1 SYSCAT.TDB and SQL_LOG.TDB files to the 3.2 installation location. See *install-path*/var.
3. Compare the configuration files in the 3.1 and 3.2 releases, update the 3.2 files as needed, then close all of the configuration files.
4. Start your Authentication Server 3.1 and 3.2. Validate that the 3.2 database and server configuration contain the content that was promoted from 3.1.

Add, Edit, or Delete an Authentication Server Definition

Follow these steps to add, edit, or delete an Authentication Server definition in your instance of Data Management Studio. You need to add a server definition before you can choose a default server or connect to a server.

1. In Data Management Studio, expand the Administration riser. If you do not have an Administration riser, disconnect from your current Authentication Server. Click on the **X** icon in the file tab in the top left corner of the window.
2. In the Administration riser, to add a new server definition, right-click **Authentication Servers** and select **New**. Enter your personal name for the new server, along with a description, a server host name, and a port number.

   The server host name needs to be fully qualified, with all of the domain information that is necessary for your local host to connect to the server. For example, if the local host is part of the same domain as the server host, then the server name might be d2251.us.myco.com.
3. To edit or delete an existing server definition, right-click the server definition in the Authentication Servers riser and select **Edit** or **Delete**.

Select a Default Authentication Server

When you select a default Authentication Server, you will be prompted to log in when you start Data Management Studio.

Before you can select a default server, you must first create a server definition.

To select a default server, right click the server definition in the Administration riser and select **Set as Default**.

You can also:

1. Click **Authentication Servers**
2. Select a server in the information pane.
3. Click the star symbol, which is entitled **Set the server as the default**.

Connect to an Authentication Server

You connect to an Authentication Server to view and edit logins, users, groups, domains, and shared logins.

Before you can connect, you must first create a server definition.

To connect to an Authentication Server:

1. Expand the Authentication Servers riser.
2. Right-click the server name.
3. Select **Open**.
4. In the Login dialog box, supply a user ID, domain, and password. Use either a login that has been associated with a user definition, or use a login that is valid on the Authentication Server host.

If you log in without a user definition, but with a host login, you can see all users, groups, and domains in that server's authentication data store.

If your login is associated with a user definition on that server, you can edit your logins in that user definition.

After you connect you will receive new risers: Domain, Users, Groups, and Shared Logins.

To disconnect from a server, click red **X** in the server tab in the top left corner.

About the Authentication Server Configuration Files

The Authentication Server configuration files work with the options on the server invocation command to tailor the server to meet your needs. The options in the configuration files determine the server's operational parameters, such as authentication mechanisms and domains, administrative user IDs, log level, and encryption level.

The default server configuration is set during installation. The default configuration is fully operational. Default authentication uses the current authentication mechanism and domain of the server host.

Most of the configuration files are stored by default in *install-path*`\etc`, or in a similar path on UNIX or Linux. Files with other locations are listed below.

It is recommended that you set authorizations to protect these files from general access, as described in Configure Authorizations in the Operating System

as_log.xml - defines the log level for the Authentication Server. The Authentication Server generates a log file by default. The default log file records user connections and server errors. You can configure the log file as needed to capture additional information, as described in Administer Log Files.

as_serv_aspsql.xml - defines values for the majority of the configuration options.

as_serv_aspsql_schema_trans.xml - describes how to build system tables for the default transactional database. This file is not intended to be edited.

as_serv_aspsql_schema_ora.xml - describes how to build the schema and system tables for an Oracle database. This file is not intended to be edited.

as_serv_aspsql_schema_odbc_ora.xml - when you use an Oracle database, and when you specify the location of the Oracle database server using an ODBC DSN (rather than a path), this file tells the server how to build the Oracle schema and and system tables. This file is not intended to be edited.

as_serv_aspsql_scf.dat - optionally stores the Oracle credentials that are used by the Authentication Server to access users, groups, domains, logins, and shared logins on Oracle. If you use this file to store your Oracle credentials, set the values of the option CredentialsLocation accordingly in the file as_serv_aspsql.xml. This file is stored by default in *install-path*`\var`.

sasauth.conf - configures the SASAUTH software on UNIX and Linux hosts, as described in Configure the SASAUTH Authentication Utility. This file is stored by default in *install-path*`/lib`.

Note that the `*.xml` configuration files are accompanied by `*.template` files, which are used as a reference to the default configuration.

If you edit a configuration file, you need to restart the Authentication Server to put your changes into effect.

Identify Administrators

Authentication Server administrators are authorized to add, edit, and delete users, groups, domains, and shared logins.

An administrator is initially identified when you install the Authentication Server. The individual who executed the installation application becomes the administrator by default unless you specify another individual (by user ID) at that time.

After installation, you can add or delete administrators by editing the SystemUsers option in the as_serv_aspsql.xml configuration file.

Configure Encryption

By default, the DataFlux Authentication Server encrypts all of the data that is transferred between clients and the Authentication Server. You can change the default encryption level by changing the value of the Clientencryptionlevel option in the configuration file as_serv_aspsql.xml. You can choose between no encryption, login encryption, and all encryption. The default value is EVERYTHING, which encrypts all network traffic.

By default, the Authentication Server uses the SASProprietary encryption algorithm. SASProprietary uses 56-bit keys. If you install the Authentication Server with DataFlux Secure, you can upgrade to AES encryption. AES encryption uses 256-bit keys.

The DataFlux Secure software also enables the Authentication Server to communicate with authentication providers that use SSL. The Secure Sockets Layer authenticates Internet connections that use HTTPS addresses, with or without the exchange of digital certificates.

DataFlux Secure also enables your Authentication Server to run in compliance with the Federal Information Processing Standard 140-2. You optionally enable FIPS compliance when you configure DataFlux Secure. FIPS compliance is available on the Authentication Server and the SAS Federation Server. When FIPS compliance is enabled on these servers, the servers are required to communicate with Data Management Studio, Web Studio, and Data Management Server using the DataFlux drivers for ODBC or JDBC, rather than using direct platform connections.

Installing DataFlux Secure affects the values of two options in the Authentication Server's configuration file as_serv_aspsql.xml: NetworkEncryptionAlgorithm and EncryptFIPS.

> **Note:** If you install DataFlux Secure on your Authentication Server, you also need to install that software on any related instances of Data Management Studio, SAS Federation Server, and Data Management Server. All platform components are required to use the same level of encryption.

For additional information on DataFlux Secure, see the *DataFlux Secure Administrator's Guide*, which is available when you select the Information riser in Data Management Studio. To order DataFlux Secure, see download section of the DataFlux Customer Care Portal, at http://www.dataflux.com/Customer-Care/.

Configure the Shared Login Manager on the SAS Federation Server

On a SAS Federation Server, the Shared Login Manager requests outbound logins from the Authentication Server. The outbound logins enable the Federation Server to authenticate users on databases such as Oracle.

To configure the Shared Login Manager, you specify a login and a shared login key. The key grants the Shared Login Manager access to all of the shared logins that were assigned that particular key.

If you assign the Shared Login Manager a non-administrative login, access is granted only to the subset of shared logins that list the non-administrative login as a shared login manager.

To display shared login managers and shared login keys, open the SAS Federation Server Manager.

To assign a login to the Shared Login Manager on a Federation Server, follow these steps:

1. Open the SAS Federation Server Manager.
2. Connect to the Federation Server.
3. In the **Options** dialog, click the **Advanced** tab.
4. Enter the user ID and password of the login of the Shared Login Manager, as well as an optional shared login key.
5. Click **OK** to save your entries.

Configure Authorizations in the Operating System

The following tables recommend that you set read, write, and execute authorizations for certain users in certain directories. Deny directory access to all users other than those listed below.

Recommended Authorizations for Windows

Directories	User Role	Authorizations
*install-path**AuthServer*	Installer	Full control
	Process user	Read, write, execute, list folder contents
install-path\var	Installer	Full control
	Process user	Read, write, execute, list folder contents
	Person who backs up the Authentication Server	Read, list folder contents

Recommended Authorizations for UNIX and Linux

Directories	User Role	Authorizations
install-path/authserver	Installer	Read, write execute
	Process user	Read, execute
install-path/authserver/var	Installer	Read, write execute
	Process user	Read, write execute
	Person who backs up the Authentication Server	Read, execute

Configure Authentication Providers

- About Authentication Providers
- Configure Authentication in Windows
- Configure Authentication in UNIX and Linux

About Authentication Providers

Authentication takes place when a client such as Data Management Studio requests a connection to a SAS Federation Server or Data Management Server. To authenticate, the client's Authentication Server works with an authentication provider in the operating environment, in the domain that is specified in the login. Successful authentication enables the client to establish a connection to the DataFlux server.

You can configure as many as three authentication providers for each Authentication Server, one of each of the following types. Each provider needs to have its own domain.

AD - Windows Active Directory authentication.

LDAP - the Lightweight Directory Access Protocol authenticates against an LDAP authentication provider, and can also enable UNIX and Linux servers to authenticate against a Windows authentication provider.

Host - Host authentication is configured by default. In Windows, you specify host authentication when the Authentication Server host uses proprietary Windows authentication. In UNIX or Linux, you specify host authentication when you use the SASAUTH authentication utility. SASAUTH can be configured to authenticate with PAM (pluggable authentication modules).

To configure authentication providers, see Configure Authentication in Windows, or Configure Authentication in UNIX/Linux.

Configure Authentication in Windows

Follow these steps to configure authentication providers when your Authentication Server is running in the Windows operating environment. You can specify up to three authentication providers, one of each type (LDAP, Active Directory, and Host), in unique domains.

1. If the Authentication Server is running, then stop the Authentication Server.
2. Open the configuration file as_serv_aspsql.xml. The default path of that file is:
   ```
   auth-server-home\etc\as_serv_aspsql.xml
   ```
3. In the configuration file, locate the **AuthProviderDomain** option. This option associates the types of authentication providers with your domains. To learn more about this option, see Appendix: Configuration File Reference.
4. To configure a single authentication provider of the type Host, specify a domain for the HOSTUSER keyword:
   ```
   <Option name="AuthProviderDomain">HOSTUSER:your-domain-name</Option>
   ```
   The HOSTUSER domain is used by default if the login being authenticated does not contain a domain. If this option is not specified in the configuration file, then the default domain name is HOSTUSER.
5. To configure a single Active Directory authentication provider, specify a domain for the ADIR keyword:
   ```
   <Option name="AuthProviderDomain">ADIR:your-AD-domain</Option>
   ```
6. To configure a single LDAP authentication provider, specify a domain for the LDAP keyword:
   ```
   <Option name="AuthProviderDomain">LDAP:your-LDAP-domain</Option>
   ```
7. To specify two or three authentication providers, use the following syntax:
   ```
   <Option name="AuthProviderDomain">(provider1:domain1,provider2:domain2,
      provider3:domain3)</Option>
   ```
   For example:
   ```
   <Option
   name="AuthProviderDomain">(HOSTUSER:NYCWIN,ADIR:BRONXAD,LDAP:BROOKLYNKLDA
   P)</Option>
   ```
   Note: You can specify a maximum of one provider of each type, and the domains must be unique.
8. If you specified an Active Directory authentication provider, then use the SetEnv option set to specify AD_HOST and AD_PORT:
   ```
   <OptionSet name="SetEnv">
       <!-- specify a host for Active Directory authentication-->
      <Option name="AD_HOST">yoursite.yourcompany.com</Option>
      <Option name="AD_PORT">host-port-number-for-AD</Option>
   </OptionSet>
   ```

If you did not configure an LDAP authentication provider, you can proceed to step 12.

9. If you specified an LDAP authentication provider, then specify the environment variables LDAP_BASE, LDAP_HOST, and LDAP_PORT:

```
<OptionSet name="SetEnv">
   <!-- specify envars for LDAP authentication -->
  <Option name="LDAP_HOST">yourldaphost.yousite.mycompany.com</Option>
  <Option name="LDAP_PORT">host-port</Option>
  <Option name="LDAP_BASE">ou=yourorgunit,o=yourorg</Option>
</OptionSet>
```

   The environment variable LDAP_BASE defines the default base DN (Distinguished Name). The values for LDAP_BASE are site-specific.

10. If you specified an LDAP authentication provider, and if that provider does not allow anonymous binds, then specify the privileged DN in the environment variables LDAP_PRIV_DN, and LDAP_PRIV_PW:

```
<OptionSet name="SetEnv">
  <!-- specify an authorized LDAP user for simple binds -->
  <Option name="LDAP_PRIV_DN">user-name</Option>
  <Option name="LDAP_PRIV_PW">password</Option>
</OptionSet>
```

    The user that you specify must be authorized to search for users.

11. If you specified an LDAP authentication provider, and if the LDAP server is configured to use a value other than DN for authentication, then specify an alternate value in the environment variable LDAP_IDATTR:

```
<OptionSet name="SetEnv">
  <!-- specify an LDAP authentication attribute other than DN -->
  <Option name="LDAP_IDATTR">CN</Option>
</OptionSet>
```

    CN is an example value. The value at your site may differ. The default value of LDAP_IDATTR is `userid`.

    Contact your site administrator to determine if additional configuration steps are required for your LDAP implementation.

12. Save and close the configuration file.
13. Start the Authentication Server.

Configure Authentication in UNIX and Linux

- Getting Started with Authentication in UNIX and Linux
- Configure LDAP Authentication
- Configure AD Authentication
- Configure SASAUTH Authentication
- Configure SASAUTH for PAM

Getting Started with Authentication in UNIX and Linux

When you install an Authentication Server on a UNIX or Linux host, you can configure a maximum of three authentication providers, in separate domains. You can configure one LDAP, one Active Directory, and one Host authentication provider. You configure multiple domains as needed to authenticate all of the users of your DataFlux Data Management Platform.

To access existing authentication providers, you configure the host of the Authentication Server accordingly. For example, if your site uses centralized LDAP authentication, then the host of the Authentication Server should be configured as an LDAP client of that central repository.

To configure authentication providers, you will need input from your network administrator so that you can apply site-specific values. Using site-specific values, you configure your LDAP and AD providers in the Authentication Server configuration file.

Host authentication is provided by your UNIX or Linux operating environment. To interact with the host authentication provider, the Authentication Server uses the SASAUTH utility. SASAUTH:

1. Looks up the submitted userid in a user database.
2. Compares the submitted password to the password in a password database.
3. Retrieves the UID number for the user and apply the access controls that are associated with that UID.

If your host authentication provider uses pluggable authentication modules (PAM), SASAUTH can be configured accordingly.

Configure LDAP Authentication

Follow these steps to configure an LDAP authentication provider on an Authentication Server that is installed on a UNIX or Linux host.

1. Begin by ensuring that your LDAP authentication provider is properly configured to authenticate UNIX users. In order for the Authentication Server to connect directly to the LDAP database, the database must include the required UNIX/Posix user attributes, such as UID. Most LDAP servers provide an LDAP schema that contains this information. Your LDAP database must conform to the RFC 2307 standard for UNIX user attributes.
2. If the Authentication Server is running, then stop the Authentication Server.
3. Open the configuration file *auth-server-home*`/etc/as_serv_aspsql.xml`.
4. In the option AuthProviderDomain, change the single authentication provider to `LDAP`, or add the LDAP provider and domain to your existing authentication providers:

```
<!-- single-provider syntax -->
<Option name="AuthProviderDomain">LDAP:your-ldap-domain</Option>
```

```
<!-- multi-provider syntax -->
<Option name="AuthProviderDomain">ADIR:domain1,HOSTUSER:domain2,
   LDAP:domain3</Option>
```

5. Use the SetEnv Option Set to configure the following LDAP environment variables:

   LDAP_HOST - identifies the LDAP server host.

   LDAP_PORT – the port number of the LDAP service. If LD_PORT is not defined, then the default port value is used.

   LDAP_BASE – specifies the default base DN (Distinguished Name) to use when performing LDAP operations.

```
 <OptionSet name="SetEnv">
   <Option name="LDAP_HOST">myldaphost.mysite.mycompany.com</Option>
   <Option name="LDAP_PORT">myport</Option>
   <Option name="LDAP_BASE">ou=myorgunit,o=myorg</Option>
</OptionSet>
```

6. If the LDAP server does not allow anonymous binds, then LDAP_PRIV_DN and LDAP_PRIV_PW are required. The LDAP_PRIV_DN user needs to be authorized to search for users:

   LDAP_PRIV_DN=*privileged-DN*

   DN LDAP_PRIV_PW=*password-for-privileged-DN*

```
 <OptionSet name="SetEnv">
   <Option name="LDAP_PRIV_DN">user1</Option>
   <Option name="LDAP_PRIV_PW">passwword1</Option>
</OptionSet>
```

7. If the LDAP server is configured to use a value other than `DN` (Distinguished Name) for authentication, then specify the alternate value using LDAP_IDATTR. The default value of LDAP_IDATTR is `userid`.

   LDAP_IDATTR=attribute-name

```
 <OptionSet name="SetEnv">
   <Option name="LDAP_IDATTR">CN</Option>
</OptionSet>
```

8. Consult with your network administrator to determine if any additional site-specific LDAP settings are required.
9. Save and close the configuration file.
10. Start the Authentication Server.

Configure AD Authentication

Follow these steps to configure an AD authentication provider on an Authentication Server that is installed on a UNIX or Linux host.

1. Begin by ensuring that your AD authentication provider is properly configured to authenticate UNIX users. In order for the Authentication Server to connect directly to the AD database, the database must include the required UNIX/Posix user attributes, such as UID. Most AD servers provide an AD schema that contains this information. To enable connections, install Microsoft Services for UNIX (SFU) 2 or 3 on the hosts of your AD repositories.
2. If the Authentication Server is running, then stop the Authentication Server.
3. Open the configuration file *auth-server-home*`/etc/as_serv_aspsql.xml`.
4. In the option AuthProviderDomain, change the single authentication provider to `ADIR`, or add the AD provider and domain to your existing authentication providers:

```
<!-- single-provider syntax -->
<Option name="AuthProviderDomain">ADIR:your-ldap-domain</Option>

<!-- multi-provider example -->
<Option name="AuthProviderDomain">HOSTUSER:domain2,LDAP:domain2,
   ADIR:domain3</Option>
```

5. Use the SetEnv Option Set to configure the following LDAP environment variables:

   AD_HOST - identifies the Active Directory server host.

   AD_PORT – specifies the port number for Active Directory.

```
<OptionSet name="SetEnv">
  <Option name="AD_HOST">your-AD-network-hostname</Option>
  <Option name="AD_PORT">port-number-on-AD-host</Option>
</OptionSet>
```

6. Save and close the configuration file.
7. Start the Authentication Server.

Configure the SASAUTH Authentication Utility

The SASAUTH utility is used to implement Host authentication. SASAUTH can be configured to use default authentication (known as pw) or pluggable authentication modules (PAM), or even both methods in series. SASAUTH also provides three levels of logging, and a configurable response to invalid authentications. All of these features are configured in the file sasauth.conf.

Note: The SASAUTH utility requires root authorizations.

Follow these steps to enable and configure the SASAUTH utility:

1. Stop the Authentication Server.

2. Login with root privileges, or contact your network administrator, to run the following script, which establishes root privileges for the SASAUTH utility:
```
sh> install-path/lib/sasauth.inst.sh
```

3. Execute the following script to enable the SASAUTH utility:
```
sh> install-path/bin/set_auth sasauth
```

4. Edit the SASAUTH configuration file.
```
install-path/lib/sasauth.conf
```

5. In the configuration file, the `methods` variable specifies authentication methods for the SASAUTH utility. The `methods` variable accepts the values `pw` and `pam`. Use the `pw` value for authentication via /etc/passwd. On some hosts, `pw` provides non-traditional authentication using protected password databases or other enhancements. Use the `pam` value if your site uses pluggable authentication modules.

   You can specify `pw`, `pam`, or both. If you specify both, then SASAUTH will authenticate with the first method, then attempt to authenticate with the second method if necessary.

   Specify one of the following values for the `methods` variable:

```
methods=pw
methods=pam
methods=pw pam
methods=pam pw
```

   If you specify `pam`, then you need to configure that method. See Configure SASAUTH for PAM.

6. Activate and configure the SASAUTH logging facility by enabling a log file. In the configuration file, remove a comment character and insert a path for one of the three log files, as shown in the following example.

   The following example enables the Access Log and populates a log file at the specified location:

```
#debugLog=
accessLog=/tmp/sasauth.log
#errorLog=
```

   Enable the `debugLog` only when testing or diagnosing errors.

   Note: You may need to configure the syslog on your Authentication Server host in order to collect log messages from SASAUTH.

7. To configure repeated authentication attempts, edit the options `maxtries`, `maxtriesPeriod`, and `maxtriesWait`.

   `maxtries` - specifies the number of authentication attempts allowed before a waiting period is imposed.

`maxtriesPeriod` - specifies the number of seconds that can elapse before the termination of the authentication process.

`maxtriesWait` - specifies the number of seconds that a user must wait if that user exceeds the `maxtries` value within the time limit of `maxtriesPeriod`. After the waiting period, the `maxtries` count is reset to zero.

The default values specify 5 attempts in 60 seconds, following by a waiting period of 5 minutes:

```
maxtries=5
maxtriesPeriod=60
maxtriesWait=300
```

To disable the limits on authentication retries, insert comment characters in front of each variable:

```
# maxtries=5
# maxtriesPeriod=60
# maxtriesWait=300
```

8. Save and close the sasauth.conf configuration file.
9. When it is installed on UNIX or Linux, the Authentication Server is configured by default to authenticate with the UNIXUSER domain. If you wish to change the name of the domain, open the configuration file as_serv_aspsql.xml. In that file, edit the UNIXUSER domain name in the following entry:

   ```
   <Option name="AuthProviderDomain">HOSTUSER:UNIXUSER</Option>
   ```

   Save and close the configuration file.
10. Start the Authentication Server.

Configure SASAUTH for PAM

If you configured the SASAUTH utility for host authentication, and if you specified PAM as an authentication method, then use this topic to configure PAM authentication.

PAM requires you to register the applications that use authentication services. In the operating environment, upgrade your PAM configuration to register SASAUTH. In the HP-UX, Solaris, and AIX operating environments, the PAM configuration is stored in /etc/pam.conf. In that file, you need to specify the authentication services that are used by SASAUTH, and you need to specify when SASAUTH performs authentication. These specifications are made in the module types `account` and `auth`.

> **Caution:** *PAM allows you to register `other`, which permits any application to use authentication services. The use of `other` is not recommended.*

PAM supports applications that run in both 32–bit and 64–bit environments. The SASAUTH utility has a 64-bit format. In the pam.conf configuration file, make sure that the modules that you associate with SASAUTH also have a 64–bit format.

PAM modules are usually provided in separate directories for 32–bit and 64–bit libraries. The pam.conf configuration file contains pathnames that are either relative (Solaris and AIX) or that contain a symbolic variable (HP- UX).

The entries in pam.conf that are used to register applications have the following form:

```
application-name module-type control-flag module-path options
```

Examples for Solaris:

```
sasauth auth requisite pam_authtok_get.so.1
sasauth auth required pam_dhkeys.so.1
sasauth auth required pam_unix_auth.so.1
sasauth account required pam_unix_account.so.1
```

Examples for HP/UX:

```
Sasauth account required /usr/lib/security/$ISA/libpam_unix.so.1
Sasauth auth required /usr/lib/security/$ISA/libpam_unix.so.1
```

Refer to the `man` page for PAM to ensure that you correctly register SASAUTH.

Note: On AIX, PAM is not activated by default. To activate PAM, refer to the IBM document Security Guide - Authentication Module.

In Linux operating environments, the directory `/etc/pam.d` contains one configuration file for each application that is authorized to use PAM. The name of the configuration file matches the name of the application. For SASAUTH, the configuration file is `/etc/pam.d/sasauth`. The SASAUTH configuration file needs to contain entries in the following form:

```
module-type control-flag module-path options
```

Examples for Linux:

```
#%PAM-1.0
auth sufficient pam_rootok.so
auth required pam_unix2.so nullok
account required pam_unix_acct.so
```

Configure Oracle to Store Users, Groups, Domains, Logins, and Shared Logins

Overview

By default, the Authentication Server uses a transactional database to store users, groups, domains, logins, and shared logins. You can choose to configure your Authentication Server to store these objects on an Oracle relational database. The Oracle configuration uses an Oracle driver, or one of two available ODBC drivers. The Oracle driver uses a path to access the database. The ODBC drivers use a data source name (DSN) to access the database.

After you install your Authentication Server, you update the primary Authentication Server configuration file to configure the Oracle database. If you choose an ODBC driver, you may also need to add an ODBC data source, as described later in this topic.

> **Note:** If you already have an ODBC data source, make sure that your odbc.ini file contains the required value `EnableNcharSupport=1`.

Configure the Oracle Database

1. Edit the Authentication Server configuration file shown in this Windows path:
   install-path\etc\as_serv_aspsql.xml

2. For the entity ASPSQL_SCHEMA, enter the name of the Oracle schema for the Authentication Server database.
   ```
   <!ENTITY ASPSQL_SCHEMA "oracle-schema-name">
   ```

3. Add comment tags as follows to prevent the creation of these two entities:
   ```
   <!-- <!ENTITY ASPSQL_TRANDBF "C:\Program
   Files\DataFlux\AuthServer\server1\var\asdb.tdb"> -->
   <!-- <!ENTITY ASPSQL_CONFIG_DBMS SYSTEM
   "as_serv_aspsql_schema_tran.xml"> -->
   ```

4. Remove comment tags as follows to create these three entities:
   ```
   <!ENTITY ASPSQL_ORAPATH "{TNSNames entry for your Authentication Server
   database}">
   <!ENTITY ASPSQL_CONFIG_DBMS SYSTEM "as_serv_aspsql_schema_ora.xml">
   <!ENTITY ASPSQL_CREDENTIALS_LOC "install-
   path\var\as_serv_aspsql_scf.dat">
   ```

5. Remove comment tags around the following option:
   ```
   <Option name="CredentialsLocation">&ASPSQL_CREDENTIALS_LOC;</Option>
   ```

6. If you plan to store Oracle credentials on disk, then continue with this step to create an Oracle credentials file. If you plan to enter Oracle credentials manually, then jump ahead to the next step.

   DataFlux recommends that you store the Oracle credentials file in the `var` directory, as indicated in the default value of the entity ASPSQL_CREDENTIALS_LOC. The `var` directory is recommended to receive access restrictions in the operating environment, as specified in Configure Authorizations in the Operating System. If you use a non-default storage location, be sure to specify an absolute path for the entity, rather than a relative path.

   Open the Oracle credentials file. Add Oracle credentials in the following format:

   ```
   UID=myuser;PWD=mypwd
   ```

   The credentials in the file will be encrypted when you start or restart the Authentication Server. Save and close the credentials file.

7. If you plan to enter Oracle credentials manually, rather than storing them on disk, you can use **Set Provider Credentials**, or you can write a script that runs when you start the server. If you write a script, then the script will need to prompt the user for credentials, which would then be maintained in memory until the server is restarted.

   To use **Tools ->** **Set Provider Credentials** in Data Management Studio to manually enter credentials, set a blank value for the entity ASPSQL_CREDENTIALS_LOC:

   ```
   <!ENTITY ASPSQL_CREDENTIALS_LOC "">
   ```

   To use a script to capture Oracle credentials, set the following two environment variables in the script:

   ```
   DFAS_PROVIDER_SOURCE_UID=my-Oracle-UID
   DFAS_PROVIDER_SOURCE_PWD=my-Oracle-PWD
   ```

8. If it is still open, save and close the configuration file as_aspsql.xml
9. Start or restart the Authentication Server.

Example Configuration File

The following version of the file as_serv_aspsql.xml shows typical values for a configuration that uses Oracle to store users, groups, domains, logins, and shared logins.

```
<?xml version="1.0"?>

<!--
     HOW TO CONFIGURE AN ORACLE DATA STORE:
     In order to configure the Authentication Server to use an Oracle
data store,
     search for the word "Oracle" in this file and follow the
instructions in the comments. -->

<!DOCTYPE Config [
<!-- ASPSQL Provider DBMS independent content -->
<!ENTITY ASPSQL_CATALOG "AS">
<!-- When configuring an Oracle data store, specify a valid Oracle schema
for ASPSQL_SCHEMA below -->
<!ENTITY ASPSQL_SCHEMA "">

<!ENTITY ASPSQL_SCHEMA_QUALIFIER "">
<!ENTITY ASPSQL_BT_DOMAINS "DOMAINS">
<!ENTITY ASPSQL_BT_SUBJECTS "SUBJECTS">
<!ENTITY ASPSQL_BT_GROUPS "GROUPS">
<!ENTITY ASPSQL_BT_SUBJECT_GROUPS "SUBJECT_GROUPS">
<!ENTITY ASPSQL_BT_GROUP_GROUPS "GROUP_GROUPS">
<!ENTITY ASPSQL_BT_PRINCIPALS "PRINCIPALS">
<!ENTITY ASPSQL_BT_PRINCIPAL_MAPS "PRINCIPAL_MAPS">
<!ENTITY ASPSQL_BT_GROUP_MAP_MGRS "GROUP_MAP_MGRS">
<!ENTITY ASPSQL_BT_GROUP_MAP_USERS "GROUP_MAP_USERS">
<!ENTITY ASPSQL_BT_SUBJECT_MAP_MGRS "SUBJECT_MAP_MGRS">
<!ENTITY ASPSQL_BT_SUBJECT_MAP_USERS "SUBJECT_MAP_USERS">
<!ENTITY ASPSQL_BT_VERSION "VERSION">
```

```
<!ENTITY ASPSQL_BT_SENTINEL "SENTINEL">

<!-- Transactional data store used by default -->

<!--
     Add comment tags around the following lines when configuring an
Oracle data store
-->
<!ENTITY ASPSQL_TRANDBF "C:\Program
Files\DataFlux\AuthServer\server1\var\asdb.tdb">
<!ENTITY ASPSQL_CONFIG_DBMS SYSTEM "as_serv_aspsql_schema_tran.xml">

<!--
     Remove comment tags from the following lines to configure an Oracle
data store
     and supply a valid Oracle Path
 -->
<!-- <!ENTITY ASPSQL_ORAPATH "{TNSNames entry for your Authentication
Server database}"> -->
<!-- <!ENTITY ASPSQL_CONFIG_DBMS SYSTEM "as_serv_aspsql_schema_ora.xml">
-->
<!-- <!ENTITY ASPSQL_CREDENTIALS_LOC "install-
path\var\as_serv_aspsql_scf.dat"> -->

]>
<Config name="ASConfig">
                <!-- Port to listen on -->
                <Option name="Port">21030</Option>

           <!-- Administrative account -->
     <OptionSet name="SystemUsers">
           <Option name="Account">LOCAL\tsadm</Option>
     </OptionSet>

           <OptionSet name="SetEnv">
          <Option name="FIREBIRD">C:\Program
Files\DataFlux\AuthServer\server1\lib\fbembed</Option>
           <Option name="FIREBIRD_LOG">C:\Program
Files\DataFlux\AuthServer\server1\var\log</Option>
     </OptionSet>

     <OptionSet name="PrependEnv">
                                 <Option name="Path">C:\Program
Files\DataFlux\AuthServer\server1\lib\fbembed;</Option>
     </OptionSet>

     <!-- Encryption Algorithm -->
     <Option name="NetworkEncryptAlgorithm">SASProprietary</Option>
     <Option
name="ObjectServerParms">CLIENTENCRYPTIONLEVEL=EVERYTHING</Option>

     <OptionSet name="License">
          <OptionSet name="Primary">
               <Option name="Provider">SAS</Option>
                                                        <Option
name="Location">install-path\etc\license</Option>
          </OptionSet>
                </OptionSet>

     <OptionSet name="TrustedUsers">
          <Option name="Account">DATAFLUX\dfcnn19</Option>
     </OptionSet>
```

```
        <!-- Provider name -->
        <Option name="AuthenticationProvider">ASPSQL</Option>

              <!-- Provider-specific root element -->
        <OptionSet name="ASPSQLProvider">
              <!-- System catalog and schema names -->
             <Option name="SystemCatalog">&ASPSQL_CATALOG;</Option>
             <Option name="SystemSchema">&ASPSQL_SCHEMA;</Option>
             <Option name="MinConnections">1</Option>
             <Option name="MaxConnections">2</Option>

             <!-- Remove comment tags from the following line to configure
an Oracle data store -->
             <!-- <Option
name="CredentialsLocation">&ASPSQL_CREDENTIALS_LOC;</Option> -->

              &ASPSQL_CONFIG_DBMS;
        </OptionSet>
</Config>
```

Add an ODBC Data Source on Windows

If your Authentication Server runs on Windows, follow these steps to add an ODBC data source:

1. Open the Windows application ODBC Data Source Administrator:
    a. Select **Start** > **Settings** > **Control Panel**, or use the current Windows equivalent path.
    b. Double-click **Administrative Tools**.
    c. Double-click **Data Sources (ODBC)**.
2. In the ODBC Data Source Administrator, click the **System DSN** tab, and then click **Add**.
3. In the Create New Data Source dialog, select **DATAFLUX 32-BIT Oracle**, or select **DATAFLUX 32-BIT Oracle Wire Protocol**. Click **Finish**.
4. In the driver setup dialog, enter your data source properties in the provided fields and tabs.
5. Click the **Advanced** tab, and then click **Enable N-CHAR Support**, to display a check mark for that property. This selection is required.
6. Click **OK** twice to save your changes.

Add an ODBC Data Source on UNIX or Linux

If your Authentication Server runs on UNIX or Linux, follow these steps to add an ODBC data source.

1. Run the DataFlux ODBC Configuration tool:
```
bin/dfdbconf
```
2. Enter **A** to add a new data source.

3. In the **Available Templates** list, choose **Oracle Wire Protocol [DataDirect 6.0 Oracle Wire Protocol]**. On certain versions of UNIX, you can choose **Oracle [DataDirect 6.0 Oracle]** instead.
4. Enter a value of 1 for the property **Enable N-CHAR Support**. This entry is required.
5. Enter your site's data source parameters, or press **Enter** to select default values.
6. Enter a name for the new data source.

Add a New Default Authentication Server

After you install, configure, and start a new Authentication Server, you and other users follow these steps to create a new server definition and select the new server as their default. The default server authenticates your login when you start Data Management Studio.

1. In Data Management Studio, click the **Administration** riser.
2. Right-click Authentication Server and select **New Authentication Server Connection**.
3. In the window Add Authentication Server Definition, create the server definition and test the connection.
4. In the Administration riser, right-click the new server definition and select **Set as Default**.

When you select a default Authentication Server, the client creates the following configuration file:

```
C:\Documents and Settings\userid\Application
Data\DataFlux\DMStudio\instance\etc\app.cfg
```

In this user-specific instance of app.cfg, Studio stores the following option/value pair:

```
BASE/AUTH_SERVER_LOC=auth-server-network-host-name:port

Example: BASE/AUTH_SERVER_LOC=d14885.ourCompany.com:21030
```

Note: If your user-specific instance of the app.cfg file is not removed before you upgrade Data Management Studio, the new version of Studio will attempt to authenticate with your previous default server.

You can change your default Authentication Server at any time by selecting **Set as Default**.

Administering the DataFlux Authentication Server

Start or Stop an Authentication Server in Windows

Follow these steps to start or stop an Authentication Server that is running on a Windows host.

1. On the Authentication Server host, click **Start** > **Settings** > **Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Computer Management**.
4. Expand the **Services and Applications** folder.
5. Double-click **Services**.
6. Right-click **DataFlux Authentication Server** and select **Stop** or **Start**. It is recommended that you ask all users to disconnect from the server before you stop it.

   Note: Version number differences between the Authentication Server database schema and the Authentication Server itself can terminate the start process. Major version numbers must match. The minor version number of the schema can be 1 less than the minor version number of the server.

Start, Stop, or Display Server Information in UNIX or Linux

Use the script dasadmin to start, stop, and display information for an Authentication Server that is running on a host in the UNIX or Linux operating environments.

The dasadmin script accepts the following commands:

start - starts the Authentication Server.

stop - stops the Authentication Server.

status - displays the operational status (running, not running) of the Authentication Server.

help - displays usage information for the dfsadmin script.

version - displays version information for the Authentication Server and records the same information in the Authentication Server log file.

To start an Authentication Server on a host that runs UNIX or Linux, enter the following command:

```
install-path/bin/dasadmin start
```

To stop an Authentication Server, use:

```
install-path/bin/dasadmin stop
```

Note that version number differences between the Authentication Server database schema and the Authentication Server itself can terminate the start process. Major version numbers must match. The minor version number of the schema can be 1 less than the minor version number of the server.

Connect to an Authentication Server

You connect to an Authentication Server to view and edit logins, users, groups, domains, and shared logins.

Before you can connect, you must first create a server definition.

To connect to an Authentication Server:

1. Expand the Authentication Servers riser.
2. Right-click the server name.
3. Select **Open**.
4. In the Login dialog box, supply a user ID, domain, and password. Use either a login that has been associated with a user definition, or use a login that is valid on the Authentication Server host.

If you log in without a user definition, but with a host login, you can see all users, groups, and domains in that server's authentication data store.

If your login is associated with a user definition on that server, you can edit your logins in that user definition.

After you connect you will receive new risers: Domain, Users, Groups, and Shared Logins.

To disconnect from a server, click red **X** in the server tab in the top left corner.

Select a Default Authentication Server

When you select a default Authentication Server, you will be prompted to log in when you start Data Management Studio.

Before you can select a default server, you must first create a server definition.

To select a default server, right click the server definition in the Administration riser and select **Set as Default**.

You can also:

1. Click **Authentication Servers**
2. Select a server in the information pane.
3. Click the star symbol, which is entitled **Set the server as the default**.

Backup or Restore the Authentication Server

Overview

Use this section to backup or restore your users, groups, domains, logins, and shared logins, either in the default transactional database or on Oracle. Also use this section to backup the executable files of your Authentication Server.

Backup Server Files

To back up your Authentication Server executable files, make copies of the following directories and subdirectories.

On Windows, copy the following directories or the equivalent directories at your site:

```
C:\Documents and Settings\admin-id\Application Data\DataFlux\AuthServer
```

Or, on Windows 7:

```
C:\Users\admin-id\Application Data\DataFlux\AuthServer
```

And:

```
C:\Program Files\DataFlux\AuthServer
```

On UNIX or Linux, copy the Authentication Server's home directory, and all of its contents.

Backup or Restore a Transactional Database

The Authentication Server uses a transactional database by default to store users, groups, domains, logins, and shared logins. The database is implemented in a file.

The file is required to be stored locally, on the host of the Authentication Server. The name and location of the database file are specified in the configuration file as_serv_aspsql.xml. The location is specifed for the entity ASPSQL_TRANDBF, as follows:

```
ENTITY ASPSQL_TRANDBF "install-path\var\asdb.tdb">
```

To backup the transactional database file, enter the following command :

```
install-path\bin\dasutil backup full-path-to-transdb-backup-file
```

In the UNIX or Linux operation environment, the command and the install path are the same.

To restore the transactional database, enter:

```
install-path\bin\dasutil restore full-path-to-transdb-backup-file
```

The full path points to a backup file, not to the file that you are backing up, as shown in the following Windows examples:

```
dasutil backup \\myBackupHost\myBackupPath\120831asdb.tdb

dasutil restore \\myBackupHost\myBackupPath\120831asdb.tdb
```

Note: If you backup your transactional database with dasutil, then you are required to restore your database with dasutil.

Backup or Restore an Oracle Database

If you use Oracle to store your users, groups, domains, logins, and shared logins, then locate the Oracle schema that contains the Authentication Server tables. The schema is identified in the configuration file as_serv_aspsql.xml.

Copy the following tables into a new schema, or copy these tables into the same schema using a different name:

DOMAINS

GROUPS

SUBJECTS

SUBJECT_GROUPS

GROUP_GROUPS

PRINCIPALS

PRINCIPAL_MAPS

GROUP_MAP_MGRS

GROUP_MAP_USERS

SUBJECT_MAP_MGRS

SUBJECT_MAP_USERS

VERSION

SENTINEL

Note: These are the default table names. If you changed the names of the tables in the as_serv_aspsql.xml configuration file, use the customized names.

Administer Log Files

Overview

By default, the DataFlux Authentication Server records a selected set of events in a file that is stored on the local host. On Windows, the default path to the log file is:

```
install-path\var\log\as_%d_%S {pid}.log
```

The `d` value becomes the date, the `S` value becomes the server hostname, and *pid* represents the process ID.

In the UNIX and Linux operating environments, the default path to the log file is:

```
install-path/var/log/das_yyyy-mm-dd_process-id.log
```

Example:

```
install-path/var/log/das_2013-05-31_24426.log
```

Log events and thresholds are specified in the log configuration file as_log.xml. In the Windows operating environment, the default location of that file is:

```
install-path\etc\as_log.xml
```

About Appenders and Loggers

As shown in the log configuration file as_log.xml, the default log configuration consists of one appender and nine loggers. The appenders specify a log output destination. The loggers specify log event types and thresholds.

The RollingFileAppender is configured by default to generate a new log file each day and for each invocation of the Authentication Server.

Loggers define the log events that are monitored. Loggers also define a threshold level for each monitored log event. The threshold levels determine the amount of information that is recorded in the log for each event.

The following list of threshold levels is ordered from least-information at the top, to most-information at the bottom:

OFF
FATAL
ERROR
WARN
INFO
DEBUG
TRACE
ALL

The default loggers and thresholds are defined in the following table.

Default Loggers and Thresholds

Logger	Description	Threshold
Cradle	records cradle messages	Info
DataFlux.licensing	records license checks	Warn
Admin	records administrative activity	Info
App	records messages from the Studio client	Info
Audit	records file reads, writes, and deletes	Info
IOM	records messages from other servers	Info
root	threshold applies to all unspecified log events	Error
App.TableServices. SQLDriver	INACTIVE, records database transactions, for use with tech support only	Trace
App.Statement.Statement . ExecDirect	INACTIVE, records statements input from Studio	Trace
App.Statement.Statement . Prepare	INACTIVE, records statements output to Studio	Trace

Note: The three inactive loggers should be enabled only when you are directed to do so by DataFlux technical support.

Change Log Events and Thresholds

The default log configuration captures most of the events that you will need to diagnose server problems. You can change the default log configuration at any time

by changing log events and threshold levels. Log changes are generally used to help diagnose errors.

Note that if you opt to receive additional log messages, by using a threshold level of DEBUG, TRACE, or ALL, you may experience a reduction in server performance. In general, it is recommended that you not select a threshold below INFO when the server is operational in a production environment.

Also note that the logging facility can be adapted to use other appenders and loggers. Please contact DataFlux Technical Support for further information.

To disable a logger or change a logger's threshold level, follow these steps:

1. Open in a text editor the log configuration file as_log.xml.
2. To prevent any further collection of log events for a given logger, enclose the logger in comment tags, as in:

```
<!-- Administration message logger -->
 <!--<logger name="Admin"> -->
  <!--<level value="Info"/> -->
<!--</logger> -->
```

3. To change the threshold of a logger, replace the existing level value with OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE, or ALL, as in:

```
<!-- Administration message logger -->
 <logger name="Admin">
  <!-- DEFAULT <level value="Info"/> -->
<level value"Warn"/>
</logger>
```

4. Save and close the log file.
5. Restart the Authentication Server.

Initial Log File

Here is the XML log file that is initially installed with the Authentication Server:

```
<?xml version="1.0" encoding="UTF-8"?>
<logging:configuration xmlns:logging="http://www.sas.com/xml/logging/1.0/">

     <!-- Rolling log file with default rollover of midnight -->
     <appender class="RollingFileAppender" name="TimeBasedRollingFile">
          <param name="Append" value="true"/>
          <param name="ImmediateFlush" value="true"/>
          <rollingPolicy class="TimeBasedRollingPolicy">               <param
name="fileNamePattern"
                    value="$home\var\log\as_%d_%S{pid}.log"/>
          </rollingPolicy>
          <layout>
               <param name="HeaderPattern" value="Host: '%S{hostname}',
                    OS: '%S{os_family}',
                    Release: '%S{os_release}',
                    SAS Version: '%S{sup_ver_long2}',
```

```
                    Command: '%S{startup_cmd}'"/>
                <param name="ConversionPattern"
                    value="%d %-5p [%t] %c %X{Client.ID}:%u - %m"/>
            </layout>
        </appender>

        <!-- Cradle message logger -->
        <logger name="Cradle">
            <level value="Info"/>
        </logger>

        <!-- DataFlux licensing message logger -->
        <logger name="DataFlux.Licensing">
            <level value="Warn"/>
        </logger>

        <!-- Administration message logger -->
        <logger name="Admin">
            <level value="Info"/>
        </logger>

        <!-- Application message logger -->
        <logger name="App">
            <level value="Info"/>
        </logger>

        <!-- Audit message logger -->
        <logger name="Audit">
            <level value="Info"/>
        </logger>

        <!-- IOM protocol message logger -->
        <logger name="IOM">
            <level value="Info"/>
        </logger>

        <root>
            <level value="Error"/>
            <appender-ref ref="TimeBasedRollingFile"/>
        </root>

        <!-- Perf -->
        <logger name="Perf.ARM.SQLServices">
            <level value="Warn"/>
        </logger>

</logging:configuration>
```

Administering Users, Groups, Domains, Logins, and Shared Logins

Overview

By default, each Authentication Server maintains a database of users, groups, domains, logins, and shared logins, as defined in About Users, Groups, Domains, Logins, and Shared Logins. The database is used for authentication and authorization. Authentication Servers use logins and shared logins to authenticate connection requests for SAS Federation Servers, Data Management Servers, and relational databases. Authorization is implemented in the Data Management Servers and Federation Servers. These servers authorize access to data collections and jobs based on group membership information that is provided by an Authentication Server.

The Authentication Server provides two interfaces that enable you to create and maintain users, groups, domains, logins, and shared logins: the Administration riser in SAS Data Management Studio, and the ASBATCH utility.

The Administration riser provides administrators with read/write access to Authentication Server data. Passwords cannot be displayed, and logins cannot be accessed by default. Non-administrative users can use the Administration riser to display authorized information and add logins to their user definitions.

The ASBATCH utility provides a scripting interface that enables you to update the Authentication Server database during off-peak hours. In the script file that drives the updates, you can use filters to update multiple rows with a single command.

Use the Administration Riser

Follow these steps to use the Administration riser to create and maintain users, groups, domains, logins, and shared logins on an Authentication Server:

1. Open Data Management Studio.
2. Click the **Administration** riser on the lower left.
3. Expand **Authentication Server** in the tree, right-click an Authentication Server, and select **Open**.

4. Enter a user ID and password.

   To create or edit users, groups, domains, logins, or shared logins, you need a login that is listed as an administrator for the current Authentication Server. Administrators cannot add or delete logins other than their own.

   If you connect with a login that is not administrative, you can add logins to your user, view memberships in groups, and view the consumers of shared logins.

5. Click a riser to display users, domains, groups, and shared logins. To add or view logins, click the **User** riser.

6. Click icons or right-click to create or maintain users, groups, domains, logins, or shared logins.

   Note: When you create users, groups, domains, logins, and shared logins, the Authentication Server may accept special characters that are not accepted in your operating environment. Be sure to follow the naming conventions of your operating environment.

Access User Logins

After an administrator creates a new user definition, the individual who is identified in that user definition adds a password and can also add more logins. Normally, and by default, administrators cannot add, delete, or modify logins. If such access is necessary on a temporary basis, you can enable access, make changes, and then disable access to logins. Similar capabilities are also available in the ASBATCH utility.

Follow these steps to access user logins in the Administration riser:

1. Stop the Authentication Server and edit the configuration file as_serv_aspsql.xml.

2. Add the following option to the configuration file:

   ```
   <Option name="AdminLoginManagementPolicy">ADD REMOVE
   UPDATE</AdminLoginManagementPolicy>
   ```

   You can specify any combination of ADD, REMOVE, or MODIFY. When you modify a login, you change the password.

3. Save and close the configuration file.

4. Start the Authentication Server and change logins.

5. Stop the Authentication Server and edit the configuration file.

6. To preserve security, delete the option AdminLoginManagementPolicy, and then save and close the file.

7. Restart the Authentication Server.

To query the status of the login management policy, use the following read-only Boolean category items:

- Server.LoginManagementPolicy.Add
- Server.LoginManagementPolicy.Remove
- Server.LoginManagementPolicy.Update

Update in Batch with the ASBATCH Utility

ASBATCH Overview

The ASBATCH utility enables you to update the Authentication Server database using a script and an XML or CMV file. You can execute the script at times of minimal server access.

The ASBATCH utility is installed as a selectable component of SAS Federation Server Client software package.

When you invoke asbatch.exe, command-line options point to an XML or CSV file. In the XML or CSV file, operators and options cause ASBATCH to add, modify, or delete users, groups, domains, and logins.

> **Note:** ASBATCH does not add, modify, or delete shared logins. Use the Administration riser in SAS Data Management Studio for that purpose.

> **Note:** CSV files do not add, modify, or delete groups. Use an XML file for that purpose.

The ASBATCH utility generates an audit file that lists all changes made, and also generates an undo file that enables you to remove newly added users, groups, domains and logins.

You can create a log configuration file that will cause ASBATCH to generate log entries at a specified level of detail.

To execute ASBATCH, you need to log in with an account that is specified separately from the accounts of Authentication Server administrators. You also need to set an option that specifies explicit permission to add, delete, and/or update the data.

Use ASBATCH

After you ASBATCH using the SAS Federation Server Client package, follow these steps to use the ASBATCH utility:

1. If you prefer to not include an administrative login in the command that executes asbatch.exe, then set the following environment variables in the operating environment:

   set ASBATCH_UID=*asbatch-username*
   set ASBATCH_PWD=*asbatch-password*

   Note: The ASBATCH password will be available in plaintext. To minimize risk, do not set these variables as defaults, and remove the variables or quit your session after you complete your update.

2. Stop the Authentication Server and edit the configuration file *install-path*\etc\as_serv_aspsql.xml.

3. Open the Authentication Server configuration file as_serv_aspsql.xml. Add the following option to the bottom of the file to permit access to logins. Specify the type of access you require.

   <Option name="AdminLoginManagementPolicy">ADD REMOVE UPDATE
   </Option>

   You can specify any combination of ADD, REMOVE, and UPDATE. When you update a login, you replace the password.

4. Save and close the configuration file.

5. Open a text editor to create your ASBATCH script file. The script file is executed when you run ASBATCH. The script file specifies operators and options for the ASBATCH utility. The name and location of the script file is specified in the command that invokes asbatch.exe. The script file uses either XML or CSV format. To create your script file, see Sample XML and CSV Files for ASBATCH and ASBATCH Operators and Options for XML and CSV Files.

6. Save and close the script file.

7. Optionally create a configuration file that will cause ASBATCH to generate a log file. See Generate a Log File for ASBATCH.

8. Start ASBATCH by specifying options for asbatch.cmd in Windows, or for asbatch in UNIX or Linux. Specify the -help option as follows to display a list of available options:

   ```
   install-path\bin> asbatch.cmd -help

   csh> asbatch -help
   ```

   Note: Be sure to specify filenames for the audit file, undo file, and log file.

   To learn more, see ASBATCH Command-Line Options.

9. Review the contents of the audit file and optional log file.
10. To remove the additions that were made by the most recent run of ASBATCH, use the undo file.
11. To preserve security and prevent impersonation, stop the Authentication Server, open the configuration file as_serv_aspsql.xml, and delete the option AdminLoginManagementPolicy. Save and close the configuration file.
12. In the operating environment, either quit your current session or remove the environment variables ASBATCH_UID and ASBATCH_PWD.
13. Start the Authentication Server and contact Data Management Studio users to permit connections to the Authentication Server.

Sample XML and CSV Files for ASBATCH

ASBATCH updates the Authentication Server's database based on the operators and options that are specified in an XML or CSV file.

The XML or CSV file is opened and read when you invoke asbatch.exe.

Sample XML File

In the following example file in XML format, the first line is required, as are the opening and closing ASBatch tags.

```
<?xml version="1.0" encoding="utf-8"?>
<ASBatch major="major-release-num" minor="minor-num" delta="delta-num">
   <Add>
      <User name="user-name" login="principal-name" domain="domain-name"
           desc="description" />
      <Domain name="ASTEST" desc="Used for testing."
           isLogin="false"
           isCase="false"
           isUPN="false"/>
       <Group name="group-name" owner="group-owner" desc="Group description" />
   </Add>
   <Update>
      <Domain name="domain-name" desc="New domain name" />
      <Group name="group-name" newname="new-group-name" />
      <Group name="group-name" owner="new-group-owner" />
      <Group name="group-name" desc="New description for new group" />
      <User name="old-name" newname="new-name" />
        <User name="name" desc="New description for new name" />
      <User filter="description='user description'" desc="New description" />
      <User name="name" isEnabled="false" />
      <User filter="isEnabled='true'" isEnabled="false" />
       <User name="name">
         <Add>
            <Login name="principal-name" domain="domain-
name" password="password" />
         </Add>
         <Remove>
            <Login domain="domain-name" />
         </Remove>
         <Update>
```

```
            <Login domain="domain-name" password="new-password" />
         </Update>
      </User>
      <Group name="group-name">
         <Add>
            <User name="user-name"/>
            <Group name="group-name"/>
         </Add>
         <Remove>
            <User name="user-name"/>
            <Group name="user-group"/>
         </Remove>
      </Group>
   </Update>
   <Remove>
      <User name="name"/>
      <User filter="description='User description'"/>
      <Domain name="dname"/>
      <Domain name="dname" isCascade="TRUE"/>
      <Domain filter="description='Domain description'" />
      <Group name="group-name"/>
      <Group filter="description='Group description'" />
   </Remove>
</ASBatch>
```

Sample CSV File

The following example of a comma-separated file shows the required descriptor on the first line.

```
TITLE:,ASBATCH,major=version-major,minor=version-minor,delta=version-delta
ADD_USER:,name=uname,domain=dname,login=lname,desc=User description
ADD_DOMAIN:,name=dname,desc=Domain description,isLogin=TRUE,isCase=FALSE,isUPN=NO
UPDATE_USER:,name=uname,desc=New description
UPDATE_USER:,filter=description='User description',desc=New description
UPDATE_USER:,name=uname,IsEnabled=FALSE
UPDATE_USER:,filter=description='User description',IsEnabled=FALSE
UPDATE_DOMAIN:,name=dname,desc=New description
UPDATE_USER_ADD_LOGIN:,name=uname,domain=dname,password=password
UPDATE_USER_REMOVE_LOGIN:,domain=dname
UPDATE_USER_UPDATE_LOGIN:,name=uname,domain=dname,password=new-password
REMOVE_USER:,name=uname
REMOVE_USER:,filter=description='User description'
REMOVE_DOMAIN:,name=dname
REMOVE_DOMAIN:,filter=description='Domain description'
REMOVE_DOMAIN:,name=dname,isCascade=TRUE
```

ASBATCH Operators and Options for XML and CSV Files

Using an XML file, you can add, update, and remove users, groups, and domains. Using a CSV file, you can add, update, and remove users and domains. Use the Administration riser to maintain shared logins.

For information on valid values for options that take boolean values, see Valid Values for Boolean Options.

Operator - Description	Options	Option Description and Syntax
Add User	name	Name of new user. XML: <Add> <User name="*user-name*" login="*user-id*" domain="*domain-name*" desc="*user-description*" /> </Add> CSV: ADD_USER:,name=*user-name*,domain=*domain-name*,login=*user-id*,desc=*user-description*
	login	User's login for authentication.
	domain	Name of domain that authenticates the login
	desc	User description, 0-256 bytes
	isEnabled	Boolean value of 1 or true enables authentication for the user.
Add Domain	name	Domain name. XML: <Add> <Domain name="*domain-name*" desc="domain-description" isLogin="*boolean*" isCase="*boolean*" isUPN="*boolean*" /> CSV: ADD_DOMAIN:,name=*domain-name*,desc=*domain-description*,isLogin=*boolean*,isCase=*boolean*,isUPN=*boolean*
	desc	Domain description, 0-256 bytes
	isLogin	Boolean value 1 or true indicates the default domain, user enters login only.
	isCase	Boolean value 1 or true indicates a case-sensitive domain name. A value of 0 or false indicates an all-caps domain name.
	isUPN	Boolean value 1 or true indicates up-level domain (login@domain). A value of 0 or false indicates a down-level domain (domain\login).
Add Group - add a new group, in XML only.	name	Group name. XML: <Add> <Group name="*group-name*" owner="*owner-name*" desc="*group-description*" />

Operator - Description	Options	Option Description and Syntax
		</Add>
	owner	Existing user name of group owner.
	desc	Group description, 0-256 bytes.
Remove User, Remove Domain, Remove Group - remove existing entry. Remove group is available in XML only.	name	Name of user, domain, or group to be removed. Use name or filter but not both. XML: <Remove> <User|Domain|Group filter="*option-name='option-value*'"/> </Remove> CSV: REMOVE_GROUP:,filter=*match-option-name='match-option-value'*,isCascade=True
	filter	Removes multiple users, domains, or groups, based on matching option values.
	isCascade	For Remove Groups, 1 or True indicates that any users who belong to the removed group will have that group membership removed. When isCascade=False, the group remove operation fails if any users are still members of that group.
Update User - change descriptions and/or enablement for existing user(s).	name	Name of user. XML 1 of 3 - for one user, change description and/or enablement): <Update> <User name="*user-name*" isEnabled="*boolean-auth-enable-or-disable*" desc="*new-description*" /> </Update> XML 2 of 3 - change description and/or enablement for all users with matching strings in the description option: <Update> <User filter="description='*desc-match-string*'" desc=”*new-description*" isEnabled=*boolean* /> </Update> XML 3 of 3 (change descriptions for all matching enablements): <Update> <User filter="isEnabled='*boolean*'" desc=”*new-description*" /> </Update>

Operator - Description	Options	Option Description and Syntax
		CSV 1 of 3: UPDATE_USER:,name=*user-name*,isEnabled=*boolean*,desc=*new-desc* CSV 2 of 3: UPDATE_USER:,filter=description='*desc-match-string*',desc=*new-description* CSV 3 of 3: UPDATE_USER:,filter=isEnabled=*boolean*,desc=*new-value*
	desc	Description of user, 0-256 bytes.
	filter	For matching user options, make the specified change.
	isEnabled	Boolean value of 1 or true enables authentication for the user.
Update User Add Login - add login to existing user.	login	User ID for authentication. XML: <Update> <User name=*existing-user-name*> <Add> <login=*new-login* domain=*new-domain* password=*new-pwd*/> </User> </Update> CSV: UPDATE_USER_ADD_LOGIN:,name=*existing-user-name*,login=*new-login*,domain=*new-domain*,password=*new-pwd*
	domain	Domain used for authentication.
	password	Password that accompanies the user ID.
Update User Update Login - change existing login, without changing the user ID.	name	Name of an existing login. XML: <Update> <User name="username" domain="new-domain" password="new-password"/> </Update> CSV: UPDATE_USER_UPDATE_LOGIN:,name=*existing-user-name*,domain=*new-domain*,password=*new-pwd*
	domain	New domain value for existing login.
	password	New password for existing login.

Operator - Description	Options	Option Description and Syntax
Update User Remove Login - remove a login from an existing user.	domain	Domain of login to be removed. XML: <Update> <User name=*existing-user-name*> <Remove> <Login domain=*domain-name*/> </Remove> </User> </Update> CSV: UPDATE_USER_REMOVE_LOGIN:name=*existing-user-name*,domain=*domain*
Update Domain - change the description of a domain.	name	Name of domain to be updated. XML: <Update> <Domain name=*existing-domain-name* desc=*new-domain-description* /> </Update> CSV: UPDATE_DOMAIN:,name=*existing-domain-name*, desc=*new-domain-description*
	desc	New description for domain, 0-256 bytes.
Update Group - change the name, description, or owner of an existing group, in XML only.	name	Name of group to be updated. XML: <Update> <Group name="existing-group-name" desc="*new-description*" newname="*new-group-name*" owner="*new-group-owner*" </Group> <Update />
	desc	New group description, 0-256 bytes.
	newname	New group name
	owner	New group owner.
Update Group Add or Remove - add or remove a user or group from an existing	name	Name of group to be updated, name of user or group to be added or removed. XML: <Update> <Group name="group-name"> <Add | Remove> <User name="user-name" />

Operator - Description	Options	Option Description and Syntax
group, in XML only.		<Group name="group-name" /> </Add | Remove> </Group> </Update>
	user	Add or remove a user.
	group	Add or remove a group.

Valid Values for Boolean Options in ASBATCH

ASBATCH accepts the following values for boolean options:

Valid values for "true:" TRUE, true, YES, yes, 1, T, t, Y, y

Valid values for "false:" FALSE, false, NO, no, 0, F, f, N, n

ASBATCH Command-Line Options

Use the following options to execute asbatch.exe:

-a |--audit *audit-file-path*

specifies the name and path of the file that ASBATCH generates to record all database changes.

-h | --help

Displays a list of available command line options and exits.

-i | --input *path-to-XML-or-CMV-file*

Specifies the name and location of the XML or CSV file that contains ASBATCH operators and options.

-lc | --log-config-loc *path-to-asbatch-log-config-file*

Specifies the name and location of the file that configures logging for ASBATCH.

-p | --port *auth-server-port-number*

Specifies the port number used by the target Authentication Server.

-pw | password *plaintext-password*

Specifies the password of the user definition that will be used along with the user and domain options to authenticate ASBATCH. Specify this value only if

you choose not to set the environment variable ASBATCH_PWD, as described in Use ASBATCH. This password is displayed in plaintext.

-r | --uri *connection-string*

Specifies an IOM connection string that is used only when the port option is not specified.

-s | --server *server-identifier*

Specifies the name of the host of the target Authentication Server. Valid values are a network name, an IP address, or `localhost`.

-t | --type *type-of-changes-file*

Specifies the format of the input file that specifies database changes. Valid values are `XML` or `CSV`. `XML` is the default.

- us | --user *user-name*

Specifies the name of the user defintion that will be use to authenticate ASBATCH, along with the values of the password and domain options. Specify this value only if you choose not to set the environment variable ASBATCH_UID, as described in Use ASBATCH.

-v | --version

Displays ASBATCH version information and exits.

Sample ASBATCH Command

```
install-path\bin\asbatch.exe
   --input c:\ProgramFiles\DataFlux\AuthServer\server1\etc\XMLfile1.xml
   --audit asboutfile.xml
   --port 21030
   --server localhost
   --user LOCAL\ADMIN -password ADMIN_PASS
   --type XML
   --logconfigloc install-path\etc\asbatch_log4sas.xml
```

Use the ASBATCH Audit and Undo Files

When you run ASBATCH, the utility generates an audit file and an undo file. The audit file records all of the changes that ASBATCH makes to the Authentication Server database. The undo file enables you to remove new entries that were added by the last run of ASBATCH. ASBATCH will not remove any new entries that were modified after the last run of ASBATCH.

The names and paths of the audit and undo files are specified by the `audit` option of the command that executes asbatch.exe. The name of the audit file determines the name of the undo file, which is of the form: *audit-file*`U.xml`.

To remove new additions with the undo file, edit the `input` option in the previous ASBATCH command. Replace the name and path of the input XML or CSV file with

the name and path of the undo file, and then execute the command. ASBATCH generates a new audit to confirm the removals.

The undo file does not replace database entries that were modified or removed. To see a list of modifications and removals, refer to the audit file.

Use the ASBATCH Log File

Default Log Configuration

As you may recall, ASBATCH is installed on a client host, as a selectable component of the SAS Federation Server Client software package. When you install ASBATCH, you receive a log file and a log configuration file. ASBATCH uses those files to collect log entries by default.

The default log file is named asbatch.log. The default path for that file is one of the following:

install-path\var\log\asbatch.log

The default path and name of the ASBATCH log configuration file is as follows:

install-path/etc/asbatch_log.xml

Changing the Default Logging Behavior

To change the default logging behavior of ASBATCH, you can edit your existing log configuration file or you can create a new log configuration file. The new log configuration file can point to a new log file in a different location on the local host. For information on setting logging levels, see About Appenders and Loggers.

Initial ASBATCH Log Configuration File

The following XML content is delivered in asbatch_log.xml when you install ASBATCH:

```
<?xml version="1.0"?>

<log4sas:configuration xmlns:log4sas="http://www.sas.com/rnd/Log4SAS/">

   <appender name="LOG" class="FileAppender">
      <param name="File" value="/home/eredwa/asbatch/asbatch.log"/>
      <param name="ImmediateFlush" value="true"/>
      <param name="Append" value="false"/>
      <layout>
           <param name="ConversionPattern" value="%d %-5p [%t] %u %c - %m
(%F@%L)"/>
      </layout>
   </appender>

 <root>
      <level value="ERROR"/>
      <appender-ref ref="LOG"/>
 </root>
```

```
</log4sas:configuration>
```

About Users, Groups, Domains, Logins, and Shared Logins

Overview

Users, groups, domains, logins, and shared logins are records in a database that is maintained by an Authentication Server. By default, the Authentication Server uses a transactional database that resides on the host of the Authentication Server. Each Authentication Server maintains a distinct database, and records are not shared between databases. As an alternative to the default configuration, you can configure your Authentication Server to maintain users, groups, domains, logins, and shared logins in Oracle. Using Oracle, multiple Authentication Servers can share a single set of system tables, with available TCP optimizations between servers. For further information about using Oracle to store authentication objects, see Configure Oracle to Store Users, Groups, Domains, Logins, and Shared Logins.

Database records are created, displayed, edited, and deleted using the Administration riser in DataFlux Data Management Studio. You can also update the database in batch using an XML or CMV file and the ASBATCH utility.

To connect to an Authentication Server, open the Administration riser in Data Management Studio, expand Authentication Servers, right-click a server, and select **Open**.

Your access to users, groups, domains, logins, and shared logins depends on your role. Passwords cannot be displayed regardless of your role. Administrators can add, edit, and delete all records other than logins, and can update the passwords of shared logins. Owners and managers of groups and shared logins can add and delete members. Users can see their logins. Everyone can see all users and groups.

DataFlux clients and servers query the Authentication Server database for user and group membership information. User and group information is used by clients and servers to manage access to data, jobs, and services.

During operation, database updates are immediately made available to DataFlux clients and servers. Changes and deletions can result in changes to existing connections between DataFlux clients and servers.

Note that when you create or rename records, the Authentication Server may accept special characters that are not accepted in the operating environment. Be sure to follow the naming conventions of your operating environment.

Domains

A domain is named collection of logins that share an authentication provider. The Authentication Server defines domains so that Data Management Studio users can connect to Data Management Servers, SAS Federation Servers, Web Studio Servers, and database servers in those domains.

As an example of how domains are implemented, assume that you have a SAS Federation Server that runs on a Windows host in a domain named CHICAGO. To enable a Data Management Studio user to connect to that server, you would follow these general steps:

1. You, the administrator, connect to an Authentication Server to create the CHICAGO domain, using the Domains riser. Use the same format that is used on Windows, such as CHICAGO/myLogin or us.ourcorp.chicago.com.
2. Identify the authentication mechanism of the CHICAGO domain in the Authentication Server's configuration file as_serv_aspsql.xml, as an added value for the option AuthProviderDomain.
3. The Studio user adds a CHICAGO login to his or her user definition.

At this point, the user can request a connection to the SAS Federation Server, authenticate in the CHICAGO domain, and access data based on his or her user definition and group memberships.

When users add logins to a new domain, they can create no more than one login per domain for their one user definition.

If a Studio user logs in without a domain, a default domain is supplied. The default comes from the PrimaryProviderDomain option. If that option has no value, then the Authentication Server uses host authentication.

Domains have properties that determine how they will be submitted for authentication. Domains can be defined as user name only (userid), user login name (userid@domain), or down-level login name (domain\userid). Additionally, domains can be case-sensitive (mixed-case), or case-insensitive (domain entries from users are converted to uppercase before authentication).

Logins

Logins consist of a combination of a user ID and a password. The Authentication Server works with three types of logins:

Inbound logins - are sent from Data Management Studio to the Authentication Server to verify the identity of the user when the user starts the Studio application or when the user connects to the Authentication Server. Inbound logins are also used to establish connections to DataFlux servers. When a Studio user requests a

connection to a DataFlux server, the Authentication Server forwards that user's inbound login to the DataFlux server's domain for authentication. If the user authenticates successfully, the Authentication Server notifies the DataFlux server, and the DataFlux server accepts the connection.

Outbound logins - are submitted to database servers to validate the identity of the users whom request connections to those databases. Outbound logins are defined for each shared login. A shared login enables consumers (users or groups) to access the database using a shared database account. When a user requests a connection to a database server, the Authentication Server confirms that the user is a consumer, and sends the login to the client. The client sends the login to the database to establish the connection. The outbound login is not displayed to the user.

Oracle login - if you choose to store your users, groups, domains, logins, and shared logins in Oracle, the Authentication Server uses an outbound Oracle login to connect to that database, as described in Configure Oracle to Store Users, Groups, Domains, Logins, and Shared Logins.

Administrators define one initial inbound login when they create a new user definition. The user can then add unique logins to his or her user definition. A user definition can have no more than one login for each domain.

Administrators cannot display passwords and they cannot edit another user's logins. However, administrators can edit the outbound logins of shared logins, including the passwords.

Logins can be shared by multiple Authentication Servers if those servers share a single set of system tables in Oracle. Otherwise, each Authentication Server maintains a separate set of logins.

Users

User definitions, or simply "users", are database entries that associate a platform name with one or more logins in the operating environment. Each login consists of a unique combination of a domain, user ID, and password.

A user can be added as a member of a group or added as a consumer of a shared login.

User passwords are not displayed. By default, administrators cannot add or delete logins, or change passwords in a user definitions.

Groups

Groups are categorized collections of users. Groups are ofted defined according to work role, such as Payroll, Accounting, and Human Resources. Groups are used to structure authorization to the jobs and data that are stored on SAS Federation Servers and Data Management Servers. The servers query their Authentication Servers as needed to determine group membership.

Each group has an owner. The owner of a group owner can edit the group definition, add and delete members, and assign a new owner. The owner is defined from the existing set of user definitions. A group is required to have an owner at all times.

Administrators can add and delete groups, add and delete members, and reassign group owners.

Groups can be members of other groups.

Groups can be designated as consumers of shared logins.

Groups can be designated as managers of shared logins.

Two top-level groups, PUBLIC and USERS, are continuously maintained by the Authentication Server. The PUBLIC group includes all Data Management Studio users who successfully authenticate in the Authentication Server's host environment. PUBLIC users are not required to have a registered user definition on the Authentication Server. This all-inclusive group receives minimal access to data.

The USERS group is a member of the PUBLIC group. The USERS group consists of all PUBLIC users who do have user definitions on the Authentication Server. The USERS group inherits the minimal permissions of the PUBLIC group. Additionally, members of the USERS group can edit their user definitions and group memberships.

By default, members of the PUBLIC group have read access to the group membership information that is displayed in Data Management Studio. You can remove this read access by changing the value of the option PublicUserGroupManagementPolicy.

Shared Logins

Shared logins are collections of users and groups that use outbound logins to connect to database servers. When a Studio user requests a connection to a database, if that user is a *consumer* of a shared login for that database, then the Authentication Server sends the outbound login (database credentials) to the Studio client, and the client connects to the database. The Studio user sees no information about the outbound login.

Consumers of shared logins do not need individual accounts on the respective database servers.

The passwords for outbound logins cannot be displayed.

Administrators can add and delete shared logins and add and delete consumers of shared logins. Administrators cannot delete or replace the outbound login.

Each shared login has a designated owner. The designated owner can be a user or a group. The owner has full access to the shared login, including the ability to read and replace the outbound user ID and password. The owner can also change his login and reassign his or her ownership to another user.

Shared logins have designated managers as well as owners. Managers can be users or groups. Managers can add and delete consumers and read the outbound login. Users that are designated as managers can add and delete memberships in the shared login. Manager logins are configured on the SAS Federation Server to read the outbound login without revealing that login to the connecting client.

Each shared login has a required key value. You can assign the same key to multiple shared logins. Keys are used by the Shared Login Managers on SAS Federation Servers. The Shared Login Manager uses a login and key value to gain access to one or more shared logins. The Shared Login Manager uses the accessible shared logins to make connections to relational databases.

Appendix: Configuration File Reference

The options in the main Authentication Server configuration file, as_serv_aspsql.xml, are defined as follows.

In the Windows operating environment, the default location of the file is:

```
install-path\etc\as_serv_aspsql.xml
```

Note that when you install an Authentication Server, the default configuration file does not contain default entries for all of the following options. To specify a value for an option that does not have a default entry, simply add that option as a new entry.

For information about other configuration files, see About the Authentication Server Configuration Files.

Valid Values for Boolean Options

Options such as AddUser and AutoAddDefaultDomain accept Boolean values. The valid values for all Boolean options in this configuration file consist of: `TRUE`, `FALSE`, `YES`, `NO`, `Y`, and `N`. Other Boolean values, such as `y`, `n`, `t`, `f`, `1`, or `0`, are inapplicable.

About ENTITY Declarations

In the configuration file, ENTITY declarations are used to define the transactional database that the Authentication Server uses to manage users, groups, domains, logins, and shared logins. DataFlux recommends that you retain the default values of these entities.

If you decide to configure your database on Oracle, then you should comment-out the entity declarations for the transactional database, as directed in Configure Oracle to Store Users, Groups, Domains, Logins, and Shared Logins.

If you use the default transactional database, then you are required to locate the database on the Authentication Server host and specify a full, non-relative, path as the value of the entity ASPSQL_TRANDBF.

Options in the Configuration File

AdminLoginManagementPolicy
ASPSQLProvider
AuthProviderDomain
AutoAddUsers
CredentialsLocation
EncryptFIPS
FIREBIRD_LOG
AppendEnv
AuthenticationProvider
AutoAddDefaultDomain
Clientencryptionlevel
Dnsname
FIREBIRD
License

AdminLoginManagementPolicy

```
<Option name="AdminLoginManagementPolicy">keywords</Option>
```

All administrators are authorized to add users and create one login per user. By default, only users can change their logins. The AdminLoginManagementPolicy option allows administrators to add logins, delete logins, and update logins. Specify any of the following keywords in any order:

ADD – administrators can add user logins
REMOVE – administrators can remove user logins
UPDATE – administrators can reset user passwords.

Example:

```
<Option name="AdminLoginManagementPolicy">ADD REMOVE UPDATE</Option>
```

You can specify any combination of ADD, REMOVE, and/or UPDATE.

None of these values are specified by default.

Specifying this option poses a security risk. As such, you should specify this option only when you need to make a specific change. After the change is made, you should remove this option from the configuration file.

AppendEnv

```
<OptionSet name="AppendEnv">
     <Option name="your-variable">your-append-value</Option>
</OptionSet>
```

The AppendEnv option will find the indicated environment variable in the operating environment and append the option value to the end of the existing value. If the environment variable does not exist, then it will be created and set to the option value. The AppendEnv option will not add a delimiter of any sort between the existing and new environment variable value. If a semi-colon (;) is needed, then it must appear as the first character in the option value.

ASPSQLProvider

```
<OptionSet name="ASPSQLProvider">
     <Option name="SystemCatalog">AS</Option>
     <Option name="SystemSchema">schema-name</Option>
```

```
    <Option name="MinConnections">1</Option>
    <Option name="MaxConnections">4</Option>
    <Option name="CredentialsLocation">file-path</Option>
</OptionSet>
```

SystemCatalog - specifies the name of the catalog of the Authentication Server database.

SystemSchema - specifies the name of the schema of the Authentication Server database.

MinConnections - specifies the minimum number of connections to keep open to the Authentication Server database.

MaxConnections - specifies the maximum number of connections to keep open to the Authentication Server database. In highly concurrent environments, this value should be raised. Generally speaking, a value of 4 should meet most needs.

CredentialsLocation - specifies the location of the credentials file that is used to connect to the Authentication Server database. This option is not required when you use the default transactional database. When you use Oracle, this option can be used to store the encrypted credentials that the Authentication Server uses to connect to that database. If your site security policy forbids the storage of database credentials, you can enter credentials manually at server startup, or store the credentials in environment variables, as described in Configure Oracle to Store Users, Groups, Domains, Logins, and Shared Logins.

AuthenticationProvider

```
<Option name="AuthenticationProvider">ASPSQL</Option>
```

The AuthenticationProvider option identifies the authentication process in the Authentication Server. The named process accesses the Authentication Server database. `ASPSQL` is the only valid value.

AuthProviderDomain

```
<Option name="AuthProviderDomain">authentication-provider:domain-
name</Option>
```

Or, for two or a maximum of three domains:

```
<Option name="AuthProviderDomain">(provider1:domain1,
    provider2:domain2,
    provider3:domain3)
</Option>
```

The AuthProviderDomain option associates authentication providers with domains. You can specify a maximum of one authentication provider for each domain, and all domain values must be unique. Also, you can specify a maximum of one authentication provider of each type. Valid values for *authentication-provider* are as follows:

ADIR - specifies that authentication is provided by a Microsoft Active Directory server.

HOSTUSER - specifies that authentication is provided by the Authentication Server's host operating system.

LDAP - specifies that authentication is provided by an LDAP directory server.

For Windows, the `domain-name` value should be a case-sensitive name that is recognized on your network. If a domain name contains spaces, use quotation marks around the name. See also PrimaryProviderDomain.

Adding an authentication provider requires additional configuration. The configuration process depends on the provider type and on the operating environment of the Authentication Server host. To configure your new authentication provider, see About Authentication Providers.

AutoAddDefaultDomain

```
<Option name="AutoAddDefaultDomain"/>
```

or

```
<OptionSet name="AutoAddDefaultDomain">
     <Option name="Enabled">boolean</Option>
</OptionSet>
```

For valid values, see Valid Values for Boolean Options.

The AutoAddDefaultDomain option instructs the Authentication Server to automatically register the host domain if that domain has not already been registered, at server start time, as shown in the following examples:

```
<Option name="AutoAddDefaultDomain"/>
```

or

```
<OptionSet name="AutoAddDefaultDomain">
     <Option name="Enabled">TRUE</Option>
</OptionSet>
```

When it is enabled, the AutoAddDefaultDomain option creates a domain definition using the value of the option PrimaryProviderDomain. The domain is created only if the PrimaryProviderDomain is mapped to host authentication in the option AuthProviderDomain. For example, on Windows, the AutoAddDefaultDomain option is valid when you set the PrimaryProviderDomain and AuthProviderDomain options as shown:

```
<Option name="AutoAddDefaultDomain"/>
<Option name="AuthProviderDomain">HOSTUSER:auth-server-domain-
name</Option>
<Option name="PrimaryProviderDomain">auth-server-domain-name</Option>
```

If the domain of the Authentication Server was DATAFLUX, then the option values would be:

```
<Option name="AutoAddDefaultDomain"/>
<Option name="AuthProviderDomain">HOSTUSER:DATAFLUX</Option>
<Option name="PrimaryProviderDomain">DATAFLUX</Option>
```

On UNIX and Linux, the following option values are specified in the configuration file by default after the installation of the Authentication Server:

```
<Option name="AutoAddDefaultDomain"/>
<Option name="AuthProviderDomain">HOSTUSER:UNIXUSER</Option>
<Option name="PrimaryProviderDomain">UNIXUSER</Option>
```

The Authentication Server uses the UNIXUSER domain during authentication if the supplied login does not specify a domain.

The AutoAddDefaultDomain option is not specified by default. Add the option to the configuration file as needed. To summarize the default behavior, when a domain is not specified, the Authentication Server uses the PrimaryProviderDomain if AutoAddDefaultDomain is enabled. Otherwise, the server uses the host authentication provider.

The domain object that is created receives attributes based on the following table:

Attributes of the Default Domain

	Authentication Server OS	
Domain Attribute	**Windows**	**UNIX and Linux**
Use as part of login	Yes	No
Logins are case-sensitive	No	Yes

AutoAddUsers

```
<Option name="AutoAddUsers"/>
```

or

```
<OptionSet name="AutoAddUsers">
     <Option name="Enabled">boolean</Option>
     <Option name="DomainFilter">filter-string</Option>
     <Option name="UserIDFilter">filter-string</Option>
</OptionSet>
```

When enabled, the AutoAddUsers option specifies that Authentication Server automatically adds users as they authenticate based on the domain and user ID that they use to connect. Automatically added users receive a single login composed of the inbound user ID, in the domain specified.

> **Note:** The AutoAddUsers option does not automatically add domains.

By default the value of the Enabled option is TRUE.

The shorthand Option element form activates the auto-add feature for all users in all domains. The longer OptionSet element activates the auto-add feature for specified users in specified domains.

For example, to automatically add user definitions for any and all users who login from the BOULDERNT and OURCO domains, add the following specification to the configuration file:

```
<OptionSet name="AutoAddUsers">
     <Option name="DomainFilter">BOULDERNT OURCO</Option>
</OptionSet>
```

The values of the UserIDFilter and DomainFilter options are case-insensitive when they are compared against the logins of connecting users.

Filter option values may contain the wildcard characters, % (percent) and _ (underscore), matching zero or more characters or any one character, respectively.

Clientencryptionlevel

```
<Option name="ObjectServerParms">
     clientencryptionlevel=none | credentials | everything
</Option>
```

This parameter of the ObjectServerParms option determines how Authentication Server data is encrypted for transmission on your network. Valid values include:

none - nothing is encrypted.

credentials - login credentials are encrypted. These credentials are used to authenticate to the Authentication Server.

everything - all client-server network communications are encrypted. This is the default value.

Dnsname

```
<Option name="ObjectServerParms">
     dnsname=dns-ip-address
</Option>
```

This parameter of the ObjectServerParms option specifies the IP address of the Domain Name Servers that is used for authentication. Specify this parameter when the operating environment of your Authentication Server uses Internet Protocol Version 6 (IPv6) addresses.

EnableFIPS

```
<Option name="EncryptFIPS">FALSE</Option>
```

This option enables the Authentication Server to run in compliance with Federal Information Processing Standard 140-2. The default value is FALSE. After you install

your Authentication Server with the DataFlux Secure software, the configuration process can be directed to change the value to TRUE.

FIREBIRD and FIREBIRD_LOG

```
<OptionSet name="SetEnv">
     <Option name="FIREBIRD">install-path\lib\fbembed</Option>
     <Option name="FIREBIRD_LOG">install-path\var\log</Option>
</OptionSet>
```

The FIREBIRD environment variable specifies the installation path of the default transactional database, which is Authentication Server to manage users, groups, domains, logins, and shared logins. The transactional database is required to be stored on the Authentication Server host.

You can reconfigure your Authentication Server to use an Oracle database rather than the transactional database, as described in Configure Oracle to Store Users, Groups, Domains, Logins, and Shared Logins. The Oracle database does not use the options FIREBIRD or FIREBIRD_LOG.

The FIREBIRD_LOG environment variable identifies the directory that stores the log files that are generated by the Authentication Server's transactional database.

In the Windows operating environment, the default values for FIREBIRD and FIREBIRD_LOG are set in the configuration file by the Authentication Server installation process. In the UNIX and Linux operating environments, the Authentication Server startup script dasadmin sets the environment variable FIREBIRD_LOG.

License

```
<OptionSet name="License">
     <OptionSet name="Primary">
          <Option name="Provider">SAS</Option>
          <Option name="Location">depot-path</Option>
     </OptionSet>
</OptionSet>
```

The License option provides information about the types of license checks that are performed by the Authentication Server. The value of the Provider option is SAS, and uppercase is required. The value of the Location option specifies the path to the SAS SETINIT. The default path is as follows:

```
depot-path\sid_files\site-filename.txt
```

In the Windows operating environment, a typical entry for the Location option is as follows:

```
<Option name="Location">C:\my-user\Depot_release-week\sid_files\
     DMPversion_09CRZD_70142972_Win_X64_Wrkstn_Srv.txt</Option>
```

The default value for the License option is set into the configuration file when you install the Authentication Server.

Each product receives a unique "site" file, as specified in your Software Order E-Mail. Multiple site files can share a single sid-files directory.

NetworkEncryptAlgorithm

```
<Option name="NetworkEncryptAlgorithm">algorithm</Option>
```

The NetworkEncryptAlgorithm option specifies the encryption algorithm that is used to encrypt network data transfers between clients and the Authentication Server. Valid values for this option are `SASProprietary` and `AES`. The default SASProprietary encryption algorithm uses 56-bit keys. You can upgrade to the 256-bit keys of the AES encryption algorithm by installing the DataFlux Secure software with the Authentication Server software. The configuration process for DataFlux Secure changes the value of this option to AES. For more information, see Configure Encryption.

ObjectServerParms

```
<Option name="ObjectServerParms">
     auth-server-parameter-options
</Option>
```

The ObjectServerParms option specifies a series of Authentication Server parameters. The parameters can be specified in any order. The parameters are delimited by blank spaces.

PrependEnv

```
<OptionSet name="PrependEnv">
     <Option name="your-variable">your-prepend-value</Option>
</OptionSet>
```

The PrependEnv option will find the indicated OS environment variable and prepend the option value to the beginning of the existing value. If the environment variable does not exist, it will be created and set to the option value. The PrependEnv option will not add a delimiter of any sort between the existing and new environment variable value. If a semi-colon (;) is needed, then it must appear as the last character in the option value.

Port

```
<Option name="Port">21030</Option>
```

The Port option identifies the port that the server runs on. 21030 is the default value.

PrimaryProviderDomain

```
<Option name="PrimaryProviderDomain">your-domain</Option>
```

The PrimaryProviderDomain option specifies the domain that is used first by default when a user submits credentials without a domain. The value of the option must be a domain name that is included in the AuthProviderDomain option set.

If `your-domain` contains spaces, then enclose the name in quotation marks.

PublicUserGroupManagementPolicy

```
<Option name="PublicUserGroupManagementPolicy">READ</Option>
```

The PublicUserGroupManagementPolicy option specifies the access permission that is applied to members of the PUBLIC group. By default, members of the PUBLIC group have read access to the group membership information that is displayed in Data Management Studio. Groups are displayed after PUBLIC members connect to the Authentication Server and select the Groups riser. To remove read access, either add comment characters around the option above, remove the option text entirely, or remove the READ value. READ is the only valid value for this option.

SetEnv

```
<OptionSet name="SetEnv">
     <Option name="your-variable">your-value</Option>
</OptionSet>
```

The SetEnv option defines environment variables and assigns values to those variables. Use this option to set environment variables that are required for Active Directory and LDAP authentication on the host of the Authentication Server. See the following options FIREBIRD and FIREBIRD_LOG. See also the environment variables that are set for AuthProviderDomain.

SystemUsers

```
<SystemUsers>
     <Option name="Account">domain\uid1</Option>
     <Option name="Account">domain\uid2</Option>
</SystemUsers>
```

The SystemUsers option defines administrative accounts for the Authentication Server. The user IDs must represent existing accounts in the specified domains.

TrustedUsers

```
<OptionSet name="TrustedUsers">
     <Option name="Account">domain\userid1</Option>
     <Option name="Account">domain\userid2</Option>
</OptionSet>
```

The TrustedUsers option set defines the user accounts that are privileged to act on behalf of other users, for the purpose of retrieving information from the Authentication Server. Trusted users are authorized to read Authentication Server data, but not to add, modify, or delete that data.

> **Note:** Because of their differing permissions, trusted users should not also be system users.

Trusted user accounts enable SAS Federation Servers to query relational databases. To execute a relational query, the Federation Server must first obtain group membership information from the Authentication Server for the user who defined the query. This is necessary because the user who defined the query must be authorized on the Federation Server to submit the query and store retrieved data. The user who requested the query only needs permission to make the request.

If the defining user is authorized, then the Federation Server connects to the relational database and submits the query. The relational query is authenticated and

authorized on the relational database using a shared login that the Federation Server obtains from the Authentication Server.

Glossary

A

Active Directory

an authentication mechanism in the Windows operating environment, with LDAP-like directory services and DNS-based naming.

administrator

an individual who has been granted access to all authentication objects except for passwords in the configuration file as_server_aspsql.xml.

AES encryption

the advanced encryption standard is optionally available on Authentication Servers to encrypt specified network traffic using 256-bit keys.

authentication

the process of verifying the identity of an individual.

authentication data store

a database that contains definitions of domains, users, groups, and shared logins. The database is accessed by an Authentication Server.

authentication mechanism

a program that authenticates users who login to that mechanism's domain.

Authentication Server

a component of the Data Management Platform that provides a central location for the management of connections between the Data Management Studio client, the DataFlux Federation and Data Management Servers, and native database servers.

authorization

the process of determining which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.

C

consumer

a user or group who is allowed to use a shared login to connect to a database.

D

DNS

the Domain Name System uses authoritative servers to assign names in domains and sub-domains. DNS provides translation services between domain names and IP addresses.

domain

a collection of logins designated to be authenticated using the same or like authentication mechanism.

DSN

Database Source Names enable ODBC drivers to connect to data sources.

E

encryption

the act or process of converting data to a form that only the intended recipient can read or use.

G

group

an object in the authentication data store that represents a collection of users and other groups. A group can be a consumer and/or manager of a shared login.

H

host authentication

a process in which a server sends credentials to its host operating system for verification.

L

LDAP

the lightweight directory access protocol is used to access directories or folders. LDAP servers provide an authentication mechanism that can be accessed by Authentication Servers.

login

a DataFlux copy of information about an external account. Each login includes a user ID and belongs to one user or group. Most logins do not include a password.

M

manager

a user or group in the authentication data store that has been granted permission to add and delete consumers from a shared login.

member

a user or group who has been added to a group.

O

ODBC

The Open Database Connectivity Standard is an application programming interface that enables applications to access data from a variety of database management systems.

owner

a user in the authentication data store that has been given permission to add and delete the members of a group. Each group is required to have one and only one owner at all times.

P

PAM

in UNIX and Linux, programmable authentication modules in the operating environment enable authentication across a network.

PUBLIC

this default group, which cannot be edited, contains all users who have authenticated in the host environment of the Authentication Server, but do not have a user definition on the server.

pw

the default authentication mechanism in UNIX and Linux.

S

SASProprietary encryption

the default encryption algorithm for the Authentication Server.

shared login

an object in the authentication data store that associates a collection of users and groups with an outbound login that connects the consumers of that shared login to a database server.

U

user

an object in the authentication data store that associates one or more logins with one individual. A user can be a member of a group, a consumer of a shared login, or be granted access on a DataFlux server.

user definition

same as user. This term is used to differentiate objects in the authentication data store from the individuals who run client applications.

USERS

a default group that includes all individuals who have a user definition and have logged in at least once.