SAS® Federation Server Manager 4.3: User’s Guide
Contents

What's New in SAS Federation Server Manager ........................................ v

PART 1  Introduction to SAS Federation Server Manager  1

Chapter 1 • Overview of SAS Federation Server Manager  ........................................ 3
  About SAS Federation Server Manager .................................................. 3

Chapter 2 • SAS Federation Server Security Features  ........................................... 5
  About SAS Federation Server Security Features ........................................ 5
  About Authorizations ............................................................................. 6
  Understanding DSNs and Permissions .................................................... 8
  Table, Column, and Row-Level Security ................................................ 9
  About the Data Masking Functions ....................................................... 10

Chapter 3 • Navigating SAS Federation Server Manager  ....................................... 13
  Log On to SAS Management Console .................................................. 13
  SAS Federation Server Manager Home Page ........................................ 14
  Server Preferences and Settings .......................................................... 15
  SAS Federation Server Objects and Views ............................................. 16

PART 2  Configuring SAS Federation Server Manager  21

Chapter 4 • Post-Installation Configuration ....................................................... 23
  Configuration Steps .............................................................................. 23

Chapter 5 • Configuring SAS Federation Server ............................................... 31
  Configuration Tasks .............................................................................. 31
  User and Group Authorizations .......................................................... 34
  Granting Permissions for Federation Server and Associated Objects ........... 36

Chapter 6 • Configuring Access to Data Sources ............................................... 39
  Working with Data Services ................................................................. 39
  Creating a Data Service ........................................................................ 40
  Using the Generic Data Service Template ............................................ 61
  Editing a Data Service .......................................................................... 61
  Working with Data Source Names (DSNs) ............................................ 62
  Establishing DSN Permissions ............................................................... 64

Chapter 7 • Working with Federated Data ......................................................... 65
  Catalogs and Schemas ......................................................................... 66
  Working with FedSQL Views ............................................................... 71
  Caching FedSQL Views ......................................................................... 74
  Refreshing Cached Data ........................................................................ 78
  Column and Row-Level Security ........................................................... 84
  Defining User Functions for Row-Level Security .................................... 87
PART 3  Advanced Topics  91

Chapter 8 • Working with the SQL Console .................................................. 93
  SAS Federation Server Manager Console .............................................. 93
  Working with Information Views ......................................................... 95

Chapter 9 • Working with DS2 Dialect ....................................................... 97
  Overview of DS2 on SAS Federation Server ........................................ 97
  Create a DS2 DSN .............................................................................. 97
  DSN and DS2 Object Permissions ..................................................... 98

Chapter 10 • Data Quality and Cleansing Functions .................................. 99
  About Data Quality on SAS Federation Server .................................... 99
  Data Quality Functions in SAS Federation Server Manager ............... 106

Chapter 11 • Data Masking ......................................................................... 109
  About the Data Masking Functions ..................................................... 109
  Displaying Data Masking Functions in the Console ......................... 111
  Data Masking in a FedSQL View ....................................................... 111
  Data Masking Examples .................................................................... 113

Chapter 12 • SQL Logging .......................................................................... 115
  About SQL Logging ............................................................................ 115
  Enable SQL Logging .......................................................................... 115
  Viewing the SQL Log .......................................................................... 116
  SQL Logging Transactions .................................................................. 120
What’s New in SAS Federation Server Manager

Adobe Flash End-of-Life

The SAS Federation Server user interface has been rewritten in HTML5. Previously, applications in the SAS Platform, as well as many SAS solutions that are based on SAS 9.3 and SAS 9.4, used the Adobe Flash Player to provide interactive user interfaces. Adobe announced that it intends to end support for Flash technology and will cease to update and distribute the Flash Player at the end of 2020. Browser vendors will disable Flash by default in 2019. For more information about Adobe Flash end-of-life, see SAS Software and Its Use of the Adobe Flash Player.
What's New in SAS Federation Server Manager
Part 1

Introduction to SAS Federation Server Manager

Chapter 1
  Overview of SAS Federation Server Manager .......................... 3

Chapter 2
  SAS Federation Server Security Features ............................... 5

Chapter 3
  Navigating SAS Federation Server Manager .......................... 13
Chapter 1
Overview of SAS Federation Server Manager

About SAS Federation Server Manager

Introduction

SAS Federation Server Manager is a web application that enables administrators to configure and secure access to SAS Federation Server and associated data sources. With SAS Federation Server Manager, you can create data services and data source names (DSNs) for connecting to data sources supported by SAS Federation Server.

Most of the functions performed with SAS Federation Server Manager can be accomplished with administration DDL statements. The SAS Federation Server: Administrator’s Guide contains information about administration DDL.

Note: The SAS Federation Server: Administrator’s Guide provides the concepts that are needed to complete tasks in SAS Federation Server Manager.

About SAS Federation Server Manager

SAS Federation Server Manager is the editor used to set up and manage SAS Federation Server. With SAS Federation Server Manager, you can also configure and secure access to a SAS Federation Server. Using SAS Federation Server Manager, an administrator can create data services and DSNs that connect to data sources supported by any SAS Federation Server.

You can access SAS Federation Server Manager through a browser session. SAS recommends that you work within only one browser session of SAS Federation Server Manager. Do not access SAS Federation Server Manager from multiple browser sessions. Operations that take place in one session might not be reflected in subsequent sessions.
Chapter 2
SAS Federation Server
Security Features

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About SAS Federation Server Security Features</td>
<td>5</td>
</tr>
<tr>
<td>Overview</td>
<td>5</td>
</tr>
<tr>
<td>Levels of Security Access (Permissions)</td>
<td>6</td>
</tr>
<tr>
<td>About Authorizations</td>
<td>6</td>
</tr>
<tr>
<td>The Authorizations Tab</td>
<td>6</td>
</tr>
<tr>
<td>Users and Groups</td>
<td>7</td>
</tr>
<tr>
<td>Permissions</td>
<td>7</td>
</tr>
<tr>
<td>Inherited Permissions</td>
<td>8</td>
</tr>
<tr>
<td>Understanding DSNs and Permissions</td>
<td>8</td>
</tr>
<tr>
<td>CONNECT Permission</td>
<td>8</td>
</tr>
<tr>
<td>Federation Server SQL Authorization Enforcement</td>
<td>8</td>
</tr>
<tr>
<td>Data Source Names Tab</td>
<td>9</td>
</tr>
<tr>
<td>Table, Column, and Row-Level Security</td>
<td>9</td>
</tr>
<tr>
<td>About the Data Masking Functions</td>
<td>10</td>
</tr>
</tbody>
</table>

About SAS Federation Server Security Features

Overview

Properly configured security for SAS Federation Server ensures that both the server and its data are secure. Data is protected against unauthorized access, and can be guaranteed secure transmission for transferring data. SAS Federation Server security configuration and maintenance are easily managed with SAS Federation Server Manager. You can assign security for these server objects:

- Federation Server
- Connections: Data Service and DSN
- DS2 package functions and threads
- Catalogs and schemas
- Tables and columns
- Views
Levels of Security Access (Permissions)

For each user or group, you can grant or deny these permissions:

- ADMINISTER
- ALTER TABLE or VIEW
- DROP TABLE or VIEW
- CREATE TABLE or VIEW
- CREATE or ALTER CACHE
- CREATE TABLESPACE
- CREATE DSN
- CONNECT
- DELETE
- EXECUTE
- INSERT
- REFERENCES
- SELECT
- TRACE
- UPDATE

By default, users do not have any permissions. The SYSTEM user or a SAS Federation Server administrator must grant privileges so that users can perform actions and gain access to data. Group permissions are granted and denied in the same manner as individual users. Users who are members of a group inherit the permissions from the group unless explicitly denied in the individual user account. At a minimum, any user who accesses any objects on SAS Federation Server must have CONNECT permissions. See “Granting Permissions for Federation Server and Associated Objects” for additional information.

About Authorizations

The Authorizations Tab

Authorization determines which permissions a user or group object contains in order to gain access to resources and associated data sources. Any object that requires security has an associated Authorizations tab. For example, a federation server object has an Authorizations tab that displays permissions for the users and groups associated with that federation server object. The following example reflects the status of permissions for the account ADDADMIN.
**Figure 2.1 User Authorizations**

**Users and Groups**

The first column on the **Authorizations** tab is **Users and Groups**, which lists the users and groups that are associated with the selected server object. You can add users or groups on the **Authorizations** tab. You can also remove users or groups. However, removing a user from the view does not delete the user object and has no impact on permissions. You can also use the drop-down list to filter the view to users only, groups only, or users and groups together. Here are the various representations of the users and group objects:

- **Italic** font represents a newly added user. It is also the default for cached users such as SASUSERS and PUBLIC.
- **Bold** font indicates that the user has at least one permission denied or granted.
- Normal font suggests that the user object has all privileges inherited.

**Permissions**

The permissions column lists the permissions that are associated with the selected user or group. From here, you can grant, deny, or edit inherited permissions. An inherited grant is a permission that is inherited from a group, or an object in the hierarchy. When Inherit is the active permission, the Grant and Deny permissions are available and Inherit is dimmed. The **Securable** column reflects the object where the granted permission occurred.

- **Permission**: All possible permissions are listed in the SAS Federation Server view.
- **Setting**:

```
<table>
<thead>
<tr>
<th>Setting</th>
<th>Grant</th>
<th>Deny</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant</td>
<td>✓ GRANT</td>
<td>✓ DENY</td>
</tr>
<tr>
<td>Inherited Grant</td>
<td>✓ GRANT</td>
<td></td>
</tr>
</tbody>
</table>
```
Inherited Permissions

SAS Federation Server contains an inherent hierarchy of objects, in the following order:
- Server
- Data Service
- Data Source Name (DSN) and Catalog
- Schema (from catalog)
- Table or View (from schema)
- Column (from table or view)

A data service inherits privileges from the server. The privileges on the data service are inherited by the DSN and catalog. Privileges on the catalog are inherited by the schema, passed to the table or view, and finally, the column. This inheritance hierarchy allows for general security settings on higher level objects, and specific exceptions on the subordinate objects. See the *SAS Federation Server: Administrator’s Guide* for a detailed description of permissions and inheritance.

Understanding DSNs and Permissions

**CONNECT Permission**

A user must have CONNECT permission to establish connection with a DSN. This permission is effective from the user object, inherited through the hierarchy, or acquired through group permissions. For a standard DSN, the CONNECT permission must be granted on the following (in order of inheritance):
- the DSN
- the parent data service of the DSN
- the federation server object

For a federated DSN, the CONNECT permission must be on the following (in order of inheritance):
- the DSN
- the federation server object

Permissions for federated DSNs behave differently than standard DSNs:
- Permissions granted on a federated DSN override any permissions that exist for child DSNs that are contained within the federated DSN.
If a user has CONNECT permission on a federated DSN, permissions on any of the child DSNs that are contained within (standard or federated) are ignored, even if the user is explicitly denied CONNECT permission on any of the child DSNs.

**Federation Server SQL Authorization Enforcement**

With Federation Server SQL Authorization Enforcement enabled, the FedSQL driver is engaged, and the SQL dialect is automatically set to FedSQL. FedSQL provides an additional layer of object-level security for the connection, and SQL statements are secured before processing them. If Federation Server SQL Authorization Enforcement is disabled, object-level security is bypassed and a user is granted all FedSQL privileges regardless of which privileges the user had been previously granted or denied. If Federation Server SQL Authorization Enforcement is disabled, an administrator can choose the Native SQL dialect associated with the data source. The SQL dialect for BASE data services is always FedSQL.

You can enable SQL Authorization Enforcement when you create a standard or federated DSN.

- **Enabled**: When SQL Authorization Enforcement is enabled, the FedSQL driver is engaged, and the SQL dialect is automatically set to FedSQL. With FedSQL an additional layer of object-level security is enabled for the connection and SQL statements are secured before processing them.

- **Disabled**: If SQL Authorization Enforcement is disabled, object-level security is bypassed and a user is granted all FedSQL privileges regardless of which privileges the user has been granted or denied. If SQL Authorization Enforcement is disabled, an administrator can choose Native SQL dialect. For example, if you are connected to Oracle, then native dialect is SQL supported by Oracle.

**Data Source Names Tab**

The **Data Source Names** tab shows the DSNs that are associated with the selected federation server object. This list also includes system-generated DSNs such as ADMIN, BASE, SQL_LOG, and SYSPROC. See the *SAS Federation Server: Administrator’s Guide* for more information about the system-generated DSNs. Information associated with Data Source Names includes the following:

- **Connect Permission**: A list of users and groups that have CONNECT permission for the selected DSN.

- **Members**: (Federated DSNs only) A list of child DSNs belonging to the federated DSN. The associated data source is annotated for each child DSN. You can add or remove data source names.

- **Attributes**: Shows properties of the DSN such as Security / SQL Authorization Enforcement, and associated connection string to name a few.

**Table, Column, and Row-Level Security**

Table, column, and row-level security provide an additional layer of security for data when there is a need to control access at a granular level. Security is provided at the table, column, or row level in the database so that users have access to only the data that they require.
By default, SQL requires a column privilege to override a privilege placed on the table. In other words, a column GRANT overrides a DENY on the table. In the event that column privileges override table privileges, the privileges that are displayed on the table reflect the user's overall permissions by taking the column privileges into account.

A DENY placed on a table where GRANT is set at the column level results in one of the following conditions:

- If the table is not inheriting privileges from a column because the privilege was directly granted at the table level, the securable column changes from the name of the table (to columns), because the table is now inheriting GRANT permission from the column. Any other columns in the table for which explicit privilege is not granted displays an inherited DENY.

- If the table is inheriting GRANT privilege from one or more columns, the explicit DENY does not result in a privilege change. If any columns with explicitly GRANT permission are set to INHERIT, the explicit DENY is reflected on the table.

Performing the inherit action on the table cascades to the column level. GRANT or DENY privileges are revoked for all columns that have explicit GRANT or DENY set as the inherited privilege. The columns inherit the permission level that is in force for the table. For additional information about table, column, and row-level security see the SAS Federation Server: Administrator's Guide.

---

### About the Data Masking Functions

Data masking in SAS Federation Server is a series of FedSQL functions that are accessed through the Console using the SYSCAT.DM.MASK function in a SELECT statement. Here is a brief description of each of the data masking functions.

**ENCRYPT**

The ENCRYPT function masks the values in a column by encrypting a single value using symmetric key encryption. Encrypted values cannot be decrypted if a KEY argument is not specified and the ENCRYPT_KEY package configuration option is not set.

```sql
SYSCAT.DM.mask('ENCRYPT', "value"
/ *,
'ALG', 'AES/FIPS|AES|SAS002|BASE64|SAS004|SAS003|SAS001',
'KEY', 'encrypt_key',
'DETERMINISTIC', YES|TRUE|ON|1|NO|FALSE|OFF|0,
'EXPAND_PREC', YES|TRUE|ON|1|NO|FALSE|OFF|0,
'CASE', 'U|L',
'STRIP', 'BLANK|UNICODESP|UNICODESPACE|ANY|ALL|WS'
) */ )
```

**Note:** The ENCRYPT function preserves the data type of the original column if the data type is character (for example, CHAR, NCHAR, VARCHAR). If the column is not a character data type, the output produces a binary data result.

**DECRYPT**

The DECRYPT function unmasks the values in a column by decrypting a previously encrypted value using symmetric key encryption. The DECRYPT rule returns NULL if a KEY argument is not specified and the ENCRYPT_KEY package configuration option is not set.

```sql
SYSCAT.DM.mask('DECRYPT', "value"
/ *, 'ALG', 'AES/FIPS|AES|SAS002|BASE64|SAS004|SAS003|SAS001', ) */ )
```
HASH
The HASH function masks the values in a column by hashing a single value into a fixed-length hash digest or HMAC string. The HASH function is not reversible.

`SYSCAT.DM.mask( 'HASH', "value"
/*[,'ALG', 'MD5|SHA256',
'CASE', 'U|L',
'KEY', 'encrypt_key']*/ )`

TRANC
The TRANC function masks the values in a column by transliterating characters from an input string to characters in an output string. Ensure that the mapped result is “lossy” (many instances of mapping multiple input character values to a single output character value) to prevent inference of the original value.

`SYSCAT.DM.mask( 'TRANC', /*Expression*/ )`

RANDOM
The RANDOM function masks the values in a numeric column by replacing them with uniformly distributed pseudo-random numbers. The RANDOM function is not reversible.

`SYSCAT.DM.mask( 'RANDOM', /*Expression*/ )`

RANDATE
The RANDATE function masks the values in a date column by replacing them with pseudo-random date values.

`SYSCAT.DM.mask( 'RANDATE', /*Expression*/ )`

RANSTR
The RANSTR function masks the values in a column by replacing the values with random strings. Strings are generated by an algorithm that uses characters from the source string in the generation process, adding padding characters if necessary. Padding is placed to the left of the string unless RIGHT is specified.

`SYSCAT.DM.mask( 'RANSTR', /*Expression*/ )`

RANDIG
The RANDIG function masks the numeric values in a column by replacing digits with strings of random digits. Strings are generated by an algorithm that uses digits that are derived from the base number system of the source value, adding padding digits if necessary. Padding is always to the left of digits.

`SYSCAT.DM.mask( 'RANDIG', /*Expression*/ )`

For additional information related to data masking, see the SAS Federation Server: Administrator's Guide.
Chapter 3
Navigating SAS Federation Server Manager

Log On to SAS Management Console .................................................. 13
SAS Federation Server Manager Home Page ...................................... 14
Server Preferences and Settings ......................................................... 15
  Application Options ........................................................................ 15
  Change the Appearance of the User Interface .................................. 15
  SAS Federation Server Manager Preferences ................................... 16
SAS Federation Server Objects and Views .......................................... 16
  Overview ....................................................................................... 16
  Federation Server ......................................................................... 16
  Data Service ................................................................................ 17
  Catalogs and Schemas ..................................................................... 17
  Tables and Views ........................................................................... 17
  Columns ....................................................................................... 18

Log On to SAS Management Console

SAS Federation Server Manager is accessed through the standard logon window for SAS applications. Logging on from this window opens SAS Data Management Console from which you can launch SAS Federation Server Manager. To log on to SAS Data Management Console:

1. Click the URL that is supplied in Instructions.html, or paste it into the address field of your browser to display the SAS logon window:
2. In the User ID field, enter your user ID.
3. In the Password field, enter the password for your user ID.

SAS Federation Server Manager Home Page

Once logged in to SAS Federation Server Manager, but prior to logging in to SAS Federation Server, the Summary tab appears, showing basic information such as server name, port, and connection attributes. It is not until you select a federation server in the tree and log on that you will see complete information as shown below.

1. **Home tab**: The home tab is located on the left navigation bar. The home tab displays all of the federation servers and associated objects in the navigation tree.

2. **Navigation tree**: The SAS Federation Server Manager contains all registered servers and associated objects. You can expand or collapse objects within the tree. Federation server objects expand in the following order: federation server → data service → catalog → schema → table/view/cache.
3 **Summary** tab: This tab is the first of the information tabs for the selected federation server. The tabs that follow are visible only when logged on to a federation server object. However, the **Summary** tab is always visible. The detail area displays summary information for the selected federation server object, such as server name, address, port, and connection attributes.

4 **Selection indicator:** The selection indicator responds to a selection in the tree and shows the type of object that is selected. A **FEDERATION SERVER** is shown in this example. Other possible indicators are **DATA SERVICE**, **CATALOG**, **SCHEMA**, and **TABLE**.

5 **Application Options:** Access documentation at the **Help Center**, Use **Settings** to configure your Region and Language, or configure server settings at **SAS Federation Server**.

---

**Server Preferences and Settings**

**Application Options**

Use **Application Options** to edit user preferences or customize accessibility settings for SAS Federation Server Manager. Changing these settings does not impact other users. To access these settings, select **Application Options** in the upper right corner, and select **Settings** from the drop-down list.

*Figure 3.3 Application Options*

---

**Change the Appearance of the User Interface**

You can change the appearance of the web interface by using the Theme setting. The theme specifies the collection of colors, graphics, and fonts that appear in SAS Federation Server Manager. The available themes are:

- **Illuminate**
  - Default. This theme has a clean and uncomplicated color palette that is easy to use.

- **High Contrast**
  - This theme presents a dark background with high-contrast foreground elements to meet the needs of users with low vision.

1. Select **Application Options** in the upper right corner, and select **Settings** from the drop-down list.
2. Select **Global, General** and using the drop-down menu under **Theme**, select the theme of your choice.
3. Click **Close** when you are finished. The theme change takes effect after you close **Settings**.
**SAS Federation Server Manager Preferences**

To configure behavior for SAS Federation Server Manager, follow these steps:

1. Select **Application Options** in the upper right corner of your screen and select **Settings** from the drop-down list.
2. Select **SAS Federation Server Manager** to set the following options:
   - **Maximum rows returned**: The default is 50.
   - **Maximum objects displayed in navigation tree**: The default is 50.
   - **SQL statement delimiter characters**: The default is semicolon (;).
   - **Refresh connection information**: The default is **Never** (manual refresh only). If you choose **Periodically**, enter a value in minutes. For example, if you enter a value of 60, the server connections are refreshed every hour.

---

**SAS Federation Server Objects and Views**

**Overview**

Federation servers and associated objects are arranged in a view in the left panel of the user interface. When selecting a federation server object in the tree, the **More Options** menu contains functions that are specific to the selected federation server object. The following objects can appear in the tree:

- Federation server
- Data service
- Catalog
- Schema
- Native Table or Native View
- FedSQL View
- Cached FedSQL View
- DS2 Package
- DS2 Method

**Federation Server**

Each federation server object displays information tabs that summarize configuration and connection properties for the selected server object. The **Summary** tab is displayed at all times. The remaining tabs are displayed after logging on to a federation server. Here are the tabs that are displayed when you are logged in to SAS Federation Server:
Summary

The Summary tab reflects the server name, description, server DNS name, port, and connection attributes. The SQL logging options for the selected server are also shown on this tab.

Current Connections

The Connections tab lists the current connections for the selected federation server object and also displays current sessions if requested. You must be logged in to a server to see all server connections and sessions. You can toggle between Show Connections or Show Sessions by using the drop-down menu. Each time that you click the Connections tab, the displayed content is refreshed. You can also use Refresh to refresh connections and sessions.

Data Source Names

The Data Source Names tab shows the DSNs that exist for the selected federation server. This list also includes system-generated DSNs such as ADMIN, BASE, and SQL_LOG. DSNs display a Connect Permission and Attributes tab. Federated DSNs also display a Members tab that lists the DSNs that belong to the federated DSN.

Authorizations

Each federation server object contains an Authorizations tab containing the users and groups that are specific to the server. By default, users are not granted any privileges as they are created. The SYSTEM user or federation server administrator must grant permissions and privileges so that users can perform actions and gain access to data. See “About Authorizations” for additional information.

Data Service

Every federation server has one BASE data service that is created by default. Each data service object contains these tabs:

Summary

Contains the data service name, data source type, and attributes.

Data Source Names

Contains all of the DSNs that are affiliated with this data service.

Authorizations

Contains the users and groups, and permissions that apply to the data service.

Catalogs and Schemas

Both of these objects contain a Summary and Authorizations tab.

Tables and Views

Native Table

When a table or view object is selected, the name and type of the object is displayed in the right panel, accompanied by a series of tabs that explain the properties of the selected object.

This is a native table that you can create a new FedSQL view from Table, view data, and display authorizations, including row-level authorizations. You can also set row-level security. Other table types can be associated with this icon (for example, a Teradata NOPI table is displayed as a native table).
Note: This icon also depicts a SYSTEM table such as the Memory table used for memory data store (MDS).

**FedSQL View**
This is a FedSQL view from which you can cache data. It displays the following tabs:

- **Summary**: Shows the name, schema, catalog, and data service associated with the FedSQL view. You can select **Cache FedSQL View** from the view if it is a Definer type cache.
- **Query**: Displays the query used to create the view.
- **Data**: Displays the data used for the view when you click **View Data**.
- **Authorizations**: Displays user and group authorizations that are associated with the table. You can set new filters, select existing filters, and delete filters.
- **Row Authorizations**: If row-level security is in place, displays existing row filters and the associated user and group authorizations. You can also set row-level security on this tab using **New Filter**.

**Cached View**
A cached view is data cached from a FedSQL view. It displays the following tabs:

- **Summary**: Shows the name, schema, catalog, and data service associated with the cached FedSQL view. You can select **Update Cache Table** from the view.
- **Cache**: Shows the start, end, and run time for the cache. From here you can view the cache definition or refresh cache. This tab also displays the location of the cache table and type: invoker or definer. You can also enable, disable, and delete cache from this tab.
- **Query**: Displays the query used to create the view.
- **Data**: Displays the data used for the view when you click **View Data**.
- **Authorizations**: Displays user and group authorizations associated with the table.
- **Row Authorizations**: If row-level security is in place, displays existing row filters and the associated user and group authorizations. You can also set row-level security on this tab using the **New Filter** icon.

**DS2 Package**
A DS2 package displays the following tabs:

- **Summary**: Shows the name, schema, catalog, and data service associated with the DS2 package.
- **Code**: Displays the code that was used to create the DS2 package. You can export or print from this tab.
- **Authorizations**: Shows the users and groups that have the required EXECUTE permission for DS2.

**Columns**
You can expand table objects to view the columns associated with the table using one of the following techniques:
• Select the table object in the navigation tree and click the **view columns** icon at the top of the data.

• Right-click the table object in the navigation tree and click the **View columns** button that pops up.

When the table is expanded, a column pane opens to the right of the table showing the columns associated with the selected table. When you select a column, a **Summary** tab and **Authorizations** tab are displayed in the right pane. Only one column can be viewed at a time. Each COLUMN_TYPE is associated with an icon. For DS2 Methods, there are two columns of icons, one for COLUMN_TYPE and one for DATA_TYPE.
Part 2

Configuring SAS Federation Server Manager

Chapter 4
Post-Installation Configuration ........................................ 23

Chapter 5
Configuring SAS Federation Server ................................... 31

Chapter 6
Configuring Access to Data Sources ................................. 39

Chapter 7
Working with Federated Data ........................................... 65
Chapter 4  
Post-Installation Configuration

Configuration Steps

The Instructions.html File

After you install SAS Federation Server, you must perform additional configuration steps before you can use SAS Federation Server Manager. At the end of the installation, the SAS Deployment Wizard produces an HTML document named Instructions.html. If your server tier and middle tier are hosted on separate machines, there is an Instructions.html file for each machine. The Instructions.html file is located in SAS\Config\Lev\Documents\. Here is an outline of tasks that require attention:

1. Verify that all installation and configuration steps in the Instructions.html file have been completed.
2. User security: Create users, groups, and roles.
   - Create a Shared Login group if shared logins will be used.
   - Verify that the trusted user was created at installation.
3. (Optional) Specify an encryption level for SAS Federation Server.
4. If you are upgrading from a previous release of SAS Federation Server Manager, move scheduled jobs to the new version.

SAS Metadata Server

User Requirements and Roles

To access SAS Federation Server Manager, users might require group membership that includes assignment of specific roles.

- A user object requires membership to the SASUSERS group, with the Federation Server Manager: Operation role.
• An administrator requires membership to the **SAS Federation Server Administrators** role. The Federation Server Manager: Operation role is assigned by default.

• The SAS Federation Server system user that is created at installation is sasfedadm, and is a member of the **SAS Federation Server Administrators** group.

• The SAS Trusted User that is created at installation is sastrust. This account replaces the trusted user formerly created in DataFlux Authentication Server.

• The SAS Federation Server object is created on SAS Metadata Server at installation. The federation server object is no longer defined in SAS Federation Server Manager.

**SAS Federation Server Administrators**

A user becomes an administrator when his or her account is added to the SAS Federation Server Administrators group in SAS Metadata Server. There are two ways to make an account administrator:

1. Adding the user account to the **SAS Federation Server Administrators** group. By default, only the Metadata Administrator can perform this action.

2. Issue GRANT ADMINISTER DDL on the SAS Federation Server. Only the system user has the authority to grant or revoke the ADMINISTER privilege through the use of administration DDL. The ADMINISTER permission is available on the server object only.

This action grants the ADMINISTER privilege to the user object. Only the SAS Federation Server System user (sasfedadm), or SAS Metadata Administrator (sasadm) can grant the ADMINISTER privilege.

*Note:* See the **SAS Federation Server: Administrator’s Guide** for information about the ADMINISTER privilege and DDL statements.

**Specify Server Encryption Level**

Use SAS Management Console to specify or change the encryption level for a particular SAS Federation Server:

1. In SAS Management Console, locate your federation server object by expanding **Environment Manager** ➔ **Server Manager** ➔ **Federation Server - hostname - logical server**.

2. Expand the logical server entry and select the server definition for which you want to change encryption. The **Connections** tab displays the current connections defined for the selected server.

3. On the **Connections** tab, select a connection and right-click. Select **Properties** from the drop-down menu.

4. Select the **Options** tab, and select **Advanced Options**.

5. Select the **Encryption** tab, and select an option from the **Server encryption algorithm** list menu.
6. Click **OK** to exit the Advanced Options dialog box, and click **OK** to close connection properties.

7. Restart SAS Federation Server to update the server encryption algorithm.

   *Note*: The encryption algorithm that you choose for SAS Federation Server should always match the encryption algorithm that is in place for SAS Metadata Server.

---

**Shared Login Accounts**

**About Shared Logins**

Shared logins consist of a shared login key, the login account, and the users or groups who are members of the (shared) login account. The SAS Federation Server administrator creates and controls the shared logins for SAS Federation Server.

When using a shared login to authenticate to a data source, users do not need to know the credentials of the shared login. The shared login retrieves credentials for the user who is logged on and provides the credentials to SAS Federation Server. In turn, the server connects the user to the database through the appropriate data service or data source name (DSN).

**Outline of Shared Login Tasks**

The implementation of shared logins has changed in SAS Federation Server 4.2. Here is a summary of the tasks:

- Create a shared login key for SAS Federation Server using administrative DDL or in SAS Federation Server Manager in the properties of a federation server object. The shared login key is case sensitive. The key that is defined in SAS Federation Server must match the key that is part of the shared login definition in the SAS Metadata Server.

- Create a shared login account (group) in SAS Metadata Server using SAS Management Console. The shared login account includes the login to be shared and its domain.
• Add consumers of the shared login as members of the shared login account. Consumers are SAS Federation Server user accounts or groups. You should never use the actual shared login group as a consumer group in a DSN.

• Create a data service for the applicable data source. In the DSN, specify that the data will be accessed with a shared login.

**About the Authentication Domain**

When establishing connection to the SAS Federation Server, the following logic is used to find the proper login:

• If connecting with a DSN that is configured to use a personal or group login, SAS Federation Server uses the authentication domain that is associated with the data service to look up a login for the user.

• If connecting with a DSN that is configured to use a shared login, SAS Federation Server uses the authentication domain that is associated with the data service and appends the domain with a suffix of "@<shared login key>" to look up a login for the user.

**Creating a Shared Login**

The following topics outline the basic steps to create a shared login for SAS Federation Server:

1. Set a shared login key (SAS Federation Server Manager).
2. Create the shared login account (SAS Management Console).
3. Create a data service and a DSN for the data source (SAS Federation Server Manager).

**Set a Shared Login Key**

The shared login key is used when configuring an authentication domain in SAS Metadata Server. The shared login key is case sensitive. The following steps show how to set a shared login key with SAS Federation Server Manager:

1. Locate the federation server object in the tree, and log on to the server if prompted to do so.
2. Click on in the upper right corner, and select Properties to open the Federation Server Properties dialog box.
3. Click the Security tab and enter a shared login key, for example, FSKey1.
4. Click OK to exit the properties dialog box.

**Create a Shared Login Account**

The shared login account is actually a group that serves as the shared login account, so the name of the group should reflect that. See Step 4a below.

1. Log on to SAS Management Console
2. On the Plug-ins tab, select User Manager.
4. In the New Group Properties dialog box:
a. On the **General** tab, enter a name for the shared login (for example, Oracle Shared Login for FedServer).

b. On the **Members** tab, add users and groups who will use the shared login.

c. On the **Accounts** tab, add the account and password.

d. Select **New** for Authentication Domain.
   - Enter an Authentication Domain name using this format:
     
     `<data_service_domain>@<shared_login_key>`

     For example, if the domain for the data service is OracleAuth and the shared login key is `FSKey1`, then the shared login domain must be `OracleAuth@FSKey1`. The shared login key is case sensitive and must match the shared login key that was set in SAS Federation Server Manager.

   - Select **Outbound only** and **Trusted only** for the domain.

   **Figure 4.2 New Authentication Domain Dialog Box**

   ![New Authentication Domain Dialog Box](image)

   **Outbound only**: An outbound domain is used only to provide SAS applications with access to external resources, such as a third-party vendor database.

   **Trusted only**: The trusted user is a privileged service identity that can act on behalf of all other users. A login in a trusted domain can be accessed only by a trusted user.

5. On the **Authorizations** tab, ensure that the SAS Administrators group has these permissions:
   - ManageMemberMetadata
   - ManageCredentialsMetadata
   - ReadMetadata
   - WriteMetadata

**Create a Data Service (with DSN)**

When you create a data service, a DSN with the same name is automatically created for you. Use SAS Federation Server Manager to perform the following tasks:

1. Select a federation server object in the tree, and log on to the server if you are prompted.

2. Click the New Data Service icon on the toolbar above the navigation tree. You can also create a data service from the toolbar in the **Summary** tab.

3. Enter the name of the data service as **Identification** for the new data service.

4. Select a **Service Type** from the list of data sources.
5. Configure one or more drivers to connect to your Data Source. If necessary, use Advanced Driver Options to specify a default driver, and to further define connection options.

6. Optional: Use Advanced Driver Options to specify a default driver, and to further define connection options.

7. Specify the name of a Catalog to contain schemas that belong to the data source.

8. Select an Authentication Domain to use for authenticating users.

**Figure 4.3 Defining the Data Service Authentication Domain**

![New Data Service](image)

**CAUTION:**
Select a stand-alone data source domain as shown in the example above. Do not select the domain with the shared login key that was created in SAS Metadata Server. When the DSN is set to use a shared login, SAS Federation Server appends the selected domain with @ and the shared login key and verifies that data source@<shared login key> exists in SAS Metadata as a valid authentication domain that includes user and password account information.

9. Review the configured settings in Summary and click Finish.

**Set the Shared Login Indicator in the DSN**

1. Select the Data Source Names tab that is affiliated with the Oracle data service that you just created. Select the DSN that is named for the new data service.

2. Select Properties for your data source [ ], and select Step 4 - Access in the DSN Properties dialog box.

3. Select Shared login where it says Specify the type of login required to access this DSN.
4. Select a group from the **Consumer group** drop-down list and click **Finish**.

*Note:* The Consumer group identifies which shared login should be used if a conflict occurs for a user. The Consumer group should be a group that is directly or indirectly a member of the shared login.
# Chapter 5

## Configuring SAS Federation Server

### Configuration Tasks

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisite</td>
<td>31</td>
</tr>
<tr>
<td>Federation Server Properties</td>
<td>31</td>
</tr>
<tr>
<td>Define Security Objects</td>
<td>32</td>
</tr>
<tr>
<td>Enable Logging</td>
<td>33</td>
</tr>
<tr>
<td>Enable Connection Pooling</td>
<td>33</td>
</tr>
<tr>
<td>Purge Unused Cache Tables</td>
<td>33</td>
</tr>
</tbody>
</table>

### Prerequisite

Federation servers must be added in SAS Metadata Server before they are visible in SAS Federation Server Manager. See “Configuration Steps” for a list of configuration tasks for SAS Metadata Server.

### Federation Server Properties

You can access the Properties for a federation server object by clicking on the toolbar.

The federation server properties dialog box contains the following tabs:

#### General

Displays the name and description of the selected federation server. This dialog box also contains the server's DNS name and port number that define the SAS Federation Server in the network. The default port number is 24141. These fields are populated with configuration information from SAS Metadata Server. If you need to edit any of these properties, use SAS Management Console to update the information in SAS Metadata Server. You can test the server’s connection using Test Connection.
Security
Defines the Shared Login Key and Data Masking encryption value and settings

SQL Log
Enables SQL logging and specific logging options. SQL logging requires the
Administer permission.

Schedule
Shows the schedule to purge unused historic cache tables. You can configure cache
settings here.

Configuration
This tab contains configuration information such as server state, loggers, and
federation server cache tables. You must have administrative permission to edit
configuration properties. Changes to properties affect the current server instance and
revert to the previous value once the server is restarted. To change server
configuration properties permanently, edit the server configuration file and restart the
server. See the SAS Federation Server: Administrator’s Guide for more information
about configuration.

Connection Pooling
Enables connection pooling where you can set additional connection options.

Other
Contains configuration settings for Trace File Path, Shutdown Timeout, FedSQL
settings, and Driver transcoding failure behavior. You can also set advanced options
for the server.

Note: Enter advanced options with <key>=<value> pairs using a semicolon to
separate pairs. Use ASCII characters only when setting advanced options.

Define Security Objects

Specify a Shared Login Key
To specify a shared login key if one has not already been specified:
1. Select a federation server object in the tree.
2. Click on ‹ in the toolbar in the upper right left corner and select Properties.
3. Click the Security tab on the Federation Server Properties dialog box.
4. Enter the Shared Login Key and click OK.

CAUTION:
The shared login key is case sensitive.

Data Masking Encryption Key
You can specify an encryption key for data masking, also referred to as RANDOM
SEED. If the encryption value was set using the ALTER SERVER DDL statement, the
key is reflected here. To specify a data masking encryption key in federation server
properties:
1. Select a federation server object in the tree.
2. Click on ‹ in the toolbar in the upper right corner and select Properties.
3. Click the Security tab on the Federation Server Properties dialog box.
4. Enter the Data Masking Encryption key and click OK.
Procedure Language Source Management
Specifies procedure language (PL) source management authorization, which allows DS2 users to publish content to a schema specified in the connection. The options are true or false. The default is true, which limits publishing of DS2 content to administrators and owners of the schema containing code. See Alter Server DDL in the SAS Federation Server Administrator’s Guide for additional information.

Enable Logging
When you enable Logging, any parameters set for logging during the session revert to the default configuration settings upon restart of the server. The information captured here does not impact regular server logging, which is set within the dfs_log.xml configuration file on SAS Federation Server. To enable logging for session events:
1. Select a federation server object in the tree.
2. Click on ☰ in the toolbar above the navigation tree and select Properties.
   
   Note: You can also select More options ☰ from the toolbar in the Summary tab.
3. At the Federation Server Properties dialog box, select the Log tab.
4. Click to select On. Log SESSION and select the events that you want to log.
5. Click OK when you are finished.

See “SQL Logging Transactions” for a brief description of the behavior of each of these transactions.

Enable Connection Pooling
Connection pooling is a reserve of database connections that are maintained so that the connections can be reused as future requests to the database are required. To enable connection pooling, follow these steps:
1. Select a federation server object in the tree, and click on ☰ in the upper right corner.
2. Select Properties from the menu.
3. Click the Connection Pooling tab on the Federation Server Properties dialog box.
4. Select Enable connection pooling.
5. Accept the Maximum Unused Connections default of 50, or select Specify and enter the number of maximum connections allowed.
6. To set a time-out, accept the Unused Connection Timeout (seconds) default of 60 or specify the number of seconds for time-out of unused connections. You can also specify No timeout for pooled connections.
7. Click OK when you are finished configuring Connection Pooling.

Purge Unused Cache Tables
To schedule cache purge on the Schedule tab in federation server properties, follow these steps:
1. Select a federation server in the tree and click on ☰ in the upper right corner.
2. Select Properties from the menu.
3. Select the **Schedule** tab on the Federation Server Properties dialog box.

4. Select an option for purging unused cache tables:
   - Periodically (default is 30 minutes): Enter a time frame, in minutes.
   - Auto: Old cache tables are removed after a CREATE CACHE, REFRESH CACHE, or PURGE CACHE command is issued.
   - Never: Manual purge only.

5. Click **OK** when you are finished.

**User and Group Authorizations**

**Overview**

By default, users are not granted any permissions. The SAS Federation Server administrator must grant privileges so that users can perform actions and gain access to data. Group permissions are granted and denied in the same manner as individual users. Users who are members of a group inherit the permissions from that group unless explicitly denied in the individual user account.

The basic permissions that users require are SELECT and CONNECT. When users log on to SAS Federation Server Manager, they can see only the permissions that have been assigned to them. All other permissions reflect a status of DENY.

**Add Users and Groups to a Server Object**

To add users or groups to a federation server object:

1. Select a federation server object in the tree.

2. Click the **Authorizations** tab in the right pane.

3. Select the **Add Users and Groups** button.

4. In the Choose Members dialog box, enter a user name in the Filter box and select a user object from the results.

5. Click **OK**. The selected object is added to the Users and Groups list. You can proceed to set permissions using the icons in the upper right corner of the pane. The icons from left to right are **Grant** – **Deny** – **Inherit**.
Removing Users and Groups

You cannot delete a user or group in SAS Federation Server Manager. These objects are managed through SAS Metadata Server. However, you can remove a user or group from a federation server object if the following conditions are met:

- The object was added as a result of the add or search feature in SAS Federation Server Manager.

- Permissions for the object have not changed.

To remove a user, select the delete button at the top of the Users and groups list.

If you add a user and grant permissions, those permissions must be reversed, or set back to original status, before you can remove the user from the federation server object. In the following example, user1 was added to the SQL Server data service. Permissions are set to INHERITED DENY and have not been altered. At this point, the delete button is active. Once you change permissions for the user object, the delete button is dimmed:
To remove user1, you must reverse the altered permissions to **INHERITED DENY**. Once the permissions are reversed, the delete button becomes active and you can remove the user from the **Users and Groups** list in the **Authorizations** tab.

---

**Granting Permissions for Federation Server and Associated Objects**

**Overview**

You can configure permissions for SAS Federation Server and associated objects such as data services, DSNs, and catalogs. Permissions set at the server object level are inherited throughout the object hierarchy as follows:

- Schema inherits from catalog.
- Catalog inherits from data service.
- DSN inherits from data service.
- Data service inherits from federation server.

*Figure 5.3  SAS Federation Server Inheritance Hierarchy*

---

**Granting Permissions**

**Federation Server**

To grant permissions for a federation server object:

1. Select a federation server in the tree and select the associated **Authorizations** tab.
2. Select +.
3. Enter a user or group name using the **Filter** field in the Choose Members dialog box.
4. Select + to move a user(s) or group(s) to the **Selected members** field.
5. Click **OK** to return to the **Authorizations** tab.
6. Select the user and set permissions by selecting one or more items in the **Permissions** list.
Data Service
Select a federation server ⇒ data service in the tree and use the following task to grant permissions:
1. Click the Authorizations tab associated with the selected object.
2. Under Users and groups, select +.
3. Enter a user or group name using the Filter field in the Choose Members dialog box.
4. Select the user (or users) and click + to move the object to the Selected members field.
5. Click OK to return to Authorizations.
6. Select the user object and set permissions by selecting one or more items in the Permissions list.

DSN
See “Establishing DSN Permissions” to configure authorizations for a DSN.

Catalog
Select a federation server ⇒ data service ⇒ catalog and use the following task to grant permissions:
1. Click the Authorizations tab associated with the selected object.
2. Under Users and groups, select +.
3. Enter a user or group name using the Filter field in the Choose Members dialog box.
4. Select the user (or users) and click + to move the object to the Selected members field.
5. Click OK to return to Authorizations.
6. Select the user object and set permissions by selecting one or more items in the Permissions list.

Schema
Select a federation server ⇒ data service ⇒ catalog ⇒ schema and use the following task to grant permissions:
1. Click the Authorizations tab associated with the selected object.
2. Under Users and groups, select +.
3. Enter a user or group name using the Filter field in the Choose Members dialog box.
4. Select the user (or users) and click + to move the object to the Selected members field.
5. Click OK to return to Authorizations.
6. Select the user object and set permissions by selecting one or more items in the Permissions list.

Tables and Columns
Select a federation server ⇒ data service ⇒ catalog ⇒ schema ⇒ table (column) and use the following task to grant permissions:
1. Click the Authorizations tab associated with the selected object.
2. Under Users and groups, select +.
3. Enter a user or group name using the Filter field in the Choose Members dialog box.

4. Select the user (or users) and click † to move the object to the Selected members field.

5. Click OK to return to Authorizations.

6. Select the user object and set permissions by selecting one or more items in the Permissions list.

*Note:* You can also configure authorizations on the rows in a table. See Row-Level Security for more information.

**Views**

Select a federation server ➔ data service ➔ catalog ➔ schema ➔ view and use the following task to grant permissions:

1. Click the Authorizations tab associated with the selected object.

2. Under Users and groups, select †.

3. Enter a user or group name using the Filter field in the Choose Members dialog box.

4. Select the user (or users) and click † to move the object to the Selected members field.

5. Click OK to return to Authorizations.

6. Select the user object and set permissions by selecting one or more items in the Permissions list.

*Note:* You can also configure authorizations on the rows in a view. See Row-Level Security for more information.
Chapter 6
Configuring Access to Data Sources

Working with Data Services

Creating a Data Service
- Defining a New Data Service
- Apache HIVE Data Service
- DB2 Data Service
- Greenplum Data Service
- MDS (Memory Data Store) Data Service
- Netezza Data Service
- ODBC Data Service
- Oracle Data Service
- PostgreSQL
- SAP Data Service
- SAP HANA Data Service
- SAS Federation Server Data Service
- SASHDAT Data Service
- Scalable Performance Data Server (SPD Server) Data Service
- SQL Server Data Service
- Teradata Data Service

Using the Generic Data Service Template

Editing a Data Service

Working with Data Source Names (DSNs)
- About DSNs
- DSN Types
- Create a Standard DSN
- Create a Federated DSN

Establishing DSN Permissions
- Overview
- Establishing Connect Permissions

Working with Data Services

Data services control information that identifies the location of data source tables. If a data source does not support native catalogs, you can use SAS Federation Server Manager to define a logical catalog name that serves as an SQL identifier. This enables you to identify each data source uniquely when performing heterogeneous operations.
Data services that require logins must be associated with a domain in SAS Metadata Server. When users connect to the data service through a data source name (DSN), the domain name is used to retrieve the user credentials that are associated with that data service. The credentials are then passed along to the back-end database. User credentials are stored in SAS Metadata Server.

Note: To manage data services in SAS Federation Server Manager, an administrator must have personal credentials to the associated databases. SAS Federation Server Manager connects to a data service behind the scenes using a credential search order of PERSONAL (CSO=PERSONAL). This means that all users attempting to connect to the data service must have their own database credentials.

The administrator can assign privileges to allow users access to the data source. In order to connect, a user must be granted CONNECT privilege on either the SAS Federation Server, a specific data service, or a specific DSN.

Creating a Data Service

Defining a New Data Service

Data services contain connection information and driver specifics to connect with data sources. When you create a data service, SAS Federation Server automatically generates a corresponding DSN that matches the name of the new data service. If the data service name conflicts with the name of an existing DSN, a DSN will not be generated for the data service.

When you create a data service, the configuration options are presented in a series of 5 steps with options that depend on the data source, or service type, that you are configuring. Step 6 presents a summary of the configured options.

• Step 1: Identification - Specify a unique name for the new data service.
• Step 2: Service Type - Select the service type that supports connection to the database.
• Step 3: Data Source - Configure a native or ODBC driver, or both, that connects to the data source.
• Step 4: Catalog - Specify the name of the catalog that will contain the schemas belonging to this data service.
• Step 5: Authentication Domain - Select the domain for user authentication.
• Step 6: Summary - Presents a summary of the activity for the previous steps.

To define a new data service, follow these steps:
1. Select a federation server object in the tree and log on if you have not already done so.
2. Select the New Data Service icon on the toolbar or select New Data Service from more options .
3. Proceed to the topic that contains the specific configuration steps for your data source.
   • “Apache HIVE Data Service”
   • “DB2 Data Service”
   • “Greenplum Data Service”
Apache HIVE Data Service

Use the following steps to create a data service for Apache Hive:

1. **Identification**: Enter the Name of the data service.
2. **Service Type**: Select Apache HIVE from the list of supported data sources.
3. **Data Source**: Configure one or more drivers for your data sources. At the Data Source dialog box, configure one or more drivers.
   
a. Select **Configure a native Apache HIVE driver** and update the fields shown below. The required fields are marked with a red asterisk in the dialog box.

   - Enter a server name.
   - Enter a port number.
   - Enter a schema name.
   - Enter a path to the configuration file where the Hadoop JAR files are located.
   - Enter the subprotocol type.
   - Select the mode of authentication.

   *Note*: See the "SAS Federation Server Driver for Apache Hive" in the SAS Federation Server Administrator's Guide for details about these connection options.

b. Select **Configure an ODBC driver** and update the following configurations:
   - Enter a DSN name or a driver connection string.
   - Enter the name for the default schema.
c. If you have configured more than one driver, select a **Default driver**.

d. Select **Advanced Driver Options** if additional configuration is needed for the data service.

   - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

   *Note:* For information about advanced options, see "SAS Federation Server Driver for Apache Hive" in the *SAS Federation Server Administrator's Guide*.

4. **Catalog**: Specify a **Catalog** name.

5. **Authentication Domain**: Select an **Authentication Domain** from the list of available domains.

6. **Summary**: Verify the settings and click **Finish**.

**DB2 Data Service**

Use the following steps to create a data service for DB2:

1. **Identification**: Enter the **Name** of the data service.
2. **Service Type**: Select DB2 from the list of supported data sources.

3. **Data Source**: At the Data Source dialog box, configure one or more drivers. The required fields are marked with a red asterisk in the dialog box.
   a. Select **Configure a native DB2 driver** and enter the *Database name*.
   b. Select **Configure an ODBC driver**, and specify either a DSN name or a driver connection string.

   *Figure 6.2 DB2 Data Source Configuration*

   ![DB2 Data Source Configuration](image)

   c. If you have configured more than one driver, select **Default driver**.
   d. Select **Advanced Driver Options** if additional configuration is required for the data service.
      - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

      *Note*: For information about advanced options, see the "SAS Federation Driver for DB2" topic in the *SAS Federation Server Administrator's Guide*.

4. **Catalog**: Specify a Catalog name.

5. **Authentication Domain**: Select a domain from the list of available domains.
6. **Summary**: Verify the settings and click **Finish**.

**Greenplum Data Service**

Use the following steps to create a data service for Greenplum:

1. **Identification**: Enter the **Name** of the data service.
2. **Service Type**: Select **Greenplum** as the service type.
3. **Data Source**: At the Data Source dialog box, configure one or more drivers.
   a. Select **Configure a native Greenplum driver** and update the following configurations. The required fields are marked with a red asterisk in the dialog box.
      - Enter the name of the database server.
      - Enter the port number for the server.
      - Enter the name of the database.
      - Enter a schema name.
   b. Select **Configure an ODBC driver**, and specify either a DSN name or a driver connection string. This step is optional.
c. If you have configured more than one driver, select a **Default driver**.

d. Select **Advanced Driver Options** if additional configuration is needed for the data service.

   • Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

   *Note:* For information about advanced options, see the "SAS Federation Server Driver for Greenplum" topic in the *SAS Federation Server Administrator’s Guide*.

4. **Catalog:** Specify a Catalog name.

5. **Authentication Domain:** Select an **Authentication Domain** from the list of available domains.

6. **Summary:** At the Summary dialog box, verify the settings and click **Finish**.

---

**MDS (Memory Data Store) Data Service**

Use the following steps to create a data service for MDS:
1. **Identification**: Enter the Name of the data service and proceed to the next step.

2. **Service Type**: Select MDS (Memory Data Store) as the service type and proceed to the next step.

3. **Data Source**: At the Data Source dialog box, configure the values below. The required fields are marked with a red asterisk in the dialog box.
   a. Enter a value, in bytes, that represents the maximum memory size for the database.
      
      *Note*: A blank or zero value indicates that no memory limit is set for the database.
   b. Select Advanced Driver Options if additional configuration is needed for the data service.
      * Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

4. **Catalog**: Specify a catalog name that contains all schemas belonging to the MDS data service.

5. **Summary**: Verify the settings and click Finish.

### Netezza Data Service

Use the following steps to create a data service for Netezza:

1. **Identification**: Enter the Name of the data service.

2. **Service Type**: Select Netezza from the service type list.

3. **Data Source**: At the Data Source dialog box, configure one or more drivers.
   a. Select Configure a native Netezza driver and update the following configurations. The required fields are marked with a red asterisk in the dialog box.
      * For Option 1, enter the name of the database server.
      * Enter the name of the database.
      * Enter the port number for the server.
      * For Option 2, specify the name of the Netezza DSN.
   b. Select Configure an ODBC driver and enter an ODBC DSN or a driver connection string.
c. If you have configured more than one driver, select a **Default driver**.

d. Select **Advanced Driver Options** if additional configuration is needed for the data service.
   - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

   **Note**: For information about advanced options, see the "Netezza Driver Reference" topic in the *SAS Federation Server Administrator's Guide*.

4. **Authentication Domain**: Select an **Authentication Domain** from the list of available domains.

5. **Summary**: Verify the settings and click **Finish**.

**ODBC Data Service**

You can create an ODBC data service ‘with native catalog support’ or ‘without native catalog support.’ The difference between the two is that a data service 'without native
catalog support requires that you specify a catalog name. Use the following steps to create an ODBC data service:

1. **Identification**: Specify the name of the new data service.

2. **Service Type**: Select one of the following options from the Service type list:
   - ODBC with native catalog support
   - ODBC without native catalog support

3. **Data Source**: In the Data Source dialog box, configure an ODBC driver to connect to the data source. The required fields are marked with a red asterisk.
   a. Specify either a DSN name or a driver connection string.

   ![Figure 6.5 ODBC Data Source Configuration](image)

   b. Select **Advanced Driver Options** to enter additional connection options for the data source.

   *Note*: Advanced options are specified as `<key>=<value>` pairs using a semicolon to separate pairs. For information about advanced options, see the "ODBC Driver Reference" topic in the *SAS Federation Server Administrator's Guide.*
4. **Catalog**: This step is only for ODBC without native catalog support. Specify a catalog name.

5. **Authentication Domain**: Choose one of the following options:
   - If authentication is required when using the data service, accept the default Database supports authentication and select an associated domain.
   - If authentication is not required for the data service, uncheck Database supports authentication.

**CAUTION:**
When creating a data service with or without authentication support, you cannot change properties of the data service to alter the behavior of authentication after setup has been completed. If the Database supports authentication option is selected for the data service, you can later edit the data service to change domains, but you cannot disable the check box for Database supports authentication. The same rule applies to a data service that is configured without authentication support. You cannot alter the properties of the data service later to select the Database supports authentication check box.

6. **Summary**: At the Summary dialog box, review the configuration and click Finish.

---

**Oracle Data Service**

Use the following steps to create a data service for an Oracle data source:

1. **Identification**: Specify the name of the new data service.
2. **Service Type**: Select Oracle from the Service Type list.
3. **Data Source**: At the Data Source dialog box, configure one or more drivers.
   a. Select Configure a native Oracle driver and update the following configurations. The required fields are marked with a red asterisk in the dialog box.
      - Specify the path to an Oracle connect identifier (for example, tnsnames.ora).
        
        **Note**: The path is an Oracle connect identifier as defined in tnsnames.ora or other naming method. A connect identifier can be a net service name or a database service name that resolves to a connect descriptor.
   b. Select Configure an ODBC driver, and specify either a DSN name or a driver connection string.
c. If you have configured more than one driver, select a Default driver.

5. **Catalog**: Specify a Catalog name.

5. **Authentication Domain**: Select Authentication Domain from the list of available domains.

6. **Summary**: At the Summary dialog box, verify the settings and click Finish.

### PostgreSQL

Use the following steps to create a data service for a PostgreSQL data source:
1. **Identification**: Enter the **Name** of the data service.

2. **Service Type**: Select **PostgreSQL** from the Service Type list.

3. **Data Source**: Configure one or more drivers. The required fields are marked with a red asterisk in the dialog box.
   
a. Select **Configure a native PostgreSQL driver** and configure one of the options below.
      
      - **Option 1**: Enter the name of the server, database, and associated port number.
      - **Option 2**: Specify the name of the DSN used to access the data source.

   b. Select **Configure an ODBC driver**, and specify either a DSN name or a driver connection string.

   c. If you have configured more than one driver, select a **Default driver**.

4. **Catalog**: Specify a **Catalog** name.

   - **Advanced Driver Options**: If additional configuration is needed for the data service.
     
     - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

     **Note**: See "SAS Federation Server Driver for PostgreSQL" in the *SAS Federation Server Administrator's Guide* for a list of available connection options.
5. **Authentication Domain**: Select an **Authentication Domain** from the list of available domains.

6. **Summary**: Verify the settings and click **Finish**.

*SAP Data Service*

Use the following steps to create a data service for SAP:

1. **Identification**: Enter the **Name** of the data service.

2. **Service Type**: Select **SAP** from the service type list.

3. **Data Source**: Configure one of the following items. The required fields are marked with a red asterisk in the dialog box.
   - **SAPGUI logical ID**: Specify an SAP Logon ID to use on a Windows 32-bit system.
   - **Application Server**: Specify the host name and system number of the application server. Use this option if load balancing is not in use.
   - **SAPRFC.INI logical name**: Specify the destination used in the SAPRFC.INI file. If working with the NetWeaver RFC library and a sapnwrfc.ini file, specify the destination that is in the SAPNWRFC.INI file.
   - **Message Server**: Specify the message server host, the name of the R3 system, and logon group that applies to the application servers used for load balancing.
Select **Advanced Driver Options** if additional configuration is needed for the data service.

- Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

  *Note:* For information about advanced options, see "SAS Federation Server Driver for SAP" in the *SAS Federation Server Administrator's Guide*.

4. **Catalog**: Specify a **Catalog** name.

   *Note:* Catalog is required for an SAP data service. If a catalog name is not specified, SAS Federation Server Manager creates a default logical catalog name with the same name as the data service.

5. **Authentication Domain**: Select an **Authentication Domain** from the list of available domains.

6. **Summary**: Verify the settings and click **Finish**.
SAP HANA Data Service

Use the following steps to create a data service for SAP HANA:

1. **Identification**: Enter the Name of the data service.
2. **Service Type**: Select SAP HANA from the service type list.
3. **Data Source**: At the Data Source dialog box, configure one or more drivers.
   a. Select **Configure a native SAP HANA driver** and update the following configurations. The required fields are marked with a red asterisk.
      - **Option 1**: Enter the name of the server, server instance, and associated port number.
      - **Option 2**: Specify the name of the DSN used to access the data source.
   b. Select **Configure an ODBC driver**, and specify either a DSN name or a driver connection string.

Figure 6.9  SAP HANA Data Source Configuration

   ![SAP HANA Data Source Configuration](image)

   c. If you have configured more than one driver, select a **Default driver**.
   d. Select **Advanced Driver Options** if additional configuration is needed for the data service.
      - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.
Note: For information about advanced options, see "SAS Federation Server Driver for SAP HANA" in the SAS Federation Server Administrator's Guide.

4. **Catalog**: Specify a Catalog name.

5. **Authentication Domain**: Select a domain from the list of available domains.

6. **Summary**: Verify the settings and click Finish.

**SAS Federation Server Data Service**

Use the following steps to create a data service that allows connectivity between federation servers:

1. **Identification**: Enter the Name of the data service.

2. **Service Type**: Select SAS Federation Server from the service type list.

3. **Data Source**: Configure the items listed below. Required fields are marked with a red asterisk.
   - **Server**: Specify the name of the federation server.
   - **Port**: Specify the port number for the federation server. The default port is 24141.
   - **DSN**: Specify a DSN for the federation server.
   - Select **Advanced Driver Options** if additional configuration is needed for the data service.
   - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

   Note: For information about advanced options, see the "SAS Federation Server Driver Reference" topic in the SAS Federation Server Administrator's Guide.
4. **Authentication Domain**: Select an **Authentication Domain** from the list of available domains.

5. **Summary**: Verify the settings and click **Finish**.

### SASHDAT Data Service

The SAS Federation Server Driver for SASHDAT (Driver for SASHDAT) is a write-only driver designed for use with Hadoop on a grid host, such as the SAS LASR Analytic Server. SAS LASR Analytic Server integrates with Hadoop by storing SAS data in the Hadoop Distributed File system (HDFS). See ‘SAS Federation Server Driver for SASHDAT’ in the *SAS Federation Server: Administrator’s Guide* for additional information.

Use the following steps to create a data service for SASHDAT:

1. **Identification**: Enter the **Name** of the data service.

2. **Service Type**: Select **SASHDAT** as the service type.

3. **Data Source**: Configure the items outlined below. Required fields are marked with a red asterisk.
   - **Grid Host**: specify the name of the grid host that has a running Hadoop NameNode.
   - **Install Path**: specify the path to the TKGrid installation on the grid host.
   - Select **Advanced Driver Options** if additional configuration is needed for the data service.
• Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

Note: For information about advanced options, see "SAS Federation Server Driver for SASHDAT" in the SAS Federation Server Administrator's Guide.

Figure 6.11  SASHDAT Data Source Configuration

4. **Catalog**: Specify a **Catalog** name.

5. **Authentication Domain**: Select a domain from the list of available domains.

6. **Summary**: Verify the settings and click Finish.

**Scalable Performance Data Server (SPD Server) Data Service**

To create a data service for Scalable Performance Data Server, use the **Generic without catalog support** data service template.

1. **Identification**: Specify the name of the new data service.

2. **Service Type**: Select **Generic without native catalog support** as the **Service type**.
3. **Data Source**: Specify SPDS for the driver type name, and enter a connection string.
4. **Catalog**: Specify a *catalog name*.

5. **Authentication**: If authentication is required, accept the default *Database supports authentication*, and select an associated *domain*.

6. **Summary**: Review the summary and click **Finish**.

---

**SQL Server Data Service**

Use the following steps to create a data service for SQL Server:

1. **Identification**: Enter the *Name* of the data service.

2. **Service Type**: Select *SQL Server* from the service type list.

3. **Data Source**: Configure the following items:
   - Specify a DSN name or a driver connection string.
   - Select **Advanced Driver Options** if additional configuration is needed for the data service.
   - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

   *Note*: For information about advanced options, see "SAS Federation Server Driver for SQL Server" in the *SAS Federation Server Administrator's Guide*.

4. **Authentication Domain**: Select a domain from the list of available domains.

5. **Summary**: Review the summary and click **Finish**.
**Teradata Data Service**

Use the following steps to create a data service for a Teradata data source:

1. **Identification**: Enter the **Name** of the data service.
2. **Service Type**: Select **Teradata** from the service type list.
3. **Data Source**: Configure one or more drivers as outlined below.
   a. Select **Configure a native Teradata driver** and specify a **server name** and **database name**. The server name is required.
   b. Select **Configure an ODBC driver**, and specify either a DSN name or a driver connection string.

   ![Teradata Data Source Configuration](image)

   Figure 6.15  Teradata Data Source Configuration

   c. If you have configured more than one driver, select a **Default driver**.
   d. Select **Advanced Driver Options** if additional configuration is needed for the data service.
      - Enter advanced options with `<key>=<value>` pairs using a semicolon to separate pairs. Use ASCII characters only when setting advanced options.

   *Note:* For information about advanced options, see the "SAS Federation Driver for Teradata" topic in the *SAS Federation Server Administrator's Guide*.

4. **Catalog**: Specify a **Catalog** name.
5. **Authentication Domain:** Select a domain from the list of available domains.

6. **Summary:** Review the summary and click **Finish.**

---

**Using the Generic Data Service Template**

If your data source does not appear in the service type list, you can use the Generic data service template to create a data service. There are two types of Generic data services that you can create:

- with native catalog support
- without native catalog support (requires catalog name)

Use the following steps to create a data service using the generic data service template:

1. **Identification:** Specify the name of the new data service.

2. **Service Type:** Select **Generic with or without native catalog support.**

3. **Data Source:** Specify a driver name and a connection string.
   
   *Note:* Options are specified as `<key>=<value>` pairs using a semicolon to separate pairs.

4. **Catalog:** (Generic without native catalog support only) Specify a **catalog name.**

5. **Authentication Domain:** If authentication is required, accept the default **Database supports authentication** and select an associated **domain.**

   **CAUTION:**
   
   When creating a data service with or without authentication support, you cannot change properties of the data service to alter the behavior of authentication after setup has been completed. If the **Database supports authentication** option is selected for the data service, you can later edit the data service to change domains, but you cannot clear the **Database supports authentication** check box. The same rule applies to a data service that is configured without authentication support. You cannot alter the properties of the data service later to select the **Database supports authentication** check box.

6. **Summary:** Review the summary and click **Finish.**

---

**Editing a Data Service**

Use the following steps to edit a data service:

1. Select a **federation server ⇒ data service** in the tree.

2. Use the icons located to the right of the **Summary** tab.

3. Select the item that corresponds to the operation that you want to perform.

   **Table 6.1 Data Service Editing Actions**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
</table>

---
Working with Data Source Names (DSNs)

About DSNs

DSNs are displayed only when logged on to a federation server. A DSN references a specific data service through which it connects, and defines how SQL security is enforced. It can be configured so that SAS Federation Server enforces SQL privileges defined for the data service. For additional information about DSN types and configuration options, see the SAS Federation Server Administrator's Guide.

DSN Types

You can create a standard DSN or a federated DSN, which is a collection of one or more standard DSNs.

- **Standard DSN**: A standard DSN is a single-service DSN created for a particular data service and is parented to that data service. The scope is limited to one data service and contains connection information, such as server name, port, path, or other connection options specific to a data service.

- **Federated DSN**: A federated DSN is a collection of one or more standard DSNs. Unlike the standard DSN, which is parented to a data service, the federated DSN is parented to the federation server itself, even if it contains DSNs from only a single data service. Federated DSNs can contain other federated DSNs.

Create a Standard DSN

Use the steps below to create a standard DSN. You can configure a DSN to use the FedSQL or DS2 dialects when using this procedure.

1. Select a **federation server** in the tree.
2. Select the **Data Source Names** tab and click the **New standard data source name** icon on the toolbar.
3. **Identification**: Specify the **Name** and **Description** of the new DSN.

Note: You cannot edit the BASE or SQL_LOG data services because these objects are created and owned by the system.
4. **Data Service**: Select the **Data Service** that is associated with the new DSN.

5. **Security**: Enable Federation Server SQL Authorization is selected by default. You can accept this option or disable it.

   *Note:* Disabling Federation Server SQL Authorization Enforcement allows for Native SQL dialect.

6. **Access**: Select the type of login required for the DSN: **Personal** login or **Shared** login.

   a. If you specify a Shared login, select a consumer group using the drop-down menu.

   *Note:* The Consumer group identifies the authorizations to use should the user exist in more than one group. Consumers are SAS Federation Server user accounts or groups. You should never use the actual shared login group as a consumer group in a DSN.

   *Note:* When shared login is selected on the DSN, SAS Federation Server appends the selected domain with an @shared login key, and verifies that data_source@<shared login key> exists in SAS Metadata as a valid authentication domain that also includes user account information.

   b. If both Personal and Shared logins are specified, select a credentials search order under **Access Order**.

   *Note:* By default, login credentials are searched in this order: Personal, Group, and Shared Login.

7. **Syntax**: Select **FedSQL dialect** or **DS2**.

8. **Options**: Specify optional parameters for the DSN and move on to the **Summary**.

   *Note:* If any of these fields are left blank, SAS Federation Server Manager uses the values that are set in the federation server configuration files.

9. **Summary**: Review the configurations and click **Finish**.

---

**Create a Federated DSN**

Since federated DSNs are parented to a federation server, they must be created from a federation server object. You cannot create a federated DSN under a data service. Use the steps below to create a Federated DSN.

1. Select a federation server object in the tree.

2. Select the **Data Source Names** tab and click the **New federated data source name** icon on the toolbar.

3. **Identification**: Specify the **Name** and **Description** of the new federated data source name.

4. **Members**: Select + to produce a list of data source names that are available for the Federated DSN.

5. Still on the **Members** step, select the DSNs to add to the Federated DSN.

   *Note:* Ensure that the child DSNs are not pointing to the same catalog, as this might result in a catalog conflict error. You cannot have duplicate catalog names within the connection.

6. **Security**: Enable Federation Server SQL Authorization is selected by default. You can accept this option or disable it.
Establishing DSN Permissions

Overview

A user must have CONNECT permission to establish connection with a DSN. Use the steps below to establish connect permissions for users and groups:

Establishing Connect Permissions

1. Select a federation server object in the tree.
2. Open the Data Source Names tab and select a DSN.
3. Select Connect Permission and click on + to open the Choose Members dialog box.
4. Enter a user or group name and click Search. You can also enter filter information to narrow the search.
5. Select the user (or group) and click Add. The user object now appears in the connect permission list.
   
   **TIP** You can add multiple users by holding down the Ctrl key while selecting a user or group object.

6. Click Close to exit the Add users and groups dialog box.
7. Select the user object and click Grant on the toolbar.

*Figure 6.16* Granting Connect Permission for a DSN
Chapter 7
Working with Federated Data

<table>
<thead>
<tr>
<th>Catalogs and Schemas</th>
<th>66</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working with Catalogs</td>
<td>66</td>
</tr>
<tr>
<td>Create a Catalog for Base SAS</td>
<td>66</td>
</tr>
<tr>
<td>Naming Conflicts and Unregistered Catalogs</td>
<td>66</td>
</tr>
<tr>
<td>Register a Catalog</td>
<td>67</td>
</tr>
<tr>
<td>Working with Schemas</td>
<td>67</td>
</tr>
<tr>
<td>Create a Schema</td>
<td>68</td>
</tr>
<tr>
<td>Assign Schema Ownership</td>
<td>68</td>
</tr>
<tr>
<td>Validate a Schema</td>
<td>69</td>
</tr>
<tr>
<td>Filter Schema Contents</td>
<td>70</td>
</tr>
<tr>
<td>Rename or Delete a Schema</td>
<td>71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Working with FedSQL Views</th>
<th>71</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>71</td>
</tr>
<tr>
<td>Invoker’s and Definer’s Rights View</td>
<td>71</td>
</tr>
<tr>
<td>Creating a FedSQL View</td>
<td>71</td>
</tr>
<tr>
<td>Creating a View from Native Sources</td>
<td>72</td>
</tr>
<tr>
<td>Modifying a View</td>
<td>73</td>
</tr>
<tr>
<td>Deleting a View</td>
<td>73</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Caching FedSQL Views</th>
<th>74</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>74</td>
</tr>
<tr>
<td>Prerequisites for Caching Views</td>
<td>74</td>
</tr>
<tr>
<td>Caching a FedSQL View</td>
<td>75</td>
</tr>
<tr>
<td>Configuring Advanced Properties for Cache</td>
<td>76</td>
</tr>
<tr>
<td>Disabling and Enabling Cache Tables</td>
<td>78</td>
</tr>
<tr>
<td>Deleting Cache Tables</td>
<td>78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Refreshing Cached Data</th>
<th>78</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>78</td>
</tr>
<tr>
<td>Manual Refresh</td>
<td>79</td>
</tr>
<tr>
<td>Scheduled Refresh</td>
<td>79</td>
</tr>
<tr>
<td>Cron Custom Scheduled Refresh</td>
<td>81</td>
</tr>
<tr>
<td>Working with the Schedule</td>
<td>83</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column and Row-Level Security</th>
<th>84</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column-Level Security</td>
<td>84</td>
</tr>
<tr>
<td>Granting Security on a Column</td>
<td>84</td>
</tr>
<tr>
<td>Row-Level Security</td>
<td>85</td>
</tr>
<tr>
<td>Apply Row Filters</td>
<td>85</td>
</tr>
<tr>
<td>Edit a Row Filter</td>
<td>86</td>
</tr>
<tr>
<td>Remove a Row Filter</td>
<td>87</td>
</tr>
</tbody>
</table>

| Defining User Functions for Row-Level Security | 87 |
Catalogs and Schemas

Working with Catalogs

All data services contain a catalog object. Databases that support catalogs contain native catalogs that are retrieved from the associated data source. If a database does not support catalogs, you can create a logical catalog when creating the data service, or by using the New Catalog function in SAS Federation Server Manager. See the SAS Federation Server Administrator's Guide for additional information about catalogs and schemas.

Create a Catalog for Base SAS

Catalogs and schemas are required for the BASE data service to expose data. Use the following steps to create a new catalog for a BASE data service:

1. Select a BASE data service in the tree and select New Catalog in the upper right corner.
2. At the New Catalog dialog box, enter a catalog name.
3. Click OK to save the catalog and exit.

When the new catalog is added to the data service, two tabs are displayed in the right pane: Summary and Authorizations. Select the catalog object in the tree and click to perform the following actions:

- Create a new schema.
- Test the connection.
- Rename the catalog.
- Delete the catalog.

Use the Authorizations tab to grant or deny permissions for the new catalog.

Naming Conflicts and Unregistered Catalogs

Normally, catalogs are registered by default when a data service is created. If a catalog is not registered, a warning icon is displayed next to each unregistered catalog in the tree and also in the Summary tab for each individual catalog object. Note: This warning is also displayed for SQL Server catalogs that have naming conflicts because they are registered to a different data service. These naming conflicts should be resolved, and the catalogs registered before attempting to set privileges for the table. Unregistered catalogs in the table will not reflect the new privileges. When you expand a data service in the tree, SAS Federation Server Manager checks the registration status of all the catalogs under the selected data service. If catalogs have not been registered or are already registered to a different data service, a warning is displayed.
Register a Catalog

If a data source supports native catalogs, a native catalog name is registered by default when a data service is created. Only SQL Server, Netezza, and SAS Federation Server use catalog registration. Use the following steps to register a catalog:

1. Select an unregistered catalog in the tree. Unregistered catalogs are denoted with this icon: 

2. Select the \ in the upper right corner of the tree, and select Registration Properties from the drop-down list.

3. At the Catalog Registration Properties dialog box, enter a name for the catalog and click OK.

Working with Schemas

You can create new schemas for the BASE, MDS, and SASHDAT data services only. Other data services retrieve their schemas from the associated data source.
TIP All Base SAS catalogs require a schema. A BASE data service cannot establish connection if a schema is not detected for each Base SAS catalog in the tree.

Note: Note that the MDS data service contains a default schema named SYSTEMINFO that is reserved for system use only. Therefore, you must create a new schema for your MDS data service.

Create a Schema

Use the following steps to create a schema for Base SAS, MDS, and SASHDAT data sources.

1. Select a federation server ➨ data service ➨ catalog that will contain the new schema.
2. With the catalog selected, click New Schema in the upper left corner of the tree.
3. Select New Schema from the drop-down menu.
4. At the New Schema dialog box enter the schema name and primary path for the schema.
   
   Note: You do not have to specify a primary path when creating a schema for MDS.
5. Under Owner, select User ID and click browse to open the Select User dialog box.
6. At the Select User dialog box, select a user name from the list and click OK.
7. Base SAS schema only: Select Advanced Options and enter additional schema options. Specify advanced options as key=value pairs using a semicolon to separate pairs from each other.
8. Click OK to return to the New Schema dialog box.
9. Click OK to create the schema.

Assign Schema Ownership

Use the following steps to assign or change a schema owner.

1. Select a federation server object that is associated with the data source containing the schema that you are updating.
2. Select the schema object by navigating through data service ➨ catalog ➨ schema.
3. Under the Summary tab to the right, select , Properties.
4. In the Schema Properties dialog box, select **User ID** and click ![open](image) to open the Choose User dialog box.

   *Note:* If SYSTEM is shown as the owner of the schema, you might encounter problems when creating FedSQL views. It is a best practice to designate a schema owner other than the SYSTEM user.

5. Select a user from the list and click **OK** to return to the Schema Properties dialog box.

6. The User ID field now reflects the new schema owner. You can also assign a primary path for the schema and enter advanced options in the Schema Properties dialog box.

7. Click **OK** twice to close the Schema Properties dialog box.

### Validate a Schema

Use the following steps to validate a schema:

1. Select a **federation server** ⇒ **data service** ⇒ **catalog** ⇒ **schema** objects in the tree.
2. With the schema selected, click ![open](image) on the **Summary** tab.
3. Select **Validate Schema** from the drop-down list.
4. In the Validate Schema dialog box, click **Validate**.

After validation, SAS Federation Server Manager displays a list of items that have passed or failed the validation test. Review each item and take corrective action if necessary. Here is an example of failed validation:
Filter Schema Contents

If a schema contains more than 1000 objects, you are given the opportunity to filter the results after opening the schema.

1. Select and open a schema in the tree.
2. The Filter Schema Contents dialog box displays if the schema is set to filter contents. Enter a keyword in the Filter field and click OK.

Note: You can also click OK to bypass the filter and display all schemas.

A list of schema objects containing the text entered in the Filter field are displayed.
To remove the filter select the schema object, right-click and select **View Filter** to remove filter properties.

1. Right-click the filter object and select **View Filter**.

2. At the **Filter Schema Contents** dialog box, remove or edit the **filter text** and click **OK**.

   **Note:** Filter preferences can be set in the user profile using **Application Options ➔ Federation Server Manager ➔ Maximum objects displayed in navigation tree.**

---

**Rename or Delete a Schema**

Click on the **Summary** tab, you can also rename or delete a schema. MDS does not allow you to rename or delete a schema if there are any active connections on the database.

---

**Working with FedSQL Views**

**Overview**

A federated data view (FedSQL view) contains the information required to access database sources and is stored separately from the data. By creating a view definition, you are storing only the instructions for where to find the data and how it is formatted, not the actual data. When you need to view information from multiple data sources or other source types, you can create a reusable FedSQL view to deliver data from multiple relational and non-relational data sources. You can also create a materialized view of the data by creating a cache from a FedSQL definer's rights view.

**Note:** Administration of FedSQL views and caches is not available when using a remote SAS Federation Server (FEDSRV) data service. You must always establish a direct connection to the remote SAS Federation Server to administer these objects.

**Invoker’s and Definer’s Rights View**

There are two types of FedSQL views:

- **Invoker’s Rights View**
  - The invoker’s rights view is accessed using the current user's authorization, credentials, and login information.

- **Definer’s Rights View**
  - A definer’s rights view is accessed using the schema owner's authorization, credentials, and login information and is always associated with a schema owner. A view must be set as definer's rights to cache.

**Creating a FedSQL View**

To create a new FedSQL view, navigate to the schema where the view will reside and follow these steps:

1. Select the schema where the view will reside.
2. Select Create FedSQL View on the toolbar.

3. Enter a name for the view and select a security setting to specify privileges for accessing the view.

   Note: If the view will be cached, you should use the definer's rights privileges (schema owner's privilege) to create the view. Only a definer’s rights view can be cached.

4. Click OK to continue. A new tab with the name of the view opens where you can build the SQL for the query. The CREATE VIEW statement is already created based on the selections made while defining the new view.

5. Using the Query tab, select additional data for the CREATE statement and move it to the query panel using the arrow.

6. For the SELECT statement, use Data on the Query tab to expand a table to expose columns.

7. Select a column and click the arrow to move columns to the SELECT statement.

8. After you have built a SELECT statement, add a FROM statement that includes one or more data sources. Position your cursor on a new line below the last-selected column and type FROM. Immediately following, select one or more tables from the available source tables on the Query tab.

9. Click Create View after you have finished defining the view.

   Note: If you receive an error message about duplicate naming, use the Settings tab to modify the name of the view.

**CAUTION:**

When creating a definer’s rights view, it is best to assign a schema owner. If the schema owner is not set or the owner is SYSTEM, a warning message indicates that the view might not operate correctly. You can proceed but be sure to set a schema owner when you are finished creating the view.

---

**Creating a View from Native Sources**

**Create a View from a Table**

Use the following steps to create a FedSQL view from a table:

1. Select a table in the tree.

2. Click on the toolbar.

3. Select New FedSQL View from Table from the drop-down menu.

4. The New FedSQL View from Table dialog box appears with a default view name. You can change the view name if desired.

5. Choose a security setting for the view and click OK.

6. A new table opens, displaying the query, which should include all the columns from the original table. You can make adjustments to the query as needed.

7. Click Create View.
Creating a View from a Native View

Use the following steps to create a FedSQL view from a native view:

1. Select a native view in the tree.
3. If necessary, change the view name, and select a security setting (for example, use the definer’s privileges when accessed).
4. Click OK to continue. A new tab opens, showing the Query tab that houses data and functions that are associated with the view. You can use this data to edit the query, such as Data Masking, Data Quality, and other settings for the FedSQL view.
5. Make the necessary changes and click Create View.

Modifying a View

You can modify a view’s security settings from Properties on the Options menu. However, you cannot modify the contents of the view (SQL statement or query) after the view is created. You must drop and re-create a new view with modifications.

Deleting a View

You can delete a view using the toolbar on the Summary tab. Deleting a view deletes both the FedSQL view and the cache definition that is associated with the view. Follow these steps to delete a view:

1. Click Delete on the toolbar located on the Summary tab.
2. Confirm by clicking the Delete button on the dialog box.
Caching FedSQL Views

Overview

SAS Federation Server uses FedSQL to enable users to cache data from a definer's rights view, creating a materialized view of the data. A materialized view is a snapshot of the target view from a specific point in time. Cache implementation requirements and other details related to working with cached data are outlined in the SAS Federation Server Administrator's Guide.

Data cache connections use a credential search order (CSO) of PERSONAL, SHARED. See Credentials Search Order for additional information.

Prerequisites for Caching Views

Definer's Rights Views and Cache

A definer’s rights view uses the credentials of the schema owner. When the view is executed, it uses the credentials of the user that created, or defined the view rather than the credentials of the current user. Therefore, only a definer’s rights view can be cached. When views are run, they access the catalogs that are referenced using the definer’s credentials, even if the user is not currently connected to that catalog. If a definer’s rights view is altered to an invoker view, the associated cache is dropped.

Changing a View from Invoker to Definer

If a view is an invoker’s rights view, cache operations are not active for the view. Use the following steps to change the view to a definer’s rights view:

1. Select the view that you want to cache and click .
2. Select Properties to open the FedSQL View Properties dialog box.
3. Select Use the definer's privileges when accessed and click OK.

Figure 7.5 Changing FedSQL View Security
Caching a FedSQL View

Overview
A FedSQL view must be created before you can create a data cache of the view. After you create a FedSQL view, you can create a cache using the Cache FedSQL view button on the Summary tab.

Figure 7.6 Caching a FedSQL View

Creating a Cache Definition
To create a cache, you must use a definer’s rights view. Use the steps below to create a cache definition:

1. Select a FedSQL view from the tree by selecting Data Service ➔ Catalog ➔ Schema ➔ View.

   CAUTION: Creating a cache with a table that is owned by a system user can potentially cause problems and is not recommended.

2. From the Summary tab, click Cache FedSQL View.

3. Click the browse button to select a schema for the cached view and enter a description for the cache.

4. Select Advanced Properties to set other properties for the cache, or you can perform this step after the cache is created.

5. Select Save Cache Definition.

   Note: You can also select Create Cache Table. Creating a cache table automatically saves the cache definition.

After the view is cached a message appears, confirming that the cache definition was saved. This indicates that you are currently using the view since a cache table was not created.
Creating a Cache Table
Use these steps to create a cache table from a cache definition that was previously saved:

1. Navigate to the view and select it to display the cache definition.
2. Select the Cache tab and click Create Cache Table.

Delete and Other Cache Actions
Using More options from the Cache tab, you can Disable or Delete a cache. To remove a cache, click and select Delete Cache from the drop-down list located on the cache tab. Choosing Disable Cache temporarily disables the cache but does not remove it from the server. To refresh a cache, click under Scheduled Refreshes at the bottom of the page.

Note: If a cache has a scheduled refresh and it is dropped, you must also delete the scheduled job. See “Viewing Job History” for additional information.

Configuring Advanced Properties for Cache
Select Open Cache Definition to edit or update advanced properties for a cache that has been saved. The cache definition function is a series of tabs that encompass the cache definition editor. Here is an explanation of each of the tabs:

General
The General tab shows the name of the view, the location of the cache tables, and a description of the cache. The name field is read only. You can change the location of the cache tables by selecting the browse button. There are two additional options on this tab:

- On creation or population failure, delete all cache tables ...FORCE: When you select this option, the system reverts to the original view when the cache fails to create or populate with data. This option is not selected by default.
- Refresh cache on server startup: Select this check box if you would like the cache to persist after restarting the server. This option is selected as the default for all MDS caches.

Table Options
The Table Options tab contains options for processing the data cache. By default, these options are blank.

- Bulk Load: Use Bulk Load to process large amounts of data. Enter an option or string in the Bulk load options field. The following example shows how to specify options for the log: BL_LOG="C:/TEMP/bulkload.log"
  BL_LOAD_REPLACE=yes
  Other bulk-load options are available.
  Note: Verify that your data source supports bulk loading because not all data services support this option. For example, BASE data sources do not support bulk load. See the SAS Federation Server Administrator's Guide for more information.
- Database commit level: sets a limit on the number of modified rows to commit at one time. This action affects transaction logging limits on the back-end database. This option overrides the ERRLIMIT option.
- Insert buffer size: sets a limit for the number of rows that can be inserted at one time. This action places a limit on a driver's row array size when inserting data.
• **Error limit**: sets a limit on the number of errors to allow before a statement stops inserting data.

• **CT Preserve**: controls how data types are mapped. The options for CT Preserve are outlined below.

**STRICT**
Data type mapping is disabled. The requested type must exist in the target database. Therefore, type promotion does not occur. If the type does not exist, an error is returned.

**SAFE**
Target data types are upcaled only if they do not result in a loss of precision or scale. When character encodings are changed, the new column size is recalculated to ensure that all characters are stored in the new encoding.

**FORCE**
FORCE is the default for all drivers. The best corresponding target data type is chosen, even if it could potentially result in a loss of precision or scale. When character encodings are changed, the new column size is recalculated to ensure that all characters can be stored in the new encoding.

**FORCE_COL_SIZE**
This option is the same as FORCE, except that the column size for the new encoding is the same as the original encoding. This option can be used to avoid column size creep. However, the resulting column might be too large or too small for the target data.

• **Other table options**: defines additional options for the cache table. Specify table options as `<key>=<value>` pairs using a space to separate pairs, for example, `DBCOMMIT=1000 INSERTBUF=100`. To impose exclusive locks to avert a user from inserting, updating, or deleting a BASE table, select **Specify other table options** and enter `locktable=exclusive`. SAS Federation Server Manager wraps the entry in braces so that it is formatted correctly for FedSQL. Here is an example `{option locktable=exclusive}`.

### Table Definition
Select the check box for **Customize table definition** to enable this tab. Once the content is enabled, you can change column types using the list of supported Column Types. Here are the column requirements for a cached view table definition:

• Column names must be identical.

• Columns must be in the same order.

• Columns must contain compatible data types. For example, `(n)(var)char` in the view must be `(n)(var)char` in the data cache. This rule applies to all data types.

Column names in the custom table definition are quoted by default. If Native Syntax is used, column names might not be quoted as expected. The information in View's Definition is read only because the cache view must remain consistent with the original FedSQL view.

**Note**: Using native syntax in the table definition might require adjusting the cache escape to use one of the following: `{CACHE}`, `{CACHE_CATALOG}`, `{CACHE_SCHEMA}`, or `{CACHE_TABLE}`. In addition, the escape should be quoted according to the case sensitivity setting of the destination database.

### Before
The BEFORE statement provides a way for a user to specify SQL that will be executed before the view data is cached. Multiple EXEC BEFORE statements are allowed, and will be executed in the order specified. IGNORE RC indicates that an
error from this statement will not fail the data cache refresh operation. Use New Statement to create an ordered list of FedSQL statements.

Note: If you change a catalog name after setting a clause, you must manually update the cache.

After
The AFTER statement provides a way for a user to specify SQL that will be executed after the view data cache is created and populated. Multiple EXEC AFTER clauses are allowed, and will be executed in the order specified. IGNORE RC indicates that an error from this statement will not fail the data cache refresh operation. Use New Statement to create an ordered list of FedSQL statements.

Cleanup
The optional CLEANUP statement provides a way for a user to specify SQL that will be executed when the data cache is removed. This might happen when a refresh operation has failed and the invalid cache must be removed. This would also occur when an old data cache table is removed as a result of a purge_cache operation, or when the cleanup thread times out and does an automatic cleanup. This clause is normally used when USING, EXEC BEFORE, or EXEC AFTER is in use and it generally reverses whatever was done. Multiple EXEC CLEANUP statements are allowed and statements are executed in the order specified. Use New Statement to create an ordered list of FedSQL statements.

**Disabling and Enabling Cache Tables**

A cache view or table can be disabled temporarily in the event that it has to be taken offline for any reason. Disabling a cache does not drop or delete a cache. The cache is just temporarily suspended while the users are rerouted to the original definer’s rights view on which the cache is built. When the cache is enabled, users are directed back to the original cache.

Use the Cache tab to disable and enable cache.

To disable a cache, click ![Disable Cache](icon) and select **Disable Cache** from the drop-down list.

To enable a cache, click ![Enable Cache](icon) and select **Enable Cache** from the drop-down list.

**Deleting Cache Tables**

Use the cache tab to delete a cache table and remove the associated definition. To remove a cache, click ![Delete Cache](icon) and select **Delete Cache** from the drop-down list.

**Refreshing Cached Data**

**Overview**

Use one of the methods below to refresh cached data:

- manual refresh
- scheduled refresh
- custom refresh
**Manual Refresh**

To perform a manual refresh of a cache table, select the **Cache** tab, and then **Open Cache Definition**. Use the **Refresh Cache** icon in the toolbar to refresh data in the selected cache table. You should refresh cache if you receive a message ‘Currently using old cache’ when opening a cached view.

![Manual Cache Refresh](image)

**Scheduled Refresh**

**Configuring a Scheduled Refresh**

Cache refreshes are scheduled using triggers that specify a schedule for refreshing the cache. This action can be performed as a one-time refresh or a regularly scheduled refresh. To schedule cache tables to refresh automatically, use **Scheduled Refreshes** located on the **Cache** tab at the bottom of a displayed cache.
To schedule a refresh of a cache table follow these steps:

1. Locate the cache table in the tree and click the Cache tab.

2. At the bottom of the panel, at Scheduled Refreshes, select on the toolbar. The Scheduled Refresh dialog box appears.

3. Specify the frequency of the job using the drop-down menu. Possible options are monthly, weekly, daily, hourly, minutes, once, or custom. The options in the Scheduled Refresh dialog box vary according to the frequency type selected.

4. To schedule a daily refresh with a start and stop time, perform the following steps:
   a. At Frequency, select Daily from the drop-down menu.
   b. Specify the interval in days as 1 and specify a start time and start date.
   c. Specify an end time and date, or select No end time and date.
   d. Click OK to schedule the job.

   Note: A monthly refresh created to occur on the 'last day of the month' is saved as a custom refresh that requires cron for editing.
Cron Custom Scheduled Refresh

Overview
Using cron expressions, you can define a custom schedule to refresh cached data. A cron expression is a string comprised of at least 6 fields separated by white space. You can set any command to run continually or at set intervals by populating fields with a combination of values and special characters. Fields can contain any of the allowed values, along with various combinations of the allowed special characters for that field. Cron expression fields and values are as follows:

<table>
<thead>
<tr>
<th>Position</th>
<th>Field Name</th>
<th>Required</th>
<th>Allowed Values</th>
<th>Allowed Special Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Seconds</td>
<td>YES</td>
<td>0-59</td>
<td>, - * /</td>
</tr>
<tr>
<td>2</td>
<td>Minutes</td>
<td>YES</td>
<td>0-59</td>
<td>, - * /</td>
</tr>
<tr>
<td>3</td>
<td>Hours</td>
<td>YES</td>
<td>0-23</td>
<td>, - * /</td>
</tr>
<tr>
<td>4</td>
<td>Day of month</td>
<td>YES</td>
<td>1-31</td>
<td>, - * ? / L W</td>
</tr>
<tr>
<td>5</td>
<td>Month</td>
<td>YES</td>
<td>1-12 or JAN-DEC</td>
<td>- * /</td>
</tr>
<tr>
<td>6</td>
<td>Day of week</td>
<td>YES</td>
<td>1-7 or SUN-SAT</td>
<td>- * ? / L #</td>
</tr>
<tr>
<td>7</td>
<td>Year</td>
<td>NO</td>
<td>empty, 1970-2099</td>
<td>- * /</td>
</tr>
</tbody>
</table>

Here are examples of some common cron expressions and their meanings:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 12 * * ?</td>
<td>Fire at 12:00 PM (noon) every day</td>
</tr>
<tr>
<td>Expression</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>0 15 10 ? * *</td>
<td>Fire at 10:15 AM every day</td>
</tr>
<tr>
<td>0 15 10 * * ?</td>
<td>Fire at 10:15 AM every day</td>
</tr>
<tr>
<td>0 15 10 * * ? *</td>
<td>Fire at 10:15 AM every day</td>
</tr>
<tr>
<td>0 15 10 * * ? 2012</td>
<td>Fire at 10:15 AM every day during the year 2012</td>
</tr>
<tr>
<td>0 * 14 * * ?</td>
<td>Fire every minute starting at 2:00 PM and ending at 2:59 PM, every day</td>
</tr>
<tr>
<td>0 0/5 14 * * ?</td>
<td>Fire every 5 minutes starting at 2:00 PM and ending at 2:55 PM, every day</td>
</tr>
<tr>
<td>0 0/5 14,18 * * ?</td>
<td>Fire every 5 minutes starting at 2:00 PM and ending at 2:55 PM, AND fire every 5 minutes starting at 6:00 PM and ending at 6:55 PM, every day</td>
</tr>
<tr>
<td>0 0-5 14 * * ?</td>
<td>Fire every minute starting at 2:00 PM and ending at 2:05 PM, every day</td>
</tr>
<tr>
<td>0 10,44 14 ? 3 WED</td>
<td>Fire at 2:10 PM and at 2:44 PM every Wednesday in the month of March</td>
</tr>
<tr>
<td>0 15 10 ? * MON-FRI</td>
<td>Fire at 10:15 AM every Monday, Tuesday, Wednesday, Thursday, and Friday</td>
</tr>
</tbody>
</table>

For more information and additional examples, see [http://www.quartz-scheduler.org/documentation](http://www.quartz-scheduler.org/documentation)

**Configure a Custom Refresh Interval**

Use the steps below to schedule a custom refresh interval for cache.

1. Select 🗓 on the Scheduled Refreshes toolbar.

2. Select **Custom** from the **Frequency** drop-down menu.

3. Enter your cron expression. The cron expression in the following example sets the schedule to fire at 10:15 am on the last day of every month:
4. Click OK to schedule the job.

**Working with the Schedule**

**Viewing Scheduled Jobs**

Click [Open Schedule](#) on the [Navigation Bar](#) to open the schedule and view active jobs.

From the [Schedule](#) tab, you can view or edit scheduled jobs, delete one or more jobs, and search for jobs.

**Viewing Job History**

Use the following steps to view job history:

1. Under [Schedule](#), select the job that you want to view.
2. Select [Job Information](#) from the toolbar at Scheduled Refreshes.
3. Select the [History](#) tab in the Job Information dialog box.
   - The History tab displays the start and end time, duration, and status of the selected job.
4. Click [Close](#) when you are finished viewing history.

**Deleting Scheduled Jobs**

Use the following steps to delete one or more scheduled jobs:

1. Select one or more jobs in the Schedule.
2. Click [Delete](#) on the toolbar.
Column and Row-Level Security

Column-Level Security

Column security in SAS Federation Server Manager is invoked by expanding the columns that are associated with a selected table. A new view opens that displays all of the columns. Each column is accompanied by two tabs that outline the details of the column, the Summary tab, and the Authorizations tab.

Summary tab
The Summary tab displays a summary of the column that shows the identification and details of the column.

Authorizations tab
The Authorizations tab displays user and group identities with associated permissions.

Granting Security on a Column

Use the following steps to grant security for a column:

1. Log on to a SAS Federation Server and navigate to a table by selecting data service catalog schema.

2. Select a data object in the navigation tree and click View Columns.
   
   Note: You can also right-click the data object and click the View columns button that pops up.

3. Select the Authorizations tab, select a column, and select a user.

4. Select the permissions that apply to the column object.

5. Click Grant or Deny.

Figure 7.11 Column-level Security
Row-Level Security

You can use row-level security to restrict access to certain types of information that require scrutiny, such as personally identifiable information (PII). By invoking row-level security in SAS Federation Server Manager, you can expand a table and define filters for the rows that are associated with the table. Use the Authorizations and Row Authorizations tabs to display or configure row access. Here is an explanation for each of these tabs.

Authorizations

Use the Authorizations tab to display user and group identities and their associated permissions. This tab also displays row filters that are currently in place for selected users and groups.

Row Authorizations

The Row Authorizations tab displays all of the row filters that currently apply to the table. Mixed values are shown if different filters are assigned to the selections. You can also delete filters from this tab. Row filters are available only on tables. When you create a row filter, the new filter replaces the previous filter for the selected user or group. You can apply multiple row-level filters to a table for different groups and users.

When working with row filters, the following conditions apply if the WHERE clause does not contain text or contains invalid syntax:

- Setting a row filter with no text is equivalent to removing the row filter.
- You cannot create a new row filter or update an existing row filter if the syntax is incorrect. The only resolution is to correct the syntax or click Cancel.

Apply Row Filters

Follow the steps below to create and apply row filters to a table:

1. Log on to a federation server object and navigate to a table by selecting data service ➔ catalog ➔ schema ➔ table.
2. Select the Data tab and click View Data to view data for the selected table.
3. Select the Row Authorizations tab to create a new filter.
4. Select Create New Filter to open the Filter dialog box.
5. On the Filter dialog box, you can enter a WHERE statement manually, or select the Clause Builder or User Function to build the WHERE statement.
   
   Note: See “Working with User Functions” for additional information.
6. Select the Users and Groups tab and perform the following actions:

   • Select add Users and Groups ➔ to add additional user or group objects to the filter.
   • Select the Assign Filter check box for each user that is affected by the filter.
7. Click **OK** when you are finished assigning filters.

**Edit a Row Filter**

Follow these steps to edit a row filter from the **Row Authorizations** tab:

1. Select the row or rows that you want to filter by placing a checkmark next to each.
2. Select **Edit Filter Properties** from the toolbar. The Filter dialog box displays, listing the selected filter.
3. Use the **Clause Builder** or **Vendor defined** functions to edit a filter and click **OK**.

*Note:* If you select a filter and there is an existing filter on the selected user or group, a warning is displayed, indicating that selecting the new filter(s) for these users or groups result in existing filter(s) being overwritten and deleted.
Remove a Row Filter

Follow these steps to remove a filter:

1. Open the Row Authorizations tab and select the filter that you want to delete. Click Clear Filter.

2. A warning message appears that prompts you to set a new permission for the user or group. Select a new permission and click OK.

Note: When selecting OK and the WHERE clause does not contain text or an otherwise invalid filter, the following conditions apply:

- OK is equivalent to Cancel if a new filter is being defined. No filter is applied, and Select permissions for selected users or groups are not automatically set to Grant if that had been necessary to apply a filter.
- If an existing filter is edited, OK is equivalent to deleting the filter. The existing filter is removed, and the user is prompted for setting the Select permissions for the selected users or groups.

Defining User Functions for Row–Level Security

Overview

You can use the “RLS Library and Library Reference” to identify the parameters required for each user function including a description of the value(s) returned by each user function. You can access the User Function dialog boxes directly from the toolbar in the Filter dialog box or from the Clause Builder.

Using the Clause Builder

Using the Clause Builder, you can build any type of condition from a simple WHERE statement to a more complex user function using RLS functions. The Clause Builder is accessible while creating a row filter or editing an existing row filter. Use the following steps to access the Clause Builder and set up filters.

1. Select the Data tab and click View Data to view data for the selected table or view.
2. Select the Row Authorizations tab.
3. Select Create New Filter to open the Filter dialog box.
4. Select the Clause Builder or Vendor Defined Function to build the WHERE statement.
5. Use the drop-down menu under Column to select a field from the source table.
6. Use the Operator list menu to select a qualifier.
7. In the Value field, enter information to qualify the statement. You can also use the list menu to select Use user function. See “About User Functions” for more information.
8. Click OK when you are finished.
Working with User Functions

About User Functions
SAS Federation Server provides various functions that return information about a user that is currently connected to a data source. These functions are referred to in SAS Federation Server Manager as User Functions. Since the results returned from each function contain information about the current user, User Functions can be valuable when constructing filters for row-level security. For example, if rows in a table should be returned based on the authentication domain of the currently connected user, you could use the DOMAIN() user function to accomplish this. Use the RLS Library and Library Reference to identify the values and formatting associated with each user function.

Insert User Function from Filter
You can select Use vendor defined function to insert a user function while building a WHERE statement. User function is also accessible through the Clause Builder.

Note: RLS filters with user functions are most valuable when the filter is assigned to one or more groups.

1. Select the Use vendor defined function \( f_0 \) from the filter dialog box.
2. Use the drop-down list in the User Function dialog box to select a user function.
3. Using the RLS Library, choose a user function that returns a value that can be compared to values stored in a column from your source table.

User Function Example
Consider a table with a User Name column that contains the name of the user that is permitted to view the corresponding row. The User Function that returns the name of the current user is current_user. When you create an RLS filter that compares the contents of the User Name column with the value returned by the current_user function, you are shown rows where the User Name column contains the name of the current user. This filter can be associated with many users, and it allows each user to view the filtered rows, without having to write a separate filter for each user.

Insert User Function from Clause Builder
Use the following steps to specify conditions using the Clause Builder.

1. Select Value and select User Function. The User Function dialog box appears.
2. Use the drop-down menu to select a user function.
3. You can also specify a reference table to look up a user function by selecting User Function to look up the value in a reference table.
4. Click OK when you are finished.

User Function Value from a Reference Table.
The Use user function to look up the value in a reference table option is helpful when a table or view exists that matches values returned with values stored. For example, a table contains the value returned from a user function and associates these values with the values stored in a column of the table to which RLS is applied. Consider this example:

You have a table, SALES_LEADS, containing a list of sales leads for your company. Each member of your organization's sales department is responsible for all the leads in one or more regions. Each region has an associated identification code that is stored in the SALES_LEADS table in the REGION_ID column. Members of the sales department
should be able to select only rows from SALES_LEADS that correspond with the regions with which the member is associated.

<table>
<thead>
<tr>
<th>LEAD_NAME</th>
<th>LEAD_ADDRESS</th>
<th>LEAD_PHONE</th>
<th>...</th>
<th>REGION_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead #1</td>
<td>...</td>
<td>...</td>
<td></td>
<td>S</td>
</tr>
<tr>
<td>Lead #2</td>
<td>...</td>
<td>...</td>
<td></td>
<td>NE</td>
</tr>
<tr>
<td>Lead #3</td>
<td>...</td>
<td>...</td>
<td></td>
<td>NE</td>
</tr>
<tr>
<td>Lead #4</td>
<td>...</td>
<td>...</td>
<td></td>
<td>W</td>
</tr>
</tbody>
</table>

Using a reference table that associates the REGION_ID column with each sales department members’ user name allows for flexible application of row-level security on the SALES_LEADS table.

<table>
<thead>
<tr>
<th>REGION_ID</th>
<th>USER_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Executive One</td>
</tr>
<tr>
<td>W</td>
<td>Executive One</td>
</tr>
<tr>
<td>NE</td>
<td>Executive Two</td>
</tr>
</tbody>
</table>

If Executive One changes regions, records need to be updated only in the SALES_LEADS_LOOKUP table in order to maintain appropriate row-level security on the SALES_LEADS table. A link between the reference table (SALES_LEADS_LOOKUP) and the target table (SALES_LEADS), such that RLS will update as the reference table updates, can be created by selecting the Use user function to look up the value in a reference table check box, and performing the following task.

If the user function is set to look up a value in a reference table, specify the source reference table and the columns to filter:

1. In the reference table field, use the browse button to navigate to the reference table or view. The Select Table dialog box appears and presents a list of data services. Drill down to the source table and select it. In the example, the reference table is SALES_LEADS_LOOKUP.
2. In the Column that the user function value appears in field, specify the column in the reference table that contains values corresponding to those returned by the user function. In the example, the column in the reference table (SALES_LEADS_LOOKUP) that matches the result of the current_user function is USER_NAME.
3. In the Column that the value to filter on appears in field, specify the column in the reference table that contains values mapped to the results of the user function, which will be used to filter the rows in the table to which RLS is being applied. In the example, the column in the reference table (SALES_LEADS_LOOKUP) that contains values to filter on in the target table (SALES_LEADS) is REGION_ID.
4. Click OK. The information is set in the Value field of the Clause Builder.
5. Click OK to add the values to the row filter.
Part 3

Advanced Topics

Chapter 8
Working with the SQL Console ................................................. 93

Chapter 9
Working with DS2 Dialect ....................................................... 97

Chapter 10
Data Quality and Cleansing Functions ................................. 99

Chapter 11
Data Masking ............................................................................ 109

Chapter 12
SQL Logging ............................................................................. 115
Chapter 8
Working with the SQL Console

SAS Federation Server Manager Console

About the Console

The SAS Federation Server Manager Console supports FedSQL and DS2 languages. The Console enables administrators to manage SAS Federation Server without navigating all of the user interfaces that SAS Federation Server Manager provides. You can connect to the Console using any DSN that you have created and submit SQL and DDL statements. The type of SQL and DDL statements executed depends on the DSN connection and what you are trying to accomplish. For example, you can use FedSQL information views to call specific information from the SAS Federation Server database. You can also use the Console to execute administration DDL to configure SAS Federation Server objects, and privileges. For a complete list of administration DDL, see the SAS Federation Server Administrator's Guide.

Working with the Console

Launching the Console

To launch the Console, select the icon below SAS Federation Server Manager home.
Figure 8.1 SAS Federation Server Manager Console

Selecting a Federation Server
Use the Server list menu to select a federation server. You might be prompted to log on to the server if you have not already done so. After establishing connection to the server, the DSNs associated with the selected server will load for the session. The ADMIN DSN is the default connection.

Changing the Connection
You can change the connection by selecting one or more DSNs under Connection.

1. Select Browse to open the Select DSN(s) dialog box.
2. At the Select Data Source Names dialog box, check each DSN that you want to use.
3. If you are using more than one DSN, use the drop-down menu at Select connection language to choose a language for the session. Depending on the DSN, you can choose FedSQL or DS2.
4. Select OK when you are finished.

Figure 8.2 SAS Federation Server Manager Console
Using More Than One DSN

If you are using more than one DSN for your console connection, you are creating a federated DSN for the session. When you choose multiple DSNs to include in the federated DSN, any dialect is allowed. However, when you set the dialect on the parent (federated) DSN, the dialects that are associated with the child DSNs are ignored. Because you are using data federation, you can no longer use native SQL for the dialect. The valid options are FedSQL and DS2.

**Dialects**

When using the Console, the DSN connection controls the dialect that is available for the session. The following table outlines possible DSN and dialect combinations:

<table>
<thead>
<tr>
<th>Selected DSN</th>
<th>Dialect Type</th>
<th>Driver Type</th>
<th>SQL Functions Displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMIN DSN</td>
<td>FedSQL</td>
<td>SYSCAT</td>
<td>FedSQL</td>
</tr>
<tr>
<td>Standard DSN</td>
<td>FedSQL, DS2, native</td>
<td>ODBC, vendor specific</td>
<td>FedSQL or native, if driver type=ODBC also shows ODBC functions.</td>
</tr>
<tr>
<td>Federated DSN</td>
<td>FedSQL</td>
<td>ODBC, vendor specific</td>
<td>FedSQL; Federated DSNs always use FedSQL dialect.</td>
</tr>
<tr>
<td>Multiple DSN</td>
<td>Various</td>
<td>ODBC, vendor specific</td>
<td>FedSQL, if driver type=ODBC also shows ODBC functions.</td>
</tr>
</tbody>
</table>

**Working with Information Views**

Using the console, you can access any information view for SAS Federation Server. You must connect to your server with the ADMIN DSN to use the information views. To display a list of information views, expand the ADMIN data service, SYSCAT catalog, and schema in the tree. You can also issue DDL statements against one or more information views.

1. Connect to the Console with the ADMIN DSN.
2. In the left panel, under Data, click to expand ADMIN data service ⇒ SYSCAT catalog ⇒ SYSCAT schema to expose the information views.

The following example returns a list of DSNs for a federation server object.

1. Log on to a federation server and open the Console.
2. Select the server that you want to query and select the ADMIN connection.
3. Enter the following statement in the Console: SELECT * FROM "SYSCAT"."SYSCAT"."DSN_CONTENT".
4. Click **Submit**. Console returns a list of configured DSNs for the selected federation server.

For additional information, including visibility rules that apply to information views, see the appendix for "Information Views" in the *SAS Federation Server Administrator's Guide*. 
Overview of DS2 on SAS Federation Server

DS2 is a SAS proprietary programming language that is used for advanced data manipulation. DS2 provides capabilities that are not available through SQL, such as scoring models. In addition, you can use DS2 code to run data quality functions on SAS Federation Server. DS2 is included with Base SAS and intersects with the SAS DATA step.

To invoke DS2, you must configure a DSN that uses the DS2 dialect and grant users CONNECT permission to the DSN. In addition, users must have EXECUTE permissions on DS2 objects, such as packages and threads, before any functions can be run against them.

DS2 objects inherit privileges from the server in the following order:

- SERVER
- (DATA) SERVICE
- CATALOG
- SCHEMA
- PACKAGE
- FUNCTION

To view the contents of DS2 programs, use the DESCRIBE PACKAGE or DESCRIBE THREAD commands. For more information about DS2, see the SAS DS2 Language Reference.

Create a DS2 DSN

To create a DSN that uses the DS2 dialect, follow these steps:
1. Select a federation server object in the tree and click the Data Source Names tab.
2. Click New Standard Data Source Name and enter a name and description for the data source at Identification.
3. Select a data service for the DSN.
4. Select the schemas that are available for the DSN.
5. Security should be enabled by default. Skip to Syntax.
6. Select DS2 as the language to use when communicating with this DSN.
7. Enter additional Options if required.
8. Review the Summary and click Finish.

DSN and DS2 Object Permissions

Assign the CONNECT and EXECUTE permissions to all users that will be using DS2. See Connect Permissions for DSN for the tasks.

1. Select a DS2 object in the tree and click the associated Authorizations tab.
2. Select the users or groups to assign permissions to. If they are not listed, click + to add users and groups to the list.
3. With the users selected, move to the Permissions list and select CONNECT, EXECUTE and any other permissions that the user might need.
4. Click GRANT in the upper right corner.

Note: See “Granting Permissions for Federation Server and Associated Objects” for additional information about object permissions.
Chapter 10

Data Quality and Cleansing Functions

About Data Quality on SAS Federation Server

Overview

Data Quality on SAS Federation Server is implemented through SAS Quality Knowledge Base (QKB) using FedSQL and DS2. The data quality methods use data quality rules from the SAS QKB to cleanse data. The rules, referred to as QKB definitions, are operation- and locale-specific. The data quality functions are exposed through a Memory Data Store (MDS) table with a reserved namespace, SYSPROC.DQ.

Use a SELECT statement to invoke each data quality method. Ensure that locale is always uppercased, as shown in the following example:

```
SELECT SYSPROC.DQ.DQUALITY.DQSTANDARDIZE {
  state,
  'State/Province (Full Name)',
  'ENUSA' } AS STANDARD_STATE
FROM employee
```

You can also use fully qualified column names in the SELECT statement, as shown in the following example:

```
SELECT SYSPROC.DQ.DQUALITY.DQSTANDARDIZE {
  "HR"."PAYROLL"."employee"."state",
  'State/Province (Full Name)',
  'ENUSA' } AS STANDARD_STATE
FROM employee
```
TIP When preparing your SQL statement, enclose SQL literals in single quotation marks. SQL literals are any numeric, character, string, date, or Boolean values that are not identifiers. This also includes literal arguments in FedSQL functions. Use double quotation marks when specifying identifiers such as catalog, schema, table, or column parts. Although not required, use double quotation marks if SQL identifier case sensitivity is an issue.

The following topics describe each data quality method that is delivered with SAS Federation Server.

**Standardization**

**DQSTANDARDIZE**

Standardization generates a preferred standard representation of data values. Standardization definitions are provided for character content such as dates, names, and postal codes. The available standardization definitions vary from one locale to the next. The return values are provided in the appropriate case, and insignificant blank spaces and punctuation are removed. The order of the elements in the return values might differ from the order of the elements in the input character values.

The value parameter for the `DQSTANDARDIZE` method accepts the following data types:

`nvarchar(256)|date|timestamp`

Here are sample SELECT statements for standardization:

```sql
SELECT SYSPROC.DQ.DQUALITY.DQSTANDARDIZE(
  state,
  'State/Province (Full Name)',
  'ENUSA'
) AS STANDARD_STATE
FROM employee
```

**Table 10.1  Results for Standard State**

<table>
<thead>
<tr>
<th>State</th>
<th>STANDARD_STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>North Carolina</td>
</tr>
</tbody>
</table>

```sql
SELECT SYSPROC.DQ.DQUALITY.DQSTANDARDIZE(
  "HR"."PAYROLL"."employee"."postalCode",
  'Postal Code',
  'ENUSA'
) AS STANDARD_POSTAL_CODE
FROM employee
```

**Table 10.2  Results for Standard Postal Code**

<table>
<thead>
<tr>
<th>Postal Code</th>
<th>STANDARD_POSTAL_CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>275130250</td>
<td>27513–0250</td>
</tr>
</tbody>
</table>
Matching

DQMATCH

Matching analyzes the input data and generates a matchcode for the data. The matchcode represents a condensed version of the character value. Similar strings receive identical matchcodes. You can specify a sensitivity value, ranging from 0–100, indicating the degree of similarity that should be applied to consider something a match. A sensitivity value of 100 yields more information, and 0 yields less. The default recommended sensitivity value is 85.

The value parameter for the DQMATCH method accepts the following data types:

nvarchar(256)|date|timestamp

Here are sample SELECT statements for matching:

```
SELECT SYSPROC.DQ.DQUALITY.DQMATCH ( 
    postalCode, 
    'Postal Code', 85, 
    'ENUSA' ) AS MATCH_POSTAL_CODE 
FROM employee
```

Table 10.3 Results for Match Postal Code

<table>
<thead>
<tr>
<th>Postal Code</th>
<th>MATCH_POSTAL_CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>275130250</td>
<td>ABC~$$$$</td>
</tr>
<tr>
<td>27513</td>
<td>ABC~$$$$</td>
</tr>
<tr>
<td>27540</td>
<td>DEF~&amp;&amp;&amp;&amp;</td>
</tr>
</tbody>
</table>

```
SELECT SYSPROC.DQ.DQUALITY.DQMATCH ( 
    "HR"."PAYROLL"."employee"."phone", 
    'Phone', 50, 
    'ENUSA' ) AS MATCH_PHONE 
FROM employee
```

Table 10.4 Results for Match Telephone Number

<table>
<thead>
<tr>
<th>Telephone Number</th>
<th>MATCH_PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>9195551212</td>
<td>ABC~$$$$</td>
</tr>
<tr>
<td>5551212</td>
<td>ABC~$$$$</td>
</tr>
<tr>
<td>1-9195551212</td>
<td>ABC~$$$$</td>
</tr>
<tr>
<td>202-555-0143</td>
<td>XYZ~$$$$</td>
</tr>
<tr>
<td>5550143</td>
<td>XYZ~$$$$</td>
</tr>
</tbody>
</table>
Pattern Analysis

DQPATTERN
Pattern analysis returns a simple representation of a character pattern based on a text string, which can be used for pattern frequency analysis in profiling jobs. Pattern analysis identifies words or characters in the input data column as numeric, alphabetic, non-alphanumeric, or mixed. The choice of pattern analysis definition determines the nature of the analysis:

* non-alphanumeric, such as punctuation marks or symbols
A alphabetic
M mixture of alphabetic, numeric, and non-alphanumeric
N numeric

Here is a sample SELECT statement for pattern analysis:

```sql
SELECT SYSYPROC.DQ.DQUALITY.DQPATTERN (State, 'Word', 'ENUSA') AS PATTERN_WORD
FROM employee
```

<table>
<thead>
<tr>
<th>Input</th>
<th>Word_Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>North Carolina</td>
<td>AA</td>
</tr>
<tr>
<td>Virginia</td>
<td>A</td>
</tr>
<tr>
<td>SC</td>
<td>A</td>
</tr>
</tbody>
</table>

Identification

DQIDENTIFY
Identification analysis returns a value that indicates the category of the content in an input character string. The available categories and return values depend on your choice of identification definition and locale.

Here are sample SELECT statements for identification analysis:

```sql
SELECT SYSYPROC.DQ.DQUALITY.DQIDENTIFY (Name, 'Field Name', 'ENUSA') AS IDENTIFY_FIELD_NAME
FROM employee

SELECT SYSYPROC.DQ.DQUALITY.DQIDENTIFY (email, 'E-mail (Country Identification)', 'ENUSA') AS IDENTIFY_EMAIL
```
Table 10.6  Results for Identification

<table>
<thead>
<tr>
<th>EMAIL</th>
<th>COUNTRY</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:P.Adams@mymail.ca">P.Adams@mymail.ca</a></td>
<td>Canada</td>
</tr>
<tr>
<td><a href="mailto:Joe.King@bleep.au.com">Joe.King@bleep.au.com</a></td>
<td>Australia</td>
</tr>
<tr>
<td><a href="mailto:joe.smith@trip.com.us">joe.smith@trip.com.us</a></td>
<td>United States</td>
</tr>
<tr>
<td><a href="mailto:xxoo@internet.es">xxoo@internet.es</a></td>
<td>Spain</td>
</tr>
</tbody>
</table>

Gender

DQGENDER

Gender analysis evaluates the name or other information about an individual to determine the gender of that individual. If the evaluation finds substantial clues that indicate gender, the function returns a value that indicates that the gender is female or male. If the evaluation is inconclusive, the stored procedure returns a value that indicates that the gender is unknown. The exact return value is determined by the specified gender analysis definition and locale.

Here is a sample SELECT statement for gender analysis:

```sql
SELECT SYSPROC.DQ.DQUALITY.DQGENDER (NAME,
     'Name',
     'ENUSA' ) AS GENDER_NAME
FROM employee
```

Table 10.7  Gender Analysis

<table>
<thead>
<tr>
<th>Name</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane Smith</td>
<td>F</td>
</tr>
<tr>
<td>Joe King</td>
<td>M</td>
</tr>
<tr>
<td>S. Adams</td>
<td>U</td>
</tr>
</tbody>
</table>

Case

Use case definitions to apply uppercase and lowercase lettering using context-sensitive rules. Case operates on character content, such as names, organizations, and addresses. You can specify one of three casing types: uppercase, lowercase, or propercase. When uppercase or lowercase is specified, the function applies Unicode uppercase or lowercase mappings to the characters in the input string. When propercasing is specified,
the function applies uppercase mappings to the first letter in each word and lowercase mappings to the remaining letters.

**DQLOWERCASE**
Here is a sample SELECT statement for lower casing:

```sql
SELECT SYSPROC.DQ.DQUALITY.DQLOWERCASE (name, 'Lower', 'ENUSA') AS LOWERCASE_PHONE
FROM employee
```

**DQUPPERCASE**
Here is a sample SELECT statement for upper casing:

```sql
SELECT SYSPROC.DQ.DQUALITY.DQUPPERCASE (name, 'Upper', 'ENUSA') AS UPPERCASE_PHONE
FROM employee
```

**DQPROPERCASE**
Here is a sample SELECT statement for proper casing:

```sql
SELECT SYSPROC.DQ.DQUALITY.DQPROPERCASE (name, 'Proper (Name)', 'ENUSA') AS PROPERCASE_NAME
FROM employee
```

<table>
<thead>
<tr>
<th>Name</th>
<th>DQ Method</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane Smith</td>
<td>DQLOWERCASE</td>
<td>jane smith</td>
</tr>
<tr>
<td></td>
<td>DQUPPERCASE</td>
<td>JANE SMITH</td>
</tr>
<tr>
<td></td>
<td>DQPROPERCASE</td>
<td>Jane Smith</td>
</tr>
</tbody>
</table>

**Parse**

**DQPARE**
Parsing segments a string into semantically atomic tokens.

The value parameter for the DQPARE method accepts the following data types:

- nvarchar(256)
- date
- timestamp

Here is a sample SELECT statement using DQPARE:

```sql
SELECT SYSPROC.DQ.DQUALITY.DQPARE (address, 'Address', 'Street Name', 'ENUSA') AS PARSE_ADDRESS_STREET_NAME
```
FROM employee

SELECT SYSPROC.DQ.DQUALITY.DQPARSE (  
    name,  
    'Name (Global)', 'Prefix',  
    'ENUSA' )  AS PARSE_NAME_PREFIX
FROM employee

Table 10.9 Results for Parsing

<table>
<thead>
<tr>
<th>Name</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ms. Jane Smith</td>
<td>Ms.</td>
</tr>
<tr>
<td>Mr. Joe King</td>
<td>Mr.</td>
</tr>
<tr>
<td>Mrs. Mary Moffet</td>
<td>Mrs.</td>
</tr>
</tbody>
</table>

Extraction

dqextract

Extraction definitions are used to extract specific entities or attributes from a text string. Extraction returns one or more extracted text values, or tokens, as output. For example, to extract a name prefix:

SELECT SYSPROC.DQ.DQUALITY.DQEXTRACT (  
    EXTRACT_COLUMN, 'NAME','NAME PREFIX',  
    'ENUSA')
FROM employee

To extract a name prefix with a given name, enter two function calls in your select statement:

SELECT SYSPROC.DQ.DQUALITY.DQEXTRACT (  
    EXTRACT_COLUMN, 'NAME','NAME PREFIX',  
    'ENUSA')

SELECT SYSPROC.DQ.DQUALITY.DQEXTRACT (  
    EXTRACT_COLUMN, 'NAME','GIVEN NAME',  
    'ENUSA')
FROM employee

Table 10.10 Results for Name Extraction

<table>
<thead>
<tr>
<th>Name</th>
<th>Output</th>
<th>prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr. James W. Church</td>
<td>PREFIX</td>
<td>Mr.</td>
</tr>
<tr>
<td></td>
<td>GIVEN</td>
<td>James</td>
</tr>
</tbody>
</table>
Data Quality Functions in SAS Federation Server Manager

Navigation

The data quality methods are installed with your SAS Federation Server. The data quality package and methods are displayed in the tree under a SYSPROC data service that is associated with your federation server object. Here is the hierarchy:

- SAS Federation Server
- SYSPROC data service
- SYSPROC catalog
- DQ Schema
- DQUALITY package
- Data quality methods

Working with Data Quality Methods

Using the Console

Follow these steps to launch the console and work with the data quality methods:

1. Select Open Console from the toolbar on the Home tab.

2. Using the drop-down menus on the toolbar, select a federation server and a connection for your data and include the SYSPROC data service.

3. At the bottom of the Select Data Source Names dialog box, use the drop-down menu to select a language (for example, FedSQL) and click OK.
4. Using the **Data** riser in the left pane, select **SYSPROC**.

5. Expand the following objects to expose the data quality methods: **SYSPROC ≫ SYSPROC ≫ DQ ≫ DQUALITY**.

6. Enter your SELECT statement and select a data quality method (for example, DQSTANDARDIZE) to move it into the editor:

   ```
   SELECT SYSPROC.DQ.DQUALITY.DQSTANDARDIZE
   ```

7. Append the statement with a custom SQL statement. The following example standardizes state names:

   ```
   SELECT CUST_STREET_STATE_NAME ,
   SYSPROC.DQ.DQUALITY.DQSTANDARDIZE (CUST_STREET_STATE_NAME , 'State/Province (Full Name)', 'ENUSA' ) AS STANDARD_STATE
   FROM DB.CUSTOMER_DATA.CUSTOMERS
   ```

   **Note:** Use uppercase when specifying a locale.

8. Click ➤ to send your code to the server. The results should look similar to the following:

   **Table 10.11 Results for Standard State**

<table>
<thead>
<tr>
<th>CUST_STREET_STATE_NAME</th>
<th>STANDARD_STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>North Carolina</td>
</tr>
<tr>
<td>VA</td>
<td>Virginia</td>
</tr>
</tbody>
</table>
Updating the QKB

SAS provides regular updates to the QKB. It is recommended that you update your QKB each time that a new one is released. For a listing of the latest enhancements to the QKB, refer to “What’s New in SAS Quality Knowledge Base.” The What’s New document is available on the SAS Quality Knowledge Base (QKB) product documentation page at support.sas.com. Either search on the product name or locate it in the product index. The updated software is available through the Downloads site.
Chapter 11
Data Masking

About the Data Masking Functions

Data masking in SAS Federation Server is a series of FedSQL functions that are accessed through the Console using the SYSCAT.DM.MASK function in a SELECT statement. Here is a brief description of each of the data masking functions.

ENCRIPT

The ENCRYPT function masks the values in a column by encrypting a single value using symmetric key encryption. Encrypted values cannot be decrypted if a KEY argument is not specified and the ENCRYPT_KEY package configuration option is not set.

\[
\text{SYSCAT.DM.mask('ENCRYPT', "value"}
\]

\* [,
  'ALG', 'AES/FIPS|AES|SAS002|BASE64|SAS004|SAS003|SAS001',
  'KEY', 'encrypt_key',
  'DETERMINISTIC', YES|TRUE|ON|1|NO|FALSE|OFF|0,
  'EXPAND_PREC', YES|TRUE|ON|1|NO|FALSE|OFF|0,
  'CASE', 'U|L',
  'STRIP', 'BLANK|UNICODESP|UNICODESPACE|ANY|ALL|WS'
\]*/ )

Note: The ENCRYPT function preserves the data type of the original column if the data type is character (for example, CHAR, NCHAR, VARCHAR). If the column is not a character data type, the output produces a binary data result.

DECRYPT

The DECRYPT function unmasks the values in a column by decrypting a previously encrypted value using symmetric key encryption. The DECRYPT rule returns NULL.
if a KEY argument is not specified and the ENCRYPT_KEY package configuration option is not set.

```
SYSCAT.DM.mask('DECRYPT', "value"
/*[, 'ALG', 'AES/FIPS|AES|SAS002|BASE64|SAS004|SAS003|SAS001',]*/ )
```

**HASH**
The HASH function masks the values in a column by hashing a single value into a fixed-length hash digest or HMAC string. The HASH function is not reversible.

```
SYSCAT.DM.mask('HASH', "value"
/*[, 'ALG', 'MD5|SHA256',
'CASE', 'U|L',
'KEY', 'encrypt_key']*/ )
```

**TRANC**
The TRANC function masks the values in a column by transliterating characters from an input string to characters in an output string. Ensure that the mapped result is “lossy” (many instances of mapping multiple input character values to a single output character value) to prevent inference of the original value.

```
SYSCAT.DM.mask('TRANC',
/*Expression*/)
```

**RANDOM**
The RANDOM function masks the values in a numeric column by replacing them with uniformly distributed pseudo-random numbers. The RANDOM function is not reversible.

```
SYSCAT.DM.mask('RANDOM',
/*Expression*/)
```

**RANDATE**
The RANDATE function masks the values in a date column by replacing them with pseudo-random date values.

```
SYSCAT.DM.mask('RANDATE',
/*Expression*/)
```

**RANSTR**
The RANSTR function masks the values in a column by replacing the values with random strings. Strings are generated by an algorithm that uses characters from the source string in the generation process, adding padding characters if necessary. Padding is placed to the left of the string unless RIGHT is specified.

```
SYSCAT.DM.mask('RANSTR',
/*Expression*/)
```

**RANDIG**
The RANDIG function masks the numeric values in a column by replacing digits with strings of random digits. Strings are generated by an algorithm that uses digits that are derived from the base number system of the source value, adding padding digits if necessary. Padding is always to the left of digits.

```
SYSCAT.DM.mask('RANDIG',
/*Expression*/)
```
For additional information related to data masking, see the *SAS Federation Server: Administrator's Guide.*

---

### Displaying Data Masking Functions in the Console

To access the data masking functions in SAS Federation Server Manager, follow these steps:

1. Select a federation server in the navigation tree and log on if necessary.
2. Select **Console** located in the **Navigation Bar**.
3. In the Console, select a server from the drop-down menu and select the **ADMIN** DSN (default).
4. In the left pane, select **Functions** and expand the **FedSQL** directory.
5. Select the **System** folder and expand it to display the data masking functions.

*Figure 11.1  Data Masking Functions*

See “Data Masking” in the *SAS Federation Server: Administrator's Guide* for a list of arguments for each data masking function.

---

### Data Masking in a FedSQL View

You can add data masking functions when you create a FedSQL view. To create a FedSQL view and mask the data:

1. Log on to a federation server and navigate to a native table: **Federation Server ➔ Data Service ➔ Table**.
2. Select **New FedSQL View from Table** located on the **Summary** tab. Ensure that **Use the definer’s privileges** is selected and click **OK** to create the view.
3. Using the left panel, select **Functions** ⇒ **FedSQL** ⇒ **System** to expose data masking functions.

4. Select a data masking function from the list and click ‬ to move the ENCRYPT SQL to the statement on the right.

5. Configure the FedSQL view to include your data masking function. You can place the masking function anywhere in your SELECT statement. Here is an example:

   ```sql
   SELECT
       SYSCAT.DM.MASK (
           'ENCRYPT',
           A."id" ) AS "id_ENCRYPT",
       A."gender",
       A."birthdate",
       A."maiden_name",
       A."lname",
       A."fname"
   FROM
       "RLS_CAT"."RLS_Schema"."PERSONAL_DATA_SAMPLE" AS A
   ```

   **Figure 11.2 Creating a FedSQL View with Masked Social Security Numbers**

6. Select **Create View**. Here is the view that shows the encrypted Social Security Numbers:
Data Masking Examples

About the Examples

In the examples that follow, an ‘Employees Data’ table is used to demonstrate the ENCRYPT and HASH data masking functions.

Encrypt Function

The Employees table contains salary data that needs encryption. The SELECT statement contains the ENCRYPT function configured to encrypt the salary and output to a new column titled ENCRYPT. The SALARY column is included only for comparison purposes. This is the syntax:

```
SYSCAT.DM.MASK ('ENCRYPT',
A."SALARY" ) AS "ENCRYPTED",
```

Here is the result:

Figure 11.4  Data Masking Encryption Output
**HASH Function**

The HASH function hashes a single value into a fixed-length hash digest or HMAC string. In this example, PHONE NUMBER is hashed and is output to a new column named ‘HASH_PHONE’.

```
SYSCAT.DM.MASK ('HASH', A."PHONE_NUMBER" ) AS "HASH_PHONE"
```

**Figure 11.5  Data Masking HASH Output**

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
<th>EMAIL</th>
<th>PHONE NUMBER</th>
<th>HASH PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>198.0</td>
<td>Donald</td>
<td>O'connel</td>
<td>DOCO</td>
<td>650.507.98</td>
<td>860DBC7F24F36...</td>
</tr>
<tr>
<td>199.0</td>
<td>Douglas</td>
<td>Grant</td>
<td>DORAN</td>
<td>650.507.98</td>
<td>47F920086F3C5...</td>
</tr>
<tr>
<td>200.0</td>
<td>Jennifer</td>
<td>Whalen</td>
<td>JWHALEN</td>
<td>515.123.44</td>
<td>22F20D6278AA8...</td>
</tr>
<tr>
<td>201.0</td>
<td>Michael</td>
<td>Harstel</td>
<td>MHARTST</td>
<td>515.123.55</td>
<td>02436CE7F850A...</td>
</tr>
<tr>
<td>202.0</td>
<td>Pat</td>
<td>Fey</td>
<td>PFAY</td>
<td>603.123.66</td>
<td>243D13510F59...</td>
</tr>
<tr>
<td>203.0</td>
<td>Susan</td>
<td>Mevris</td>
<td>SMevris</td>
<td>515.123.77</td>
<td>821CF844571C7...</td>
</tr>
<tr>
<td>204.0</td>
<td>Hermann</td>
<td>Beer</td>
<td>HBAER</td>
<td>515.123.88</td>
<td>8DC19D993FF98...</td>
</tr>
<tr>
<td>205.0</td>
<td>Shalley</td>
<td>Higgins</td>
<td>SHIGG</td>
<td>515.123.90</td>
<td>1681E48A2DDD...</td>
</tr>
</tbody>
</table>
Chapter 12
SQL Logging

About SQL Logging

SQL Logging is the ability to view SQL statements and DS2 functions submitted to SAS Federation Server. With SQL logging enabled, you can view information such as the user who submitted the request accompanied by a breakdown of the request into prepare, execute and cursor phases. By using SQL logging, an administrator can easily determine what users are accessing the system, when they were connected, and what work was performed. For additional details about SQL logging, see the *SAS Federation Server: Administrator’s Guide*.

Enable SQL Logging

Use the following task to enable logging for a server session. Any parameters set for SQL logging during the session revert to the default configuration settings upon restart of the federation server. The information captured here does not affect regular server logging which is set within the `dfs_log.xml` configuration file on SAS Federation Server.

1. Select a federation server object in the tree.
2. Click \(^{1}\) in the upper left corner and select Properties from the drop-down list.
3. At the Federation Server Properties dialog box, click the SQL Log tab.
4. Click to select **On. Log SESSION transaction**, then select the events to record. See “SQL Logging Transactions” for a brief description of the behavior for each of these transactions.

5. Click **OK** to accept the changes and close Federation Server Properties.

**Figure 12.1  Federation Server Properties: SQL Log**

---

**Viewing the SQL Log**

**Overview**

SQL Logging status is displayed under the **Summary** tab of the selected federation server object. SQL Logging must be active for the federation server that you are working with, and you must be logged in to view logging results.

**Viewing the Reports**

SQL Log reports are accessed through the **Summary** tab of SAS Federation Server properties. To access the reports:
1. Select a federation server object in the tree and log on.

2. Select Log, located under Home in the left navigation menu. You can also select Open SQL Log using More options.

3. After opening the Log, use the drop-down list to select a report to view:
   - Individual Requests
   - Summarized Requests
   - User Report

**About the Individual Requests Report**

This detailed report shows complete SQL statements submitted by users within the previous hour. The data is sorted by start time in descending order. Total elapsed time is recorded in milliseconds. The following list shows some of the columns reflected in the default report.

- User ID
- Requests
- Statement ID
- Start Time
- Host Name and Port

To extend the time interval for this report or set additional parameters, use the Filter options located next to the reports selection list. You can also customize the report to add or remove columns. Select to the right of the last column of the report.

Additional information for each report entry is contained in the following tabs listed beneath the item selected in the report.

**Details**

Lists details of the selected request or transaction.

**Request**

Displays the statement that was used for the selected request.

**Plan**

Specifies the execution plan that the underlying database uses to execute the SQL statement. It appears in XML format on the screen. If there is a problem with the XML, such as truncation, the message Invalid Plan is displayed until the problem is fixed.

**Cache Access**

Displays data if a cache view was accessed. The Cache Access column reflects a status of true if cache data was used and false if the request did not access any cached views.

**Summarized Requests Report**

The rows in the table are grouped based on the hash key returned from the server. The hash key in the server hashes to the same value when the submitted SQL is identical. This table is sorted with the most recent submission at the top. The default report includes the following columns:

- Requests
- Number of Requests
- Last Submitted
- Mean Request Lifetime (s)
- Mean Cursor lifetime (s)
- Cache Access
- Mean Work Time (s)

To extend the time interval for this report or set additional parameters, use the Filter options located in the upper right corner of the window. You can also customize the report to add or remove columns.

Additional information for each report entry is contained in the following tabs listed beneath the item selected in the report.

**Details**
Lists details of the selected statement or transaction.

**Request**
Displays the SQL used for the selected statement or transaction.

**Plan**
Specifies the SQL execution plan that the underlying database uses to execute the SQL statement. It appears in XML format on the screen. If there is a problem with the XML, such as truncation, the message Invalid Plan is displayed until the problem is fixed.

**Cache Access**
Displays data in this view if any of the SQL Statements accessed a cache view. Cache is reflected as 'true' in the Cache Access column of the table.

**User Report**

The SQL user report shows the same information that is displayed in the SQL Statements Table, but it is organized in a hierarchical manner. The hierarchy is: user ⇒ session ⇒ connection ⇒ SQL statement. The default report includes the following columns:

- Transaction
- Details
- Request Lifetime (s)
- Data (kb)
- Cache Access
- Start Time

To extend the time interval for this report or set additional parameters, use the Filter options located in the upper right corner of the window. You can also customize the report to add or remove columns.

Additional information for each report entry is contained in the following tabs listed beneath the item selected in the report.

**Details**
Lists details of the selected statement or transaction.
Request
Displays the SQL used for the selected statement or transaction.

Plan
Specifies the SQL execution plan that the underlying database uses to execute the SQL statement. It appears in XML format on the screen. If there is a problem with the XML, such as truncation, the message Invalid Plan is displayed until the problem is fixed.

Cache Access
Displays data in this view if any of the SQL Statements accessed a cache view. Cache is reflected as 'true' in the Cache Access column of the table.

Customizing the Reports

Setting Filters for Logging
When you first open SQL Log, a Filter status message is displayed, showing the current filters in effect for the session. You can set filters for each report using the filter configuration panel to the right of individual reports.

Using the Filter option on the right side of the SQL Log panel, you can set filters to limit how much information is queried from the server.

1. Click Filter to display options for SQL logging and configure one of the following options:
   • Select Last and enter a number of Minutes, Hours, Days, or Weeks to filter results for a specific time frame.
   • Select Start and enter a start time and date, and an end time and date, to filter results for a specific period of time.
   • Select Show all to display everything. Note that selecting this option can impede performance.
   • Set a number for Maximum Rows Returned to prevent potential long-running queries. The default setting for this option is 5000.

2. To save your selections, click OK.

Figure 12.2 SQL Logging Filters
Exporting Report Data

You can export report data for SQL Logging to a CSV file, but you must have permissions to write to the folder that you are exporting to. To export report data to a CSV file:

1. Open a report and select Export in the upper right corner of the screen.
2. The system creates a report named SQLLog_Export.csv that is placed at the bottom of the screen. Open the report to format and save.

SQL Logging Transactions

The following table shows the ARM transactions that are captured in SAS Federation Server Manager. When SQL Logging is enabled, information in each of the transactions is captured. Following the table is a brief explanation of each of the transactions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Associated Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>SESSION</td>
<td>Session transaction</td>
<td>Perf.ARM.FederationServer.Session.Transaction.SESSION</td>
</tr>
<tr>
<td>DBC</td>
<td>Database connection</td>
<td>Perf.ARM.SQLServices.Connection.Transaction.DBC</td>
</tr>
<tr>
<td>DBTRAN</td>
<td>Database transaction</td>
<td>Perf.ARM.SQLServices.Connection.Transaction.DBTRAN</td>
</tr>
<tr>
<td>SQL</td>
<td>SQL statement</td>
<td>Perf.ARM.SQLServices.Statement.Transaction.SQL</td>
</tr>
<tr>
<td>Prepare</td>
<td>SQL Statement</td>
<td>Perf.ARM.SQLServices.Statement.Prepare</td>
</tr>
<tr>
<td>Execute</td>
<td>SQL Statement</td>
<td>Perf.ARM.SQLServices.Statement.Execute</td>
</tr>
<tr>
<td>CURSOR</td>
<td>SQL Statement</td>
<td>Perf.ARM.SQLServices.Statement.Transaction.CURSOR</td>
</tr>
<tr>
<td>Fetch Scroll</td>
<td>SQL Statement</td>
<td>Perf.ARM.SQLServices.Statement.FetchScroll</td>
</tr>
<tr>
<td>SetPos</td>
<td>SQL Statement</td>
<td>Perf.ARM.SQLServices.Statement.SetPos</td>
</tr>
<tr>
<td>BulkOps</td>
<td>SQL Statement</td>
<td>Perf.ARM.SQLServices.Statement.BulkOperations</td>
</tr>
<tr>
<td>Fetch</td>
<td>SQL Statement</td>
<td>Perf.ARM.SQLServices.Statement.Fetch</td>
</tr>
</tbody>
</table>
SESSION
(Session Transaction) A session transaction starts when a user initiates a server session.

DBC
(Database Connection) A database connection transaction is a child object of the SESSION transaction. A database connection begins when a user connects to a data source and ends when the user disconnects from the data source.

DBTRAN
(RDBMS Transaction) DBTRAN is the actual database transaction. It is a child object of the DBC transaction. A DBTRAN transaction begins with an established driver connection, or when a previous transaction is committed or rolled back, and a new one begins. DBTRAN records are written to the log only if AUTOCOMMIT is set to OFF. The DBTRAN transaction stops when AUTOCOMMIT is set to ON or when a COMMIT or ROLLBACK command is issued. SQL statements can span DBTRAN transaction boundaries.

SQL
(SQL Statement) SQL is a logical transaction. It encapsulates a series of activities that are related to one SQL statement. It is a child object of a DBC transaction. An SQL transaction starts when a user issues an SQL statement. Regardless of the statement type (DQL, DML, or DDL), the SQL transaction stops when the statement is either closed or unprepared. Subsequent executions of the same statement are recorded under the same SQL transaction, even if the statement is a DQL and the result set associated with it is closed.

Prepare
The Prepare transaction measures the Prepare phase of an SQL statement. It is a child object of an SQL transaction. The Prepare transaction starts when a user Prepares an SQL statement and stops when the call to prepare returns.

Execute
The execute transaction measures the Execute phase of an SQL statement. It is a child object of the SQL transaction. The Execute transaction starts when a user executes an SQL statement and stops when the call to execute returns.

CURSOR
CURSOR is a logical transaction. CURSOR is a child object of an SQL transaction and it encapsulates all the operations executed in a cursor, including reading, positioning, and updates. The CURSOR transaction starts when the Execute transaction finishes. It stops when the cursor is closed. All operations on the same result set belong to the same CURSOR transaction.

Fetch, Fetch Scroll
The FETCH transaction is a child object of the CURSOR transaction. The FETCH transaction has an Execute transaction as its predecessor. It is started when a user issues the first fetch on a result set using Fetch or Fetch Scroll. It stops when the call to Fetch or Fetch Scroll returns.

SetPos
The SetPos transaction is a child object to a CURSOR transaction. The SetPos transaction has an execute transaction as its predecessor. It is started when a user issues a SetPos call and stops when the call returns.

BulkOperations
The BulkOperations transaction is a child object of a CURSOR transaction. The BulkOperations transaction has an Execute transaction as its predecessor. It is started when a user issues a call to BulkOperations and stops when the call returns.