# SAS® Viya™ 3.3 Administration: Data

## Data Administration: Overview

This document provides instructions for administrative tasks such as adding caslibs and loading data. This document assumes that you are familiar with the data and caslib concepts that are explained in *SAS Cloud Analytic Services: Fundamentals*.

Use the interface that best meets your needs. Here are suggestions:

- To manage caslibs and CAS tables interactively, use the Data area in SAS Environment Manager. See "Data Administration: How to (SAS Environment Manager)" on page 1.
- If SAS Environment Manager is not deployed, use CAS Server Monitor to manage caslibs interactively. See "Data Administration: How to (CAS Server Monitor)" on page 10.
- To programmatically manage CAS data, use the Tables Action Set. To get started, see *SAS Viya Quick Start*.

An in-memory table is loaded from the physical source that is associated with a caslib. Each in-memory table should always be loaded from the same corresponding physical source file, in the same caslib. Loading data from one caslib into another can introduce ambiguity about the physical source for an in-memory table, yielding unexpected results in SAS Visual Analytics.

**Note:** If you use SAS Environment Manager exclusively to load data, this requirement is met automatically.

## Data Administration: How to (SAS Environment Manager)

### Introduction

The SAS Environment Manager Data area enables you to view and manage tables, caslibs, user-defined formats, and SAS Viya servers. With the Data area you can perform the following tasks:

- examine available caslibs, loaded tables, and the unloaded tables that are assigned to each caslib
- add new caslibs
- load and unload tables

- delete loaded and unloaded tables and caslibs
- examine the properties for a server, stop a server, and view and terminate sessions
- manage the libraries that are assigned to a server, including the library tables and columns
- import, add, and administer global user-defined formats

Depending on user permissions and privileges, assignments can be viewed and managed for the different tables, caslibs, and servers.

The following instructions explain how to manage caslibs and tables using SAS Environment Manager.

## Navigation

From **SAS Environment Manager**, select **Data**.

At the top of the **Data** page, from the **View** drop-down list, select one of the following views:

**Loaded tables**
lists all in-memory, global-scope tables that you are authorized to see. This view of tables does not include unloaded tables. In this area you can view the column information for each table, view and edit authorizations, unload a table, delete a table, or view the table properties. Specific actions are granted based on permission settings. This is the default view.

**Libraries**
lists all global and personal caslibs that you are authorized to see. Use this view to add and manage caslibs.

In this area you can create a new library or edit an existing library. You can view tables, import data or tables into the library, delete a library, and view library properties. You can also view and edit authorization settings for the library.

You can drill into a caslib to see its tables, both loaded and unloaded. You can perform table-specific tasks, such as view column information for a loaded table, import data, load and unload tables, delete a table, or view the tables properties. You can also view and edit authorization settings for a table. Specific actions are granted based on permission settings.

For information about the different types of predefined caslibs, see “Predefined Caslibs” on page 22.

**Servers**
lists all CAS servers. In this area you can view the running state of a server, content with respect to libraries and tables, and server properties. You can also view and terminate sessions for the server.

Properties for a server enable you to view and modify server settings including: the basic properties and state of a server, nodes for the server, paths list, Superuser role membership, and caslib management privileges.

This view is relevant to data administration if you need to assume Superuser status. When you assume Superuser status, you can stop the server.

**User-defined formats**
lists SAS provided format libraries and global user-defined formats in each library. In this area you can add new user-defined formats to one of the SAS provided format libraries. You can then edit, copy, delete, and view properties for the user-defined formats. You can also import formats from item stores that are created with the FMTC2ITM procedure.

**Note:** Session formats are not shown in the **User-defined formats** area.

When you select a table, caslib, server, or user-defined format, available options for that item are accessible from a pop-up menu or from the icon task menu. You can also double-click to open a table, caslib, or server.

Here are some additional navigation features:

- You can modify the display and sort order of columns in the display by right-clicking a column heading and selecting from the available sort options. You can also reorder columns by selecting and dragging a column to the left or right.

  You can customize which columns are displayed in the Data area by selecting and selecting **Manage Columns**. The Columns window contains all available columns for the currently selected table, caslib, format, or server. From here you can choose to display or hide columns in the Data area. On this window you assign table columns to the **Hidden columns** list or to the **Displayed columns** list.

  These preferences persist until the end of your SAS Environment Manager login session.
- For loaded tables, libraries, and servers, you can filter the current displayed list. On the **Filter By** drop-down list, select from the list of available options. You can also enter filter text in the search field.
- You can refresh the current view by selecting . **Refresh** updates the display of your current user session. Changes made by other users are not dynamically updated.

## Import Data

In SAS Environment Manager 3.2 you can import data on the Import Data To Caslib window. You can import from a database server, remote file system, local files, and social media feeds. In the Data area, you can import data from the **Libraries** or **Servers** view.

1. In the **Libraries** or **Servers** view, select a library or server and then select from the taskbar.
2. In the Import Data To Caslib window, select either of the following tabs:
   - **Data Sources** enables you to create a connection between a caslib and a database server or a remote file system. For more information about the **Data Sources** tab see "Data Sources Tab: Access Databases or Remote File Systems" in SAS Data Explorer: User's Guide .
   - **Import** enables you to create a connection between a caslib and a local file or other data source. For more information about the **Import** tab see "Import Tab: Access Local Files, Social Media Content, or Esri Data" in SAS Data Explorer: User's Guide .

The ability to import data or tables in SAS Environment Manager is affected by two factors:

- Whether you have Read permission on the `/casManagement_capabilities/importData` object URI in the **Security** ⇨ **Rules** area of SAS Environment Manager. You must have Read permission granted in order to import data or tables. If you do not have Read permission granted, the **Import** function is disabled. In the initial configuration, all authenticated users have the necessary access. For information about restricting the ability to import tables in SAS Environment Manager or SAS Visual Analytics, see Adjust Rules for Access to Functionality.
- Whether you have the necessary access to the target caslib. For information about specific requirements, see "Compound Tasks" in SAS Viya Administration: Cloud Analytic Services Authorization .

## Manage Tables

In the Data area, you can load, unload, and delete tables when needed. You can also view table properties, the different columns for a table, and authorization settings for users.

### Load a Table

1. In the **Libraries** view, select a caslib.
2. Right-click, and select **Tables**. Or select from the taskbar.
3. Select the table that you want to load.
4. Right-click, and select **Load**. Or select from the taskbar.

**Note:** In the **Tables** view for a caslib, loaded tables are identified with a green circle in the **State** column.

### Unload a Table

1 In the **Loaded tables** or **Libraries** ⇨ **Table** view, select a loaded table. Loaded tables are identified by a green circle in the **State** column.
2 Right-click, and select **Unload**. Or select from the taskbar.
3 In the Confirmation window, click **OK**.

**Note:** In the **Tables** view for a caslib, unloaded tables are identified by a red square in the **State** column.

### Delete a Table

1 In the **Loaded tables** or **Libraries** ⇨ **Table** view, select a table.
2 Right-click, and select **Delete**. Or select from the taskbar.
3 A confirmation window appears. Select **Yes** to delete the table. You can also select whether you want to delete any direct access controls for the table.

### View Table Properties

1 In the **Loaded tables** or **Libraries** ⇨ **Table** view, select a table.
2 Right-click, and select **Properties**. Or select from the taskbar. Read-only information is displayed in the Table Properties window.

### View Table Columns

1 In the **Loaded tables** or **Libraries** view, select a loaded table.
2 Right-click, and select **Columns**. You can also double click on a table row or select from the taskbar. A list of columns is displayed.
3 To view the properties of a column, select a column, right-click, and select **Properties**. Read-only information is displayed in the Column Properties window.

**Note:** The table must be loaded to view columns.

## Manage Caslibs

In the Data area, you can examine available caslibs and the assigned tables for each caslib. You can add a new caslib, modify path and description settings for a caslib, and delete a caslib. You can also view caslib properties.

### Add a Caslib

1 In the **Libraries** view, click .
2 In the New Caslib window, specify general settings as follows:

| | |
|---|---|
| **Server** | Select a server. Only servers to which you are authorized to add a global caslib are listed. See "Caslib Management Privileges" in SAS Viya Administration: SAS Cloud Analytic Services . |
| **Data source type** | Select the type of data source. The **Data Source** area automatically displays the settings for the selected data source. |

| | |
|---|---|
| **Path** | Specify data source-specific information for the caslib. |
| **Name** | Specify a name for the caslib. |

3 Depending on the data source type that you selected, different settings are available on the **Data Source** area. Below are the different data sources that you can select.

- `PATH`
- `HDFS`
- `DNFS`
- `LASR`
- `Oracle`
- `Teradata`
- `Hadoop`
- `Postgres`
- `Impala`
- `ODBC`
- `DB2`

For further information about these data sources and the specific parameters for each data source, see the addCaslib Action.

4 After you have entered all of the parameter settings, click **Save**. The new caslib will be listed in the **Libraries** display.

## Modify a Caslib

1 In the **Libraries** view, select a global caslib.
2 Right-click, and select **Edit**. Or select from the taskbar.
3 In the **Edit Caslib** window, change the caslib path or description as needed.
4 Click **Save**.

When editing a caslib, the following restrictions apply:

- Only **Path** and **Description** fields can be edited for PATH, HDFS, and DNFS caslibs.
- Personal caslibs are created for each user and have a specific meaning that does not allow them to be edited.

## View Caslib Properties

1 In the **Libraries** view, select a caslib.
2 Right-click, and select **Properties**. Or select from the taskbar. Read-only information is displayed in the Library Properties window.

## View Tables for a Caslib

1 In the **Libraries** view, select a caslib.
2 Right-click, and select **Tables**. You can also double-click a caslib row or select from the taskbar.

A list of the tables that are assigned to the caslib is displayed.

### Delete a Caslib

**CAUTION! When you delete a caslib, all associated in-memory tables are immediately unloaded.**

**Note:** Deleting a caslib does not affect persisted files in the corresponding data source.

**Note:** You cannot delete a personal caslib.

1. In the **Libraries** view, select a caslib.
2. Right-click, and select **Delete**. Or select from the taskbar.
3. In the confirmation window, click **Yes**.

## Manage Servers

In the Data area, you can view servers and the caslibs that are assigned to an individual server. You can view and manage sessions for a server and you can also view and manage various properties for a server.

### View Caslibs for a Server

1. In the **Servers** view, select a server.
2. Right-click, and select **Libraries**. You can also double click a server row or select from the taskbar.

   A list of the libraries that are assigned to the server is displayed. You can continue to drill down into tables and columns.

### Manage Sessions for a Server

1. In the **Servers** view, select a server.
2. Right-click, and select **Sessions**. Or select from the taskbar. On the Sessions window, you can view sessions for the current server. You can refresh the session view by selecting .
3. You can also select a session and terminate the session. On the Sessions window, select the check-box for the session or sessions that you want to terminate. Click . The session is terminated.

   For more information see "Terminate a CAS Server Session" in SAS Viya Administration: SAS Cloud Analytic Services .

### Manage Properties for a Server

1. In the **Servers** view, select a server.
2. Right-click, and select **Properties**. Or select from the taskbar. The following properties are displayed in the Server Properties window:
   - **Basic Properties** displays information such as the type, host, and port.
   - **Nodes** displays the node information for the selected server including the **Name**, **Role**, **Connected** status, and **IP Address**.

     For more information see "Manage CAS Nodes" in SAS Viya Administration: SAS Cloud Analytic Services .
   - **Superuser Role Membership** displays users who are members of the Superuser role. For more information about the Superuser role, see CAS Server Roles.
   - **Caslib Management Privileges** displays directly granted privileges for different users. For more information, see Adjust Caslib Management Privileges.

If you are a member of the SAS Administrators group, you can assume the Superuser role and see additional server properties.

1 In the **Servers** view, select a server.
2 Right-click, and select **Assume the Superuser role**.
3 Right-click, and select **Properties**. Or select from the taskbar.

   You can now edit property settings on the Server Properties window. You can edit settings for **Nodes**, **Superuser Role Membership**, and **Caslib Management Privileges**.

   In addition to the standard properties, the **Paths List** properties are displayed. The **Paths List** displays a list of unavailable paths also known as **Black List** paths.

   Only users who assume the Superuser role for a server can see and manage that server's paths list.

   To manage Paths List properties, see "Manage Whitelists and Blacklists" in SAS Viya Administration: SAS Cloud Analytic Services .

### Stop a Server

If you are a member of the SAS Administrators group, you can assume the Superuser role and stop a Server.

**CAUTION! You cannot restart the server from SAS Environment Manager.**

For more information see Stop a CAS Server.

## Manage User-Defined Formats

In the Data area, you can view and manage user-defined formats. The **Format Filter** pane displays the available format libraries and user-defined formats. You can select the CAS server that you want to work with from the **Server** list. And you can select specific format libraries or specific formats. Items that you select on the **Format Filter** pane are displayed on the **Format name** table.

You can also search for specific formats in the **Format name** table. From this view you can import formats contained in item stores that you create with the FMTC2ITM procedure. You can also add new user-defined formats. You can then edit, copy, and delete user-defined formats. You can also view properties for a user-defined format.

### Format Libraries

There are five format libraries that are included with SAS Viya. These format libraries are: USERFORMATS1, USERFORMATS2, USERFORMATS3, USERFORMATS4, and USERFORMATS5. The format libraries initially do not contain any formats. Some application products add formats during installation.

### Import an Item Store

You can import SAS client user-defined formats from a SAS item store that was created with the FMTC2ITM procedure. To import formats from an item store:

1 Click to open the Import Formats window.
2 Enter a directory path for the item store on the **Import itemstore** field.
3 Select a format library from the `Target format library` drop-down list.
4 Click **Compare**. Formats that are found in the item store are populated in the **Comparison Results** list. The displayed list of format libraries is derived from the CAS server option FMTSEARCH.

   The number of formats that are found is displayed. The status of the formats that are located is also indicated. The following information values are updated:

   - **Information** - This value shows the number of formats that are found in the item store that can be loaded without conflict.

- **Warning** ⚠ - This value indicates that there is a possible conflict with one or more of the formats that are listed in the **Comparison Results** list. For example, a duplicate format has been found with an identical range as an existing format.
- **Error** ⊗ - This value indicates that there is a conflict with one or more of the formats that are listed in the **Comparison Results** list. For example, a duplicate format of an existing format has been found. However, the range of values is different between the two formats.

5 In the **Comparison Results** list, select the formats that you want to import.

For any formats that have a warning or error conflict, you can analyze the conflict on the **Format Properties** panel. Select a format that has a conflict by clicking on the format name. The problem range values for both the new format and the existing format are displayed.

For formats that are in conflict with existing formats, you can assign the new format a different name than the existing format. In the **New Format** field, enter a name for the new format. Click **Apply**. When you select **Import**, the new format is imported with the new name.

6 You can also click **Select all** to select all of the found imports. Select **Import**. The new formats are now available.

**Note:** Update the `FMTSEARCH` server option list of format libraries to change the displayed list of format libraries. For more information about `FMTSEARCH` see "Converting a SAS Catalog to an Item Store with the FMTC2ITM Procedure" on page 19.

## Add a User-Defined Format

To add a new user-defined format:

1 Select + to open the New Format window.
2 Select one of the SAS provided format libraries from the **Format Library** drop-down list. The new format will be assigned to this library.
3 Select either `Character` or `Numeric` from the **Type** drop-down list.
4 Enter a name for the format in the **Name** field.
5 Check the **Save without locale** option if you do not want to include a default locale. By default a format is created with a locale.
6 Add rows to the **Range** table. Select +. Enter values in the **Name** or **Value** columns for each row. You can delete a row by selecting 🗑.

You can also select a range of multiple rows to delete.

7 After you have added any needed rows, click **Save**. The new format is listed in the **Format name** table.

## Edit a User-Defined Format

To edit a user-defined format:

1 Select a format and click ✎.
2 On the Edit Format window, make any needed changes to the **Range** table of rows. You can edit values in the **Name** or **Value** columns for existing rows.

Click + to add rows and 🗑 to delete rows.

3 Click **Save** when finished.

## Copy a User-Defined Format

To copy a user-defined format:

1 Select a format and click ⧉.
2 On the Copy Format window, enter a name for the copied format in the **Name** field. Make any needed changes to the **Range** table of rows. You can edit values in the **Name** or **Value** columns for existing rows.

Click + to add rows and 🗑 to delete rows.

3 Click **Save** when finished. The new format is listed in the **Format name** table.

### Delete a User-Defined Format

To delete a user-defined format:
1 Select a format and click 🗑. The Delete pane opens.
2 Select **Delete**.

### View Properties for a User-Defined Format

To view properties for a user-defined format:
1 Select a format and click ▦. The Format window appears.
2 You can now view properties and row values for the format.

## Authorization

You can view and modify authorization settings for tables or libraries. Select one of the following options:

**View Authorization**

Right-click, and select **View Authorization** or select from the taskbar. The View Authorization window appears. You can view different users and access levels. If you select **Show individual permissions**, different permission settings are available to view.

You can also select + to open the Select Identities window.

If you select **Edit**, the Edit Authorization view for the selected item is displayed.

**Edit Authorization**

Right-click, and select **Edit Authorization** or select from the taskbar. The Edit Authorization window appears. You can view different users and modify access levels. If you select **Show individual permissions**, different permission settings are available for edit.

You can also select + to open the Select Identities window.

You can also right-click on a table or library and select **View Authorization** or **Edit Authorization**.

For further information about managing access, see *SAS Viya Administration: General Authorization* and *SAS Viya Administration: Cloud Analytic Services Authorization*.

## Data Encryption

When you add a caslib with the **New CAS Library** function, you can choose to encrypt the data for that caslib. Depending on the data source type that you select, the **Enable encryption** option is available on the **New CAS Library** window. Select **Enable encryption** and select a domain from the list of available domains or create a new domain by selecting .

You can also view the encryption status for individual tables in a caslib. Select a table and select ▦. The encryption status is listed on the Table Properties window.

**Note:** In the Data area, the **Encryption** column is hidden by default.

For more information about encrypting data, see Encryption in SAS Viya: Data at Rest.

# Data Administration: How to (CAS Server Monitor)

## Introduction

CAS Server Monitor enables you to monitor and administer your CAS server. Within CAS Server Monitor, the **System State** view contains various CAS server properties and settings, including the **Global Caslibs** table. This table displays the global caslibs for your environment. From here you can add and delete global caslibs and modify access controls for users and groups.

These instructions explain how to manage global caslibs using CAS Server Monitor.

## Add a Global Caslib

1 On the **System State** page, select **Global Caslibs**.
2 Click **Add**.

> **TIP** If the **Add** button is disabled, you are not authorized to add a global caslib. See "Caslib Management Privileges" in SAS Viya Administration: SAS Cloud Analytic Services .

3 On the Add Global Caslib pane, specify general settings as follows:

| | |
|---|---|
| **Caslib** | Enter a caslib name. |
| **Description** | Enter a description for the caslib. |
| **Path** | Enter data source-specific information. |
| **Subdirectories** | For a path-based caslib, specifies whether tables and files in subdirectories of the specified path are accessible from the caslib. |
| **Create directory** | For a path-based caslib, creates the host directory that you specify in the **Path** field, if that directory does not already exist. |
| **Permission** | For a path-based caslib, sets host-layer permissions on the directory. See "Using CAS to Modify Host Access " in SAS Viya Administration: Cloud Analytic Services Authorization . |
| **Active on add** | Specifies whether the new caslib becomes the active caslib in your current session. |
| **Hidden** | Makes the caslib and its tables unlisted in certain contexts. See "Reduced Visibility: Hidden Caslibs" in SAS Viya Administration: Cloud Analytic Services Authorization . |
| **Transient** | Specifies that the caslib is scoped to the current session only. |
| **Data source** | Specifies the type of source data for the caslib. |
| **Data encryption password** | Specifies the encryption password for the caslib. |
| **Encryption domain** | Specifies the encryption domain for the caslib. |

4 Specify additional settings as needed. For information about caslib properties, see addCaslib Action.

5 Make sure your settings are as intended. In CAS Server Monitor, caslib properties are not editable.
6 Click **OK**.

## Delete a Global Caslib

**CAUTION! When you delete a caslib, all associated in-memory tables are immediately dropped.**

**Note:** Deleting a caslib does not affect persisted files in the corresponding data source.

1 On the **System State** page, click **Global Caslibs**.
2 At the end of the row for the caslib, click ⋮, and select **Drop Caslib**. On the Drop Global Caslib pane click **OK**.

## Manage Access to a Global Caslib

1 On the **System State** page, click **Global Caslibs**.
2 At the end of the row for the caslib, click ⋮, and select **Edit Access Controls**. The Edit Access Controls window appears. From here you can grant or deny permission settings to different users.

See *SAS Viya Administration: Cloud Analytic Services Authorization*.

# Loading Geographic Polygon Data as a CAS Table

## Overview of Loading Geographic Polygon Data

Some SAS Viya applications such as SAS Visual Analytics can display geographic maps with colored map regions. By default, countries and their first-level subdivisions can be displayed as a region map. To display other types of map regions, such as postal codes or sales regions, you must define a custom polygon provider that contains the polygons (geographic region shapes).

You can load two types of polygon data into CAS for use in a polygon provider: Esri shapefiles, and SAS map data sets.

After you have loaded the polygon data into CAS, you must define a polygon provider that specifies the parameters for the polygon data. For details about defining a polygon provider in SAS Visual Analytics, see "Create a Geography Data Item By Using Custom Polygonal Shapes" in SAS Visual Analytics: Working with Report Data .

**Note:** By default, SAS Visual Analytics can retrieve up to 250,000 polygon vertices at a time. If you encounter an error message in a geo map object about the number of polygon vertices, then you might need to reduce the density of your polygon data or filter the data query for your geo map object. In some cases, a very wide ID column in your polygon data can further limit the number of polygon vertices that are retrieved. Check the width of your ID column in SAS Data Explorer if you encounter this message.

## Loading Polygon Data from Esri Shapefiles

### Overview

To load Esri shapefile data into CAS, you must first convert the shapefile into a SAS data set.

SAS provides two autocall macros to help you inspect and load Esri shapefiles:

%SHPCNTNT
displays the contents of the specified shapefile.

%SHPIMPRT
converts a shapefile into a SAS data set and loads it into CAS.

> **TIP** Where possible, use shapefiles with unprojected latitude and longitude values. Configuring a polygon provider for projected data can be difficult for users who are inexperienced with map data.

## %SHPCNTNT Autocall Macro

The %SHPCNTNT macro displays the contents of the specified shapefile. You can use the %SHPCNTNT macro to identify which variable in the shapefile should be used as an ID variable.

The syntax for the %SHPCNTNT macro is as follows:

%SHPCNTNT(SHAPEFILEPATH=*path-to-shapefile*)

**SHAPEFILEPATH=*path-to-shapefile***
specifies the full path to the shapefile with the .SHP extension. Do not enclose the file path in quotation marks.

## %SHPIMPRT Autocall Macro

The %SHPIMPRT macro converts the shapefile into a SAS data set and then loads it into CAS.

**Note:** To load tables into CAS, you must configure an authentication file. See *Client Authentication Using an Authinfo File*,

The syntax for the %SHPIMPRT macro is as follows:

%SHPIMPRT(*options*)

**SHAPEFILEPATH=*path-to-file***
specifies the full path to the shapefile with the .SHP extension.

**ID=*id-column***
specifies a field in the shapefile that identifies the polygons in the map.

**Requirement** The ID column must contain character data, and cannot contain special characters or double-byte characters.

**OUTTABLE=*table-name***
specifies the name of the output table that is loaded into CAS.

**CASHOST=*machine-name***
specifies the machine name of the CAS server.

**CASPORT=*port-number***
specifies the port for the CAS server.

**CASLIB=*library-name***
specifies the library on the CAS server where the output table is loaded.

**REDUCE=0|1**
(Optional) specifies whether to reduce the density of the polygon data. A value of 1 specifies that the data density is reduced, and a value of 0 specifies that the data density is not reduced.

Reducing the density of your polygon data can improve performance and might enable a greater number of map regions to be displayed at one time.

| | |
|---|---|
| **Default** | 0 |
| **Requirement** | A license for SAS/GRAPH software is required to reduce the density. |

The following example loads a shapefile without reducing the polygon density:

```
%shpimprt(shapefilepath=/tmp/myfile.shp, id=GEOID, outtable=mytable, cashost=cloud.example.com,
   casport=5570, caslib='casuser');
```

The following example loads a shapefile and reduces the polygon density:

```
%shpimprt(shapefilepath=/tmp/myfile.shp, id=GEOID, outtable=mytable, cashost=cloud.example.com,
   casport=5570, caslib='casuser' reduce=1);
```

## Loading Polygon Data from SAS Map Data Sets

To use a SAS map data set as a polygon provider, you must perform the following steps:

1. Create a sequence variable to enable the polygon segments to be read in the correct order. In a SAS DATA step, you can use the _n_ automatic variable to store the observation number as a sequence variable. For example, the following DATA step creates a sequence variable for the MYMAP data set:

   ```
   data mymap;
     set mymap;
     sequence = _n_;
   run;
   ```

2. (Optional) Subset your polygon data to decrease the level of detail and improve performance. Reducing the level of detail might also enable you to display a greater number of map regions at one time.

   If you have a license for SAS/GRAPH, then you can use the GREDUCE procedure to create a DENSITY variable that enables you to reduce the density of your polygon data. Depending on the source of your map data sets, a DENSITY variable might already be present. For more information about the DENSITY variable and the GREDUCE procedure, see *SAS/GRAPH and Base SAS: Mapping Reference*.

   You can use the DENSITY variable in a WHERE statement in a DATA step to reduce the detail in your polygon data. For example, the following DATA step reduces the MYMAP data set to exclude segments that are density level 4 or greater:

   ```
   data mymap;
     set mymap;
     where(density<4);
   run;
   ```

3. Load the data set in your SAS Cloud Analytic Services environment.

# CAS Table State Management

## Overview of CAS Table State Management

CAS table state management enables you to manage the import, load, and unload of source files in CAS. CAS table state management is performed through the use of jobs that are created from sample jobs that are provided by SAS.

For SAS Viya 3.3, you can import batch data that is directly accessible using a caslib, but might not be in the desired caslib or format. This type of import is used to take source data and make a copy in SASHDAT format.

For example, business processes might produce new data each night in CSV or SAS7BDAT format. It is possible to access the data directly using a global caslib that points to the source of the data. However, for performance reasons, it might be desirable to make a copy of the data in SASHDAT format.

Sample jobs should be used as a starting point. These sample jobs can be used as is for the Public caslib that is associated with the cas-shared-default CAS server. The jobs can be copied, and the copies can be edited or deleted. A job includes the specific options required by the job. In the context of CAS table state management, a job performs an import, load, or unload operation on input files, tables, or loaded tables. Jobs can be listed, copied, updated, and deleted on the SAS Environment Manager Scheduling page. For CAS table state management sample jobs, this page also enables you to make copies that are used to import, load, and unload batch data. Each job can be submitted manually, or scheduled for later execution.

## Sample Jobs in SAS Environment Manager

For SAS Viya 3.3, there are three sample jobs provided by SAS for managing table state. These jobs are available on the Scheduling page of SAS Environment Manager. Below are the sample jobs:

**`Sample: Import cas-shared-default Public data`**
This job demonstrates settings that import all CSV, SAS7BDAT, and EXCEL files in the Public caslib to SASHDAT files in the same caslib.

**`Sample: Load cas-shared-default Public data`**
This job demonstrates how to load all SASHDAT files found in the Public caslib.

**`Sample: Unload cas-shared-default Public data`**
This job demonstrates how to unload all loaded CAS tables in the Public caslib that have not been accessed within the past 7 days.

On the Scheduling page of SAS Environment Manager, the sample jobs are listed on the **Jobs** pane. The sample jobs operate on a CAS server named **`cas shared default`**. You cannot edit or delete the sample jobs. However, you can copy the sample jobs to create unique jobs that you can further customize. Copied jobs contain the options that you can define and update as needed.

To create a new job, select **Copy** on one of the sample jobs. You can now customize the options for the new job by selecting .

For further information about the Scheduling page in SAS Environment Manager, see “Scheduling: How to (SAS Environment Manager)” in SAS Viya Administration: Scheduling .

## Viewing Properties for a Job

You can view properties for copied jobs on the Scheduling page. You must select the job and then select . The Job Properties window opens.

On the Job Properties window, select the **Properties and Arguments** tab. On the **View** drop-down list, select **Arguments**. You can now view the settings for the job under the **Value** column. Holding your pointer over the value shows all of the options in a tooltip.

## Editing Options for a Job

You can edit options for copied jobs on the Scheduling page. Select the copied job and then select . The Job Properties window appears. On the Job Properties window, select the **Properties and Arguments** tab. On the **View** drop-down list, select **Arguments**. You can now select .

The Edit Job window appears. The different options for the job are listed in the **Options** pane. The job options panel contains the full set of job options as a JSON-formatted string. You can manually edit the settings and simple edits can be made in place.

**TIP** For more complex edits, copying the options to a JSON editor, making the changes, and copying the result back into options might be helpful.

The following table contains options that are common to the import, load, and unload jobs:

*Table A.1 Common Options*

| Option | Description |
|---|---|
| serverName | The name of the CAS server on which the operation will be performed. |
| inputCaslib | The caslib name used as input for the job. For import and load jobs, this is the caslib that contains source files or tables. For an unload job, this is the caslib that contains potential tables to unload. |
| outputCaslib | The caslib name used for output of the job. For an import job, this is the caslib where output files are written. For a load job, this is the caslib where CAS tables are loaded. outputCaslib is not applicable for an unload job. |
| filter | The filter is used to subset the list of items from the inputCaslib upon which job operations are performed.<br>See "Filter Syntax" on page 15 for more details and example filters. |

## Filter Syntax

Job options can also contain filters. In its simplest form, a filter selects an item based on whether a condition passes. For example, to select an in-memory table whose name is exactly MYDATA, the following example filter could be used:

```
eq(name,'MYDATA')
```

In the next example, the filter is used to select a source table name ending in lowercase SASHDAT:

```
endsWith(sourceTableName,'.sashdat')
```

There are several operators that can be used in a filter. The following table contains these operators:

*Table A.2 Filter Operators*

| Operator | Description | Example | Example Result |
|---|---|---|---|
| eq | True if the parameters specified are equal. | `eq(name,'MYDATA')` | Only the table named MYDATA is selected. |
| startsWith | True if the value of the first parameter begins with the value of the second parameter. | `startsWith(sourceTableName,'DEPTA_')` | Only source tables beginning with 'DEPTA_' are selected. For example: DEPTA_CUSTOMERS, DEPTA_ADDRESSES. |

| Operator | Description | Example | Example Result |
|---|---|---|---|
| endsWith | True if the first value of the parameter ends with the value of the second parameter. | `endsWith(sourceTableName,'.sashdat')` | Only source file or table names ending in lowercase .SASHDAT are selected. |
| contains | True if the value of the first parameter contains the value of the second parameter. | `contains(name,'SPECIAL')` | Only tables whose name contains SPECIAL are selected. For example: MYSPECIALDATA, SPECIALDATA, THISSPECIALDATA. |
| in | True if the value of the first parameter contains any following value. | `in(name,'TABLE1','TABLE2','TABLE3')` | Only tables TABLE1, TABLE2, or TABLE3 are selected. |

The following table contains filter fields that can be used in expressions:

***Table A.3*** *Filter Fields*

| Field Name | Content |
|---|---|
| name | This field represents the CAS table name (whether loaded or unloaded). |
| sourceTableName | This field represents the name of the source file in the input caslib. |
| tableReference.sourceTableName | This field is an alias for the sourceTableName field and can be used in place of it. |

The following table contains filter examples:

***Table A.4*** *Filter Examples*

| Example | Filter |
|---|---|
| by file extension (.SASHDAT, .CSV, .SAS7BDAT) | `or(endsWith(sourceTableName,'.sashdat'), endsWith(sourceTableName,'.csv'), endsWith(sourceTableName,'.sas7bdat'))` |
| by substring (contains some string) | `contains(name,'DATA')` |
| by exact match | `eq(name,'MAILORDER')` |
| by list of inputs | in(name,'AIRLINE','CUSTOMERS','WORLDBANK') |
| using multiple conditions where either are true | `or( eq(name,'MYDATA'), endsWith(name,'YOURDATA') )` |

| Example | Filter |
| --- | --- |
| using multiple conditions where both are true | `and( contains(tableReference.sourceTableName,'DEPTA_'), endsWith(tableReference.sourceTableName,'.sashdat') )` |

## Importing Data

The **`Sample: Import cas-shared-default Public data`** job imports CSV, SAS7BDAT, and EXCEL files to SASHDAT files. It imports to the **`Public`** library on an example CAS server named **`cas-shared-default`**.

This job enables an import from the caslib source defined for the CAS server. By default, the import job imports CSV, SAS7BDAT, and XLS,XLSX (EXCEL) files. It imports those files to the target caslib's source location as SASHDAT files of the same name. Source files can therefore be placed in a path-based caslib (PATH, and DNFS for example) that is accessible by the CAS server controller. The default path for imported files is **`/opt/sas/viya/config/data/cas/default/public/`**.

**Note:** For situations where the SASHDAT copy is not required, the load job can be used to load the file directly into memory as a CAS table.

On the **Scheduling** page of SAS Environment Manager, the sample job *Sample: Import cas-shared-default Public data* is available.

## Loading Data

The **`Sample: Load cas-shared-default Public data`** job performs a load operation on managed files or tables in the target caslib. It then creates an in-memory CAS table of the same name in the target caslib.

This job enables you to preload tables for which there is a high user demand. Or, for scenarios where the amount of time needed to load the table is too long due to data size.

On the **Scheduling** page of SAS Environment Manager, the sample job **`Sample: Load cas-shared-default Public data`** is available. You can modify the following settings for copies of this job:

***Table A.5*** *Load Settings*

| Setting | Value | Default Value | Sample Values |
| --- | --- | --- | --- |
| refresh | Boolean | false | true, false |

**Note:** When the refresh option is set to **`true`**, each table selected by the filter is unloaded first. If the table is not sourced from the input caslib, it is not reloaded. Therefore, it is important to ensure that the filter is properly set to select only the tables for which you want a refresh. Tables are refreshed only if they are sourced from the caslib that is specified with the inputCaslib setting.

## Unloading Data

The **`Sample: Unload cas-shared-default Public data`** job unloads tables in the target caslib either immediately, or based on recent access. This enables you to schedule forced unloads of tables on a routine basis. Or you can schedule an unload request that is based on how often a table is used. The sample job unloads infrequently accessed data in the **`Public`** table on the **`cas-shared-default`** server.

The setting *unloadAccessThreshold* is available in the settings for this job. If *unloadAccessThreshold* is set to PT0D, all tables in the target caslib are unloaded when the job is run. However, if it is set to a specific time period, those tables that are not accessed within the set time period are unloaded. The sample job is **`Sample: Unload cas-shared-default Public data`**, and by default, uses an *unloadAccessThreshold* setting of P7D (7 days).

On the **Scheduling** page of SAS Environment Manager, the sample job **`Sample: Unload cas-shared-default Public data`** is available. For copies of this job, you can modify the *unloadAccessThreshold* setting. The following example time threshold values are possible:

***Table A.6*** *Unload Settings*

| Value | Description |
|---|---|
| P0D | zero days. This setting results in an immediate unload. There is no threshold. |
| P7D | period of 7 days |
| P5M | period of 5 months |
| PT4H | period of time of 4 hours |
| PT5M | period of time of 5 minutes |
| PT45S | period of time of 45 seconds |

## Source-Data-Specific Settings

Currently you can import CSV, SAS7BDAT, and EXCEL files with the *Sample: Import cas-shared-default Public data* sample job. Settings for these import file types are listed in the following tables.

***Table A.7*** *CSV*

| Setting | Value Type | Default Value | Sample Values |
|---|---|---|---|
| delimiter | character | , | , |
| guessRows | integer | 200 | 20,50,500 |
| allowTruncation | Boolean | true | true,false |
| encoding | string | utf-8 | utf-8 |
| getNames | Boolean | true | true,false |

***Table A.8*** *SAS7BDAT*

| Setting | Value Type | Default Value | Sample Values |
|---|---|---|---|
| charMultiplier | decimal | 2 | 1,2,2.5,3,4 |

*Table A.9* *EXCEL and XLS*

| Setting | Value Type | Default Value | Sample Values |
|---|---|---|---|
| getNames | Boolean | true | true,false |

## Execution and Monitoring of Jobs

The Scheduling page of SAS Environment Manager enables you to schedule and execute the jobs that you define and customize. You can choose to run jobs as the SAS Administrator or as a different user. You can also schedule or unschedule a job.

In the **Jobs** pane of the Scheduling page, right-click on a job. Available functions for that job are listed. You can also execute functions from the icon menu on the **Jobs** pane.

If you select **Run** for a job, you can check the execution of that job by accessing the Data page of SAS Environment Manager. On the Data page, select **Libraries** from the **View** menu. Right-click on the `Public` library for the `cas-shared-default` server. Select **Tables**. From here you can check for the source tables that you are importing or loading.

You can also check the execution of that job from SAS Job Monitor. On the **Jobs** pane of the Scheduling page, select **Monitor Jobs**. This opens SAS Job Monitor. From here you can view the different jobs that have been executed and download the log file that contains details of the execution.

**Note:** The log file is updated as progress is made. So downloading the log file while the job is running shows progress until that point only. To see later progress, you must download the log again for those jobs.

> **TIP** Open SAS Job Monitor in a separate browser tab or browser instance. You will be able to see job progress while still keeping Tables open in the Libraries view of the Data page. You can watch tables populate, load, or unload.

# Preliminary Tasks for User-Defined Formats in SAS Viya 3.3

## Overview

For SAS Viya 3.3, CAS does not read a SAS catalog directly. One way to move formats stored in a SAS catalog to a CAS server is to use the FMTC2ITM procedure. This procedure copies the formats in the catalog to a physical file. The physical file is often referred to as an item store. The physical file must be accessible from the CAS server controller and the user must have permission to read the file.

SAS Environment Manager can then import formats from an item store. To create an item store the FMTC2ITM procedure is used. The FMTC2ITM procedure is available in SAS 9.4M3 and later.

## Converting a SAS Catalog to an Item Store with the FMTC2ITM Procedure

The FMTC2ITM procedure uses a SAS format catalog as input and as output, and produces an item store that is read in CAS. The FMTC2ITM procedure is used to convert one or more format catalogs into a single item store.

### Syntax

**FMTC2ITM** <options>;

SELECT<member-list>;

RUN;

### Options

**PRINT**
displays information about each member that is being written.

**DEBUG**
debugs information about the records that are being written.

**CATALOG=memname | libname.memname | ( list )**
specifies catalog(s) that will be converted to an item store.

**ITEMSTORE= fileref | 'filename'**
specifies the item store file that is being created.

**XMLFILE= fileref**
specifies an XML file that is created to accompany the item store file.

**PAGESIZE=n**
specifies the page size.

**LOCALE**
specifies locale-sensitive prefixes that are added to item store memnames.

### FMTC2ITM Procedure Example

The following example converts the catalogs *formats.orionfmt* and *formats.siriusfmt* to the item store *format1*.

```
proc fmtc2itm catalog=(formats.orionfmt, formats.siriusfmt)
print locale itemstore="/users/dsmith/formats/format1";
run ;
```

### About the FMTC2ITM Procedure

Note that the item store is always written as new, so if the **ITEMSTORE=** option refers to an existing item store, it will be completely overwritten. For a file-based item store, the **XMLFILE=** option can also be provided. It is populated with a small XML stream that accompanies the item store file.

If the **CATALOG=** option is not given, then the default value is **WORK.FORMATS**. If the **CATALOG=** option is given, it can be a single-level name, which is interpreted as a catalog name in WORK. It can also be a two-level name, which is interpreted as a libname.memname for a catalog. It can also be a list of catalog names that is enclosed in parentheses.

For an item store that is a file, either a fileref or a quoted string pathname is given. For a list of catalog names that is enclosed in parentheses, each catalog is opened in order and the members of the catalogs are written to the item store. Only the first occurrence of the member is written out.

The SELECT statement is optional. If specified, it lists the formats that are selected to be placed in the item store. If a SELECT statement is not given, all formats are written to the item store.

## Using the Casstartup.lua File

The **casstartup.lua** file is a file that is processed as a LUA client session into the CAS server. It is used to perform static, default deployment tasks. Casstartup.lua comes pre-configured with default settings as part of

the deployment process. **Casstartup.lua** is included in the start-up processing hierarchy and is used during CAS server start-up. The command line default start-up value is **-startup casstartup.lua**.

During server start-up, you can establish default and custom user-defined format libraries for use in SAS Viya. In a default SAS Viya installation, the caslib **Formats** is created. The **casstartup.lua** file contains the default **addFmtLib** actions. These actions add format libraries **userformats1** through **userformats5**. The **casstartup.lua** file also contains the **setServOpt** action that is used to establish the format search list that each session starts with. Currently, **userformats1** through **userformats5** are placeholders for use with SAS Environment Manager.

Below is an example **casstartup.lua** file:

```
s:sessionProp_addFmtLib{caslib="Formats",fmtLibName="userFormats1",name="userformats1.sashdat",promote=true}
  s:sessionProp_addFmtLib{caslib="Formats",fmtLibName="userFormats2",name="userformats2.sashdat",promote=true}
  s:sessionProp_addFmtLib{caslib="Formats",fmtLibName="userFormats3",name="userformats3.sashdat",promote=true}
  s:sessionProp_addFmtLib{caslib="Formats",fmtLibName="userFormats4",name="userformats4.sashdat",promote=true}
  s:sessionProp_addFmtLib{caslib="Formats",fmtLibName="userFormats5",name="userformats5.sashdat",promote=true}
  s:configuration_setservopt{fmtsearch="userformats1 userformats2 userformats3 userformats4 userformats5"}
```

The recommended deployment best practice is to target a configuration path for all default filenames. If present, the default names will be found. For example, the following **cfgpath** option

```
-cfgpath /my/config/path
```

will find

```
/my/config/path/casstartup.lua
```

After the deployment tasks have been accomplished, **casstartup.lua** invokes **casstartup_usermods.lua**. Site specific format libraries associated with the **addFmtLib** action should be placed in **casstartup_usermods.lua**. The **casstartup_usermods.lua** file is used for site customization and is automatically applied when present. The local system administrator can add any site-specific start-up processes, such as table loading, to **casstartup_usermods.lua**.

**Note:** **Casstartup_usermods.lua** is not replaced when software is updated.

# Data Administration: Reference

## Data Administration: Interfaces

Interfaces

All CAS data management requirements and constraints are always fully enforced. Not all interfaces enable you to see and interact with all CAS data management features.

In the following table, the shaded part of each circle is an approximation of the amount of CAS data management functionality that a particular interface exposes.

***Table A.10*** *Interfaces to Data Administration*

| | | |
|---|---|---|
| ● | Tables Action Set | A programmatic interface for CASL (the CAS procedure), Python, and Lua. |
| ◕ | SAS Environment Manager | The enterprise graphical web application for administration. |

| | | |
|---|---|---|
| ◒ | CAS Server Monitor | A graphical web application that is embedded in the CAS server. Supports adding and deleting global caslibs. |
| ◔ | CASLIB statement | A programmatic interface for adding caslibs. See CASLIB statement. |

## Predefined Caslibs

The following caslibs are automatically created during deployment. Each caslib has a default assignment and specifications.

| | |
|---|---|
| AppData* | **`/opt/sas/viya/config/data/cas/default/appData/`**<br>Stores data that specific applications use for internal purposes. |
| Formats | **`/opt/sas/viya/config/data/cas/default/formats/`**<br>A shared location for user-defined formats.<br>All users can read. Administrators can read and write. |
| Models* | **`/opt/sas/viya/config/data/cas/default/models/`**<br>Stores models created by SAS Visual Analytics for use in SAS Studio. |
| Public | **`/opt/sas/viya/config/data/cas/default/public/`**<br>A shared location for data.<br>All users can read and write.<br>See “Protecting Files in the Public Caslib” in SAS Viya Administration: Cloud Analytic Services Authorization . |
| ReferenceData* | **`/opt/sas/viya/config/data/cas/default/referenceData/`**<br>Stores per-server data that specific applications use for internal purposes. |
| Samples | **`/opt/sas/viya/config/data/cas/default/samples/`**<br>Stores sample data, supplied by SAS. |
| SystemData* | **`/opt/sas/viya/config/data/cas/default/sysData/`**<br>Stores application-generated data that is used for general reporting. |
| VAModels | **`/opt/sas/viya/config/data/cas/default/vamodels/`**<br>This is a library for ASTORE objects that are used within a SAS Visual Analytics report. |

| | |
|---|---|
| ProductData | `/opt/sas/viya/home/share/productData/`<br>Stores product data supplied by SAS. |

* Not included in a programming-only deployment.

**Note:** Some predefined caslibs are hidden or have limited access. For more information about hidden caslibs, see "Reduced Visibility: Hidden Caslibs" in SAS Viya Administration: Cloud Analytic Services Authorization

## Access to SAS 9.4 Data

If you are moving data from SAS 9.4 to SAS Viya, you will need to consider some preliminary information:

- You can move and you can share data between SAS 9 and SAS Viya environments using SAS/CONNECT.
- SAS Viya operates with UTF-8 encoded data. If your SAS 9 installation is not UTF-8 compliant, you might need to re-create your data sets.

See the following topics for more information:

- Comparing SAS 9 and SAS Viya
- SAS 9 and SAS Viya
- Sharing Data Between SAS 9 and SAS Viya using SAS/CONNECT
- Migrating Data to UTF-8 for SAS Viya 3.3