Encryption for Data at Rest: Overview

SAS Viya provides encryption in two contexts:

- **Data at rest** is data that is stored in databases, file servers, endpoint devices, and various storage networks. This data can be on-premises, virtual, or in the cloud. This data is usually protected in conventional ways by access controls. Numerous layers of defense are needed, and encrypting sensitive data is another layer.

  This document covers administrative tasks for encrypting files at rest in the SASHDAT format and it shows how tables that are imported into caslibs are encrypted. See Concepts for details.

- **Data in motion** is data that is being transmitted to another location. Data is most vulnerable while in transit. Sensitive data in transit should be encrypted. You can protect all traffic in transit between servers and clients. See “Overview” in Encryption in SAS Viya: Data in Motion.
SAS Viya uses Advanced Encryption Standard (AES) algorithms with 256-bit keys to encrypt data at rest. Refer to Caslibs, Files, and Tables if you need additional background on data and caslib concepts.

Use one of the following interfaces to encrypt files at rest:
- To manage encryption of data files interactively, use SAS Environment Manager.
- To programmatically encrypt data files, use the CASLIB Statement.

How To (SAS Environment Manager)

Introduction

Authorized administrators use SAS Environment Manager to create and manage data security. The Domains area enables you to create a stored credential (an encryption key) that is available to designated identities to facilitate loading of encrypted files. By default, when you are creating a new caslib, enabling that caslib for encryption is disabled. If you choose to enable encryption, this can be done by creating an encryption domain, and then associating that domain with your path-based caslib.

In the Domains area, you can perform the following tasks:
- Create a new encryption domain or use an existing encryption domain.
- Add users or group identities to the encryption domain.
- Create an encryption key (passphrase).

You must be a member of the SAS Administrators group and assume groups when you log on to SAS Environment Manager in order to create and manage Domains. For more information, see “Accessing SAS Environment Manager” in SAS Viya Administration: Using SAS Environment Manager.

From the Data area of SAS Environment Manager, you can use SAS Data Explorer to create and manage caslibs. By default, caslibs are not enabled for encryption. You can encrypt the tables in your caslibs by associating the caslibs with an encryption domain.

**CAUTION!** Be sure to keep a separate record of your encryption key. Once a caslib is created, it is not possible to change the encryption key setting or change the domain. The encryption key value cannot be retrieved through the software. If the caslib is deleted, the tables remain encrypted. To access those encrypted tables, you need to define a new caslib using the same path, the same domain, or a new domain using the same encryption key.

Note: When the encrypted data is loaded into CAS, it is decrypted at load time. Caslib authorizations apply to accessing the loaded data. Identities that are set in the encryption domain provide authorization for who can request that encrypted tables be loaded. Identities can be custom groups or individual user IDs. Custom groups provide access control by simple group membership.

The following instructions explain how to encrypt the tables in your caslibs using SAS Environment Manager.

About the Domains Page

The Domains page in SAS Environment Manager enables you to manage the following types of domains and credentials:

| authentication domain | Makes stored credentials (user IDs and passwords) available to designated identities to facilitate connections to servers that require a password. See “External Credentials: Overview” in SAS Viya Administration: External Credentials. |

Note: The Domains area is available only if you are a member of the SAS Administrators group.
connection domain Makes stored credentials (user IDs) available to designated identities to facilitate connections to servers that do not require authentication. See “External Credentials: Overview” in SAS Viya Administration: External Credentials.

encryption domain Makes a stored credential (an encryption key) available to designated identities to facilitate loading of encrypted files. This document describes this process.

Navigation
The Domains area is available if you are a member of the SAS Administrators group and you have opted into your assumable groups. For more information, see Accessing SAS Environment Manager.

1 In the applications menu (≡), under Administration, select Manage Environment.

2 In the navigation bar, locate the Security section and click Domains 🗝️.

3 You can select one of two views from the Domains page. The default view is Domains. From the View dropdown list, select one of the following views:

- **Domains**
  
  Lists all domains that are displayed. Domains is the default view. There are three types of domains: Authentication, Connection, and Encryption. On the Domains page, you can view the information for each domain that is defined, or you can create a new domain.

  When you create an encryption domain, you cannot delete that domain.

  Three default Authentication domains appear in the Domains view. These are described in Authentication Domains Defined In the Deployment.

  Note: This view is available only to SAS Administrators.

  Note: This document discusses only the Encryption domain.

- **Credentials**
  
  Enables you to access external data sources and other third-party products requiring authentication. Credentials are associated with a specific domain for use with a specific data source type. For more information, see External Credentials: Overview.

Manage Encryption Domains

Create a New Encryption Domain

1 In the Domains view, click 🌟.

2 In the New Domain window, specify general settings as follows:

- **ID**
  
  Create an ID name. Enter a unique ID for your encryption domain.

- **Type**
  
  Select the type of domain. There are three domain types, Encryption, Authentication, and Connection. Select Encryption from the list of available domains.

- **Description**
  
  Add a description.
Identities

Select Identities. You can select from users, groups, and custom groups. See below for how to add an identity.

Encryption key

Encryption passphrase or key.

Confirm encryption key

Enter the same passphrase as above.

For additional information about identities, from the New Domain window, click 📂.

3 Add Identities. From the New Domain window, click 📆.

a In the left pane of the Choose Identities window, you can filter by Users, Groups, and Custom Groups. From the drop-down menu, select 🧑‍💼Users, 🧑‍ ...(Groups), or 🧑‍ ...(Custom Groups).

You can also filter using the search 🔍.

Note: A best practice is to use a custom group. Then you can add additional users to this custom group as needed to grant access to the data. Be sure to assign correct permission for this custom group in the associated caslib Authorization.

b Move the selected user, group, or custom group to the Selected Identities pane. Click 🗑️.

c Click OK.

4 Add an encryption key.

5 Confirm encryption key.

6 After you have entered all of the parameter settings needed, click Save.

View Properties of an Encryption Domain

1 In the Domains view, select an ID row for Type Encryption.

2 Right-click, and select Properties. Or select 📋 from the taskbar. From the Domain Properties window, the ID, Type, Description, Date created, Date modified, who created the domain, and who modified the domain is displayed.

3 Click close.

Edit an Encryption Domain

From the Edit Domain window, you can add only a description. To modify the identities, go to the credentials view.

1 In the Domains view, select an ID.

2 Right-click, and select Edit. Or select 📋 from the taskbar.

3 From the Edit Domain window, add a Description of the Encryption Domain.

4 Click Save.
**View and Modify the Credentials of an Encryption Domain**

From the Credentials View for a selected encryption domain, you can view existing credentials that are associated with an Encryption domain, add new credentials either by clicking the edit or the new icons, delete identities, or just view the properties.

From the Domains page, you can view, edit, and add credentials to an existing domain from the Credentials view.

**View Credentials**

1. From the Domains page, select an ID of type Encryption.
2. Right-click, and select **Credentials**. Or select from the taskbar. In the Credentials for Domain view, credential properties are displayed.
3. To see all hidden columns, at the right edge of the table, click the **Options** icon, and select **Manage Columns**.
4. From the Manage columns window, select items to move from the **Hidden columns** pane to the **Displayed columns** pane. After selecting the **Hidden columns** to display, click the **Add arrow**.
   After selecting the **Displayed columns** to hide, click the **Remove arrow**.

   **TIP** To select more than one item, use the Shift key.

5. Click **OK**.

**Edit Credentials**

If you are a member of the domain, you can add identities to and remove identities (users, groups, and custom groups) from an existing Encryption Domain. You cannot change the type of domain or change the Encryption key from the credentials view.

1. In the **Credentials for Domain** view, select a credential entry.
2. Right-click, and select **Edit**. Or select from the taskbar.
3. To add and delete identities, in the Edit Credential window, click **Users**.
   a. In the left pane of the Edit Credential window, you can filter by Users, Groups, and Custom Groups. From the drop-down menu, select **Users**, **Groups**, or **Custom Groups**.
      You can also filter using the search.
   b. To add a selected user, group, or custom group to the **Selected Identities** pane, click **Add arrow**.
      To remove a selected user, group, or custom group from the **Selected Identities** pane, click the **Remove arrow**.
   c. Click **OK**.
4. Click **Save**.

**Delete Credentials**

Perform the following tasks in the Credentials for Domain view.
Right-click the credential that you want to delete, and select **Delete**. Or select 📱 on the taskbar.

In the Delete window, this message is displayed: “Are you sure you want to delete the credential for the identity "name"?”

Click **Yes**.

**Delete Encryption Domains**

Encryption domains cannot be deleted in the current release. If you try to delete an Encryption domain, you receive the following message: You cannot delete the encryption domain named 'domain-name'. Libraries associated with this domain will need to be recreated if the domain is deleted.

**Manage Caslibs That Are Enabled for Encryption**

For additional information, see “Working with SAS Data Explorer” in SAS Data Explorer: User’s Guide.

**Add a New Caslib That Connects to a Remote File System**

SAS Data Explorer enables you to discover data and copy it to a SAS Cloud Analytic Services (CAS) server. In this section, we use the SAS Data Explorer to add a new caslib that connects to a PATH, HDFS, or DNFS file system. Only these types of file systems can be encrypted. For detailed information about using SAS Data Explorer, see, “Working with SAS Data Explorer” in SAS Data Explorer: User’s Guide.

1. In the applications menu (═), under **Administration**, select **Manage Environment**.

2. In the navigation bar, click **Data**.

3. Click the **Data Sources** tab. The **Data Sources** tab enables you to add a new caslib that connects to a DNFS, HDFS, or PATH-based remote file system. For more information, see “SAS Data Explorer and the Choose Data Window” in SAS Data Explorer: User’s Guide.

4. Click 📁 on the **Data Sources** tab. The Connection Settings dialog box is displayed.

5. Enter a name for the caslib in the **Name** field. If you change the name, follow the conventions for caslib names as described in “Names for Caslibs, Tables, and Columns” in SAS Data Explorer: User’s Guide. The target operating system and data source might have additional constraints on these names.

6. Accept the default CAS server or select another CAS server in the **Server** field.

7. Select **File System** in the connection **Type** field.

8. Select the remote file system that you want to access in the **Select source type** field: DNFS, HDFS, or PATH.

9. Select the **Persist this connection beyond the current session** check box to add a global caslib for this connection. Deselect this check box to add a session-based caslib for this connection. For more information about this option, see “Caslibs on the Data Sources Tab and Import Tab” in SAS Data Explorer: User’s Guide.

10. Under the **Settings** tab, enter the physical path to the remote file system in the **Path** field.

   If the new caslib is global in scope, you can specify one additional directory level to an existing path that does not exist on the target file system, and that directory will be created. If the new caslib is a session-based caslib, you must specify an existing path, or the connection fails. For more information about global and session-based caslibs, see “Caslibs on the Data Sources Tab and Import Tab” in SAS Data Explorer: User’s Guide.
11 Enter a Description for the connection, if desired.

12 Click Test Connection to test your connection. You will see the green check box and the message "The connection was successful."

13 Click the Advanced tab to specify an Encryption domain for the current data source. From the drop-down list, select an Encryption domain.

   Note: If no encryption domains are shown in the drop-down list, you can create one. See “Create a New Encryption Domain” on page 3.

14 Click Test Connection to test your connection. You will see the green check box and the message "The connection was successful."

15 Click Save to save your connection.

   If the connection succeeds, tables that you are authorized to access in the remote directory will be available from the caslib that you specified in Step 4. Information about caslibs and tables on the Available and Data Sources tabs is stored in the cache for your web browser. If you think this information does not reflect the current state of your system, click in the nearest toolbar.

   If the connection fails, see “General Usage Notes” in SAS Data Explorer: User’s Guide.

16 If the target caslib is not visible from your current view, click to find the caslib and its tables on the specified CAS server.

17 You can see the options that are available to manage this new caslib, which is now available to receive encrypted tables. See “Working with Caslibs” in SAS Data Explorer: User’s Guide.

View the Properties of the New Caslib

When you select a caslib on the Data Sources tab, properties for that caslib are displayed on the right. These properties were specified when the caslib was added, such as the name, the CAS server, the source type, and the domain name.

1 Click the Data Sources tab.

2 From the drop-down list, select the CAS server where your new caslib exists.

3 Scroll down to find the caslib that you created and click on the name of the caslib to display the properties of the caslib. The Authentication domain property should show the name of the encryption domain that you are using.

Edit Properties of the Caslib

You can change the caslib path or change the description of a caslib. The new path uses the same encryption domain defined in this caslib. It is not possible to change the encryption domain assigned to a caslib.

Note: Tables created in the previous topics are still encrypted and require the same key. Changing the path for a caslib with encryption enabled is not recommended. It is recommended that you create a new caslib using the same encryption domain and include a new path.

1 Right-click, and select Edit from the drop-down list.

2 In the Edit Properties window, change the caslib path or description as needed.

3 Click Save.

When editing a caslib the following restrictions apply:

- Only path-based libraries can be edited.
You cannot edit a personal caslib.

You cannot change the encryption domain assigned to a caslib. If you need to change the encryption domain, you must create a new caslib.

**View and Edit Caslib Authorization**

To manage tables (promote, create, drop, edit), delete source tables, alter tables and caslibs, and manage access to a selected caslib, you need the appropriate authorization. You can view your permissions and edit your permissions for a specified caslib. For additional information, see *Working With Data in CAS* and *CAS Authorization*.

View the authorization level for your caslib.

1. Click the **Data Sources** tab.
2. Scroll down to find the caslib that you are working with. Right-click, and select **View Authorization** from the drop-down list.
   
   The levels of authorization are shown for your caslib.
3. Select **Close**.

Change the access level of the caslib.

1. Right-click, and select **Edit Authorization** from the drop-down list.
   a. Change the access level by sliding the control bar to the level of control that you want.
   b. Click **Save**.
   c. Click **Close**.

From both the View Authorization and the Edit Authorization windows, you can add and remove identities from your caslib.

1. Click the **Data Sources** tab.
2. Right-click your caslib and select **Edit Authorization** from the drop-down list.
3. From the **Edit Authorization** window, click **🔍**.
   a. In the left pane of the Choose Identities window, you can filter by Users, Groups, and Custom Groups. From the drop-down menu, select **Users**, **Groups**, or **Custom Groups**.

   You can also filter using the search **🔍**.

   Note: A best practice is to use a custom group. Then you can add additional users to this custom group as needed to grant access to the data. Be sure to assign correct permission for this custom group in the associated caslib Authorization.

   b. Move the selected user, group, or custom group to the **Selected Identities** pane. Click **🚀**.

   c. Click **OK**.
4. Click **Close**.
Manage Tables

A caslib can contain a mix of encrypted and unencrypted tables. However, loading any table still requires the user to be a member of the encryption domain.

Import an Unencrypted Table into a Caslib Enabled for Encryption

To import tables into your caslib, use the Add to Import feature. The Add to import option enables you to copy a table or a file on the Available tab or the Data Sources tab to a global caslib. You can copy a table or a file from any caslib to any global caslib to which you have access. For details, see, “Copying Data from the Available Tab or Data Sources Tab” in SAS Data Explorer: User’s Guide.

Add tables to the queue that you want to import into your caslib.

1 In the Data window, display the window that contains the Available tab or the Data Sources tab. See “SAS Data Explorer and the Choose Data Window” in SAS Data Explorer: User’s Guide.

2 Click the Available tab or the Data Sources tab.

3 Right-click a table or file that you want to copy and select Add to import. The selected table or file is added to the queue on the Import tab. The table or file that you want to copy appears on the left side of the Import tab view. Import properties for the copy appear on the right side.

4 The Target table name field is set to the default target location. For our example, accept this name as appropriate for the copy.

5 Make sure that the Target desintation field contains the path to the caslib where you want to store the copy of the selected table. You can change the default destination by selecting a caslib either by clicking on Find or on the Select Destination icon. From the Data Sources tab, you can then set this new destination as the default by right-clicking on the caslib and selecting Set as default target location.

   - If the source caslib is global in scope, the name of the source caslib is copied into the Target destination field.
   - If a caslib has been set as the default target location, that caslib is selected as the default target destination.
   - If a default cas-shared-default server is deployed with the Public caslib, the Public caslib is set as the default target destination.

For more information, see “Copy Data from One Caslib to Another” in SAS Data Explorer: User’s Guide.

6 Specify which action the import operation should take if the target filename exists in the caslib that is specified in the Target destination field. The options are to cancel the import or to replace the existing item that has the same name.

7 You can right-click the table or file to be copied and select Import Item. If the import succeeds, a copy of the table or file is loaded into memory on the CAS server that is specified in the caslib.

   The copy of the table or file can now be selected from the Available tab or the Data Sources tab. Click the Find button next to the Target table name field to automatically display the imported table in the Data Sources tab.

   If the import fails, see “General Usage Notes” in SAS Data Explorer: User’s Guide.

    Note: You will see a note that states that the item is being imported. When it is complete, you will see the following message: “The table was successfully imported on Date/Time and is ready for use.”

8 Select the Data Sources tab and scroll to the caslib that you imported tables into.
A list of the tables that are imported into the caslib are displayed. If the target caslib is not visible from your current view, click to find the caslib and its tables on the specified CAS server. Select the green arrow to see the tables that were imported into the caslib.

Click on any of the listed tables to view the details of the imported tables.

**How To (Programming Tasks)**

SAS Cloud Analytic Services supports encryption of SASHDAT files at the file level. As an administrator, you might want to simplify encryption for data at rest by configuring caslibs with an encryption password so that all files in a directory are encrypted.

For information that describes how to set an encryption password in a program, see CASLIB Statement, DATASOURCE options in *SAS Cloud Analytic Services: User’s Guide*. For an example of using the CASLIB statement to encrypt caslibs, see *Encrypt Tables in a Caslib*.

**Encryption for Data at Rest: Concepts**

**Overview**

When a caslib is created and enabled for encryption (an encrypted domain is associated), a table imported into that caslib is then encrypted in SASHDAT (.sashdat) format. A domain is associated with a caslib to provide access. Domains are used to store both the credentials (passwords and keys) that are required to access external data sources and the identities that are allowed to use those credentials.

When the encrypted tables are loaded into CAS (in-memory tables), these tables are identified as using an AES encrypted source. However, in-memory tables are not encrypted. The encryption applies to source tables, not to tables that are in memory.

You can specify encryption for a caslib only when it is created. To change it you must re-create the caslib.

SAS Viya supports encrypting files at rest in a path location. Only path-based (PATH, DNFS, HDFS) caslibs are supported.

SAS Viya uses Advanced Encryption Standard (AES) algorithms with 256-bit keys to encrypt data at rest.

Refer to *SAS Cloud Analytic Services: Fundamentals* if you need additional background on data and caslib concepts.

**What Is a Domain?**

**Overview**

Domains are used to store both the credentials (passwords and keys) that are required to access external data sources (for example, databases like Oracle, Teradata, and other data sources like Facebook and Amazon) and the identities that are allowed to use those credentials. A domain contains one or more references to identities (users or groups) who have access to the credentials in the domain. A user can gain access to the credentials either directly with their user ID or indirectly as a member of a group that is defined as an Identity.

The ID, or name, of a Domain is used in the definition of a non-path-based caslib to access and load tables from external databases. A domain is associated with a caslib to provide access. Examples of external sources
include SAS LASR, Oracle, Teradata, Hadoop, Postgres, and Impala. Users of a caslib with an associated
domain do not have to know or enter data source credentials to access or load external data.

There are three Domain types: Authentication, Connection, and Encryption.

**What Is an Encryption Domain?**

An encryption domain is used to store an encryption key that is required to read data at rest in a path assigned
to a caslib. The Identities selected in this encryption domain have access to the key. When you create a path-
based caslib, you can choose to enable encryption. You then select an encryption domain to assign an
encryption key. Tables imported to this caslib are now encrypted. If the path contains preexisting tables, those
tables are not encrypted. Users who are not defined in Identities as individuals or as members of a group cannot
load data from this caslib.

**Note:** It is recommended that you start with an empty caslib and import either encrypted or unencrypted tables.
Mixing these types is not recommended.

Encryption domains are used to store encryption keys that can then be associated with a caslib type of PATH,
HDFS, or DNFS.

**What Is a Connection Domain?**

A connection domain is used when the external database has been set up to require a user ID but no password.
For example, a third-party database like Hadoop might be configured with accounts for authentication that do not
require a password. For information about using connection domains, see “External Credentials: Overview” in
SAS Viya Administration: External Credentials.

**What Is an Authentication Domain?**

An authentication domain is a name that facilitates the matching of logons with the servers for which they are
valid. Authentication domains are used to store credentials that are used to access an external source (for
example, an Oracle database) that can then be associated with a caslib of the appropriate type. They can also
be used for batch processing and scheduling where you store your credentials in a domain to run jobs in batch
mode.

Each user ID and password is valid within a specific scope. For example, the user ID and password that you use
to log on to your computer at work are probably not the same as the user ID and password that you use to log
on to a personal computer at home. It is also common for database servers and web servers to have their own
authentication mechanisms, which require yet another, different, user ID and password.

The software attempts to use only the credentials that it expects to be valid for a particular resource or system.
The software’s knowledge of which credentials are likely to be valid is based entirely on authentication domain
assignments. For this reason, you must correctly assign an authentication domain to each set of resources that
uses a particular authentication provider, and also assign that same authentication domain to any stored
credentials that are valid for that provider.

Authenticating to SAS can be done through SAS logon. For more information, see “Authentication: Overview” in
SAS Viya Administration: Authentication.

For information about using Authentication Domains, see “External Credentials: Overview” in SAS Viya
Administration: External Credentials.

**Defaults**

In a new deployment, encryption for data at rest is not automatically enabled. You can configure encryption of
data that is added to PATH, HDFS, and DNFS caslibs by using SAS Environment Manager and the
programming interfaces.
Encrypting Caslibs

SAS Viya supports encryption as an option for tables in caslibs. The encryption applies to source tables, not to tables that reside in memory. When you create a caslib, you enable the caslib to receive encrypted tables when you assign an encryption domain. The tables that you import into the caslib become encrypted and all tables use the same encryption key.

When you import a table (SAS table, .csv file, .txt file, Excel file, and so on) into a caslib, a .sashdat file is created in the same path location. If the caslib is enabled for encryption, those .sashdat files at rest are now encrypted. When these tables are loaded into CAS, these in-memory CAS tables are not encrypted. You must have authorization to access the table in the caslib. Table access can be inherited from the caslib, but it can also be granted or denied at the table level.

If you delete the caslib, the tables in the associated path remain encrypted. To access those tables again, you must create a new caslib, enable encryption, and use the same encryption domain with the same key value, or create a new encryption domain with the same key value.

Considerations for Encrypting Tables in Caslibs

Here are a few best practices and considerations when encrypting data at rest:

- Only PATH, HDFS, or DNFS files can be encrypted.
- It is best not to mix encrypted and unencrypted tables in a caslib path. Only the user IDs and groups in the domain identities are able to read the encrypted data.
  
  When you create a new caslib and enable encryption, only the newly imported tables are encrypted and stored in the path. A best practice is to first make sure that the path is empty before you import the tables that you want encrypted.

- Encryption of data at rest has some performance costs, and user and administrative overhead. You must balance the goals of security and performance at your site when deciding to encrypt data. Users and administrators must keep track of keys (passphrases and passwords) when accessing the data. The system uses additional CPU resources when loading and saving encrypted tables.

Reference

PROC PWENCODE

In the programming environment, the ENCRYPTIONPASSWORD= option in the CASLIB statement specifies a password for encrypting or decrypting tables. For additional password security, you can use the PWENCODE procedure to encode that password. Encoded passwords can be used in place of plaintext passwords in SAS programs that access databases and various servers. For information, see PWENCODE Procedure.