Administering Models: Overview

This document covers how to configure access to models and the modeling integration points between the common model repository and SAS Viya products. The Model Repository service provides access to a common model repository for SAS applications, and enables users to perform the following actions:

- import models into SAS Model Manager
- register models from Model Studio, SAS Visual Analytics, and SAS Studio into a common model repository
- add a model from the common model repository into a decision flow in SAS Decision Manager and SAS Intelligent Decisioning
- publish models or decisions to SAS Cloud Analytic Services (CAS), Hadoop, Teradata, and the SAS Micro Analytic Service
Modeling Integration Points

Here are the integration points for the common model repository:

CAS
CAS stores analytic store files in the ModelStore caslib. Users can also publish models from SAS Model Manager and Model Studio to a CAS publishing destination.


Common Model Repository
The common model repository contains multiple repositories that are used to store models that have been registered from SAS Model Manager, Model Studio, SAS Studio, and SAS Visual Analytics. The models can then be accessed from other SAS applications, such as SAS Intelligent Decisioning, or published to external publishing destinations.


Model Studio
A suite of SAS products that enables you to build SAS Visual Data Mining and Machine Learning models as well as SAS Text Analytics models and then to register them into the common model repository. SAS Visual
Data Mining and Machine Learning models can also be published from Model Studio to a configured published destination.


**SAS Drive**
A hub for the SAS Viya applications that enables you to easily view, organize, and share your content from one place. The availability of the features in SAS Drive depends on the applications that have been installed and the features and permissions that have been specified by your administrator.

See *SAS Drive: Getting Started*

**SAS Intelligent Decisioning**
SAS web application that enables you to combine analytical models, rule sets, and conditional logic into decisions. You can investigate various scenarios, test and refine the decision logic, and then publish the decisions for use in batch applications and online transactions. After a decision has been published, it is available for use by other applications. See *SAS Intelligent Decisioning: User’s Guide*.

**SAS Model Manager**
SAS web application that enables you to store and manage models in a common model repository, as well as to organize them within projects and folders. You can import models that you developed using a SAS application (such as Model Studio, SAS Visual Analytics, and SAS Studio) as well as SAS code and PMML models. You can also create a new model with the model’s files in a folder or project. Models that are located within a project can be evaluated for champion model selection, monitored for performance, and then published to a configured publishing destination that can be defined for CAS, Hadoop, SAS Micro Analytic Service, and Teradata.


**SAS Studio**
SAS development application for running SAS programs, which enables users to use macros to create, update, and delete objects within the common model repository. A user can also use the Register task to import a scoring model from SAS Studio into a SAS Model Manager project that is located within the common model repository. A scoring model is an analytic object in a CAS table. Scoring models can be created using several SAS Studio tasks such as the Forest task.


**SAS Visual Analytics**
SAS web application that enables you to explore, discover, and predict using your data. If SAS Visual Statistics is licensed at your site, then you can create, test, and compare models based on the patterns that are discovered during exploration of your data. You can export a model before or after performing model comparison in order to use it with other SAS products in a production environment. If SAS Visual Data Mining and Machine Learning is licensed at your site, then additional models are available. You can also register SAS Visual Data Mining and Machine Learning and SAS Visual Statistics models from SAS Visual Analytics to the common model repository.


---

**Access to Models**

**Overview**
Registered models are stored within the repository folders that reside within the Model Repositories content root folder. Initially, models that are registered to the SAS Model Manager default repository and standard repositories are accessible by authenticated users. See “Reference: Initial Access to Registered Models”.

Model Studio projects contain models and are stored beneath a user’s folder in My Folder. Initially, Model Studio projects and models are private. For more information, see the following documentation:


You can create the standard repositories before they have been deployed using the SAS Model Manager web application or the %MM_CREATE_REPOSITORY macro. If you do not have SAS Model Manager, you can create the initial reserved repositories by registering a model from a supported product to the common model repository. You can also use the Model Repository API to submit a request. The initial default repositories and standard repositories can initially be accessed by all authenticated users.

Important: The May 2019 release of SAS Viya 3.4 contains updates to the Model Repository service, which includes changes to authorization rules for endpoints and repository folders. Only SAS Administrators and other authorized users can create, update, or delete repository folders. In addition, custom repositories can no longer be accessed by authenticated users. A SAS Administrator must grant user’s access to custom repositories. The name of the default repository for new installations has also changed from Repository 1 to Public.

CAUTION! Projects and models that are registered into a standard repository should not be moved using content selection windows within SAS Environment Manager and other SAS web applications. Moving a project or model that was registered from Model Studio breaks the connection between SAS Model Manager and Model Studio. Models that are copied to another folder do not retain the connection to the model in Model Studio.

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>SAS Model Manager default repository for authenticated users to create projects and import models. The initial default repository is predefined as Public or Repository 1.</td>
</tr>
<tr>
<td>DMRepository</td>
<td>SAS Visual Data Mining and Machine Learning models registered from Model Studio to the common model repository are stored within the DMRepository. The DMRepository folder is created the first time a SAS Visual Data Mining and Machine Learning model is registered from Model Studio to the common model repository.</td>
</tr>
<tr>
<td>VARepository</td>
<td>SAS Visual Data Mining and Machine Learning and SAS Visual Statistics models registered from SAS Visual Analytics to the common model repository are stored within the VARepository. The VARepository folder is created the first time a SAS Visual Analytics report object is registered as a model to the common model repository.</td>
</tr>
<tr>
<td>VTARepository</td>
<td>SAS Visual Text Analytics models registered from Model Studio to the common model repository are stored within the VTARepository. The VTARepository folder is created the first time a SAS Visual Text Analytics model is registered from Model Studio to the common model repository.</td>
</tr>
</tbody>
</table>

Here are the initial authorizations for the Model Repositories folder, the default repository folder, the standard repository folders, and a custom repository folder.

Note: The SAS Demo User is an example of a user that assumed the SAS Administrators group when signing in to SAS Environment Manager or another SAS web application.

The following figure is an example of the initial authorization for the Model Repositories folder.
**Figure 1**  Authorization for the Model Repositories Folder

<table>
<thead>
<tr>
<th>Principal</th>
<th>Read</th>
<th>Update</th>
<th>Delete</th>
<th>Secure</th>
<th>Add</th>
<th>Remove</th>
<th>Read (convey)</th>
<th>Update (convey)</th>
<th>Delete (convey)</th>
<th>Secure (convey)</th>
<th>Add (convey)</th>
<th>Remove (convey)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticated Users</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>SAS Administrators</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>SAS Demo User</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

The following figure is an example of the initial authorization for the default repository (Public or Repository 1) and standard repository (DMRepository, VARervoiritory, and VTAREpository) folders.

**Figure 2**  Initial Authorization for the Default Repository and the Standard Repository Folders

<table>
<thead>
<tr>
<th>Principal</th>
<th>Read</th>
<th>Update</th>
<th>Delete</th>
<th>Secure</th>
<th>Add</th>
<th>Remove</th>
<th>Read (convey)</th>
<th>Update (convey)</th>
<th>Delete (convey)</th>
<th>Secure (convey)</th>
<th>Add (convey)</th>
<th>Remove (convey)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticated Users</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>SAS Administrators</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SAS Demo User</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

The following figure is an example of the initial authorization for a custom repository folder.

**Figure 3**  Initial Authorization for a Custom Repository Folder

<table>
<thead>
<tr>
<th>Principal</th>
<th>Read</th>
<th>Update</th>
<th>Delete</th>
<th>Secure</th>
<th>Add</th>
<th>Remove</th>
<th>Read (convey)</th>
<th>Update (convey)</th>
<th>Delete (convey)</th>
<th>Secure (convey)</th>
<th>Add (convey)</th>
<th>Remove (convey)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticated Users</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>SAS Administrators</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SAS Demo User</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
How to Narrow Access to Models in Standard Repositories

Overview
Perform the following tasks to narrow access to models in standard repositories:

1. Create custom groups.
2. Check whether the licensed products exist in your SAS Viya deployment.
3. Create the standard repositories.
5. Reduce the availability of model applications and features.

Create Custom Groups
Create two new custom groups as follows:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>_modelers</td>
<td>Modelers</td>
<td>Anyone who uses model functionality.</td>
</tr>
<tr>
<td>_modelRepoAdmins</td>
<td>Model Repository Administrators</td>
<td>Anyone who should be able to add, update, and delete repositories.</td>
</tr>
</tbody>
</table>

Note: The names and IDs are suggested values. You can specify different values.

Note: If you have planned to use separated access, where different groups access different models, create additional groups as members of the Modelers group.

Check Whether the Licensed Products Exist in Your SAS Viya Deployment

1. Select to view the Licensed Products page.
2. Enter SAS Model Manager in the Product filter field.
3. Enter the following licensed products names in the Product filter field:

<table>
<thead>
<tr>
<th>Standard Repository</th>
<th>Licensed Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMRepository</td>
<td>SAS Data Mining and Machine Learning</td>
</tr>
<tr>
<td>VARRepository</td>
<td>SAS Visual Analytics Explorer</td>
</tr>
<tr>
<td>VTARepository</td>
<td>Text Analytics UI Setinit</td>
</tr>
</tbody>
</table>

TIP If the licensed product does not exist, then you do not need to create the associated standard repository or to secure access to the repository.
Create the Standard Repositories

Not all deployments include all repositories. The default repository is included for all installations. The standard repositories are not available until you have a license for the associated product and a SAS Administrator creates them or an authorized user registers a model from Model Studio or SAS Visual Analytics.

On the Content page in SAS Environment Manager, verify that the following repository folders do not already exist directly beneath the Model Repositories root folder. If they already exist, you do not need to create the standard repositories. You can continue to the next step and secure the Model Repositories folder and its immediate children.

- VARepository
- DMRepository
- VTARepository

If you have a license for SAS Model Manager, and the standard repositories do not already exist, a SAS Administrator can create them using the SAS Model Manager web application or the %MM_CREATE_REPOSITORY macro. When creating the standard repositories before they have been deployed, the names must match exactly. See “Create a New Repository using SAS Model Manager”.

If you do not have a license for SAS Model Manager, then the standard repositories can be created by registering models from Model Studio and SAS Visual Analytics. See “Create Standard Repositories by Registering Models”.

Secure the Model Repositories Folder and Its Immediate Children

1. In SAS Environment Manager, click to view the Content page.

2. Locate the Model Repositories folder. It is a top-level folder in the Content tree. Right-click the folder and select Edit authorization. Adjust access as follows:

   a. Click

   b. Select the Modelers and Model Repository Administrators custom groups as identities. Click OK.

      - For the Model Repository Administrators group, grant object permissions of Read, Update, Add, and Remove, as well as grant container (convey) permissions of Read, Update, Delete, Add, and Remove.
      - For the Modelers group, grant object permissions of Read and Add.

Here are the authorizations for the Model Repositories folder, after it has been secured.
Right-click a repository (any immediate child of the Model Repositories folder) and select Edit authorization.

a. Click 🔄.

b. Select the Modelers custom group as an identity. Click OK.

- Remove the grant setting for all object permissions for Authenticated Users. In each cell, change the Direct setting value to None.

  Note: Do not change the Direct setting value to Prohibit.

- For the Modelers group (or, if you are securing a custom repository, which is a more specific group), add object grants of Read, Add, and Remove and container (convey) grants of Read, Update, Delete, Add, and Remove.

Here is an example of the authorizations for the VARRepository folder.

<table>
<thead>
<tr>
<th>Principal</th>
<th>Read</th>
<th>Update</th>
<th>Delete</th>
<th>Secure</th>
<th>Add</th>
<th>Remove</th>
<th>Read (convey)</th>
<th>Update (convey)</th>
<th>Delete (convey)</th>
<th>Secure (convey)</th>
<th>Add (convey)</th>
<th>Remove (convey)</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧙 Authenticated Users</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>🧙 Model Repository Admin…</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>🧙 Modelers</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>🧙 SAS Administrators</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>🧙 SAS Demo User</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
</tbody>
</table>

Repeat this step for each repository (each immediate child of the Model Repositories folder). Here are key points:
The DMRepository, VAREpository, and VTARepository model repositories are created automatically the first time a user registers a model. It is recommended that an administrator create those repositories and adjust access before the deployment is made available to users. For details, see “Create the Standard Repositories”.

Whenever a custom repository is added, access to that repository must be adjusted. (For that reason, these instructions allow only Model Repository Administrators to add repositories.) For more information, see “How to Narrow Access to Models in a Custom Repository” on page 9.

Reduce the Availability of Model Applications and Features
The purpose of this task is to enhance usability rather than to increase security.

1. In SAS Environment Manager, select the Content page.

2. In each rule in the following table, replace the original principal (principalType=Authenticated Users) with the _modelers group (principalType=group, name=_modelers).

<table>
<thead>
<tr>
<th>Application or Feature</th>
<th>Target URI (or URIs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to use model objects in SAS Visual Analytics</td>
<td><code>/SASVisualAnalytics_capabilities/buildAnalyticalModel</code></td>
</tr>
<tr>
<td>Ability to access SAS Model Manager</td>
<td><code>/SASModelManager/</code> and <code>/SASModelManager</code></td>
</tr>
<tr>
<td>Ability to access Model Studio**</td>
<td><code>/ModelStudio/**</code></td>
</tr>
</tbody>
</table>

* These are two alternate forms of the same URI. In the current release, because SAS Model Manager uses both forms, both rules are necessary.

** In the current release, you cannot limit direct access to Model Studio. The adjusted rule limits access to Model Studio only from the applications menu.

How to Narrow Access to Models in a Custom Repository

Overview
Perform the following tasks to narrow access to models in a custom repository:

1. Create custom groups.
2. Check whether the licensed product exists in your SAS Viya deployment.
3. Create a custom repository.
5. Add model repository rules for a custom repository.
6. Reduce the availability of model applications and features.

Create Custom Groups
Create two new custom groups as follows:
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>_modelers</td>
<td>Modelers</td>
<td>Anyone who uses model functionality.</td>
</tr>
<tr>
<td>_modelRepoAdmins</td>
<td>Model Repository Administrators</td>
<td>Anyone who should be able to add, update, and delete repositories.</td>
</tr>
</tbody>
</table>

Note: The names and IDs are suggested values. You can specify different values.

Note: If you have planned to use separated access, where different groups access different models, create additional groups as members of the Modelers group.

**Check Whether the Licensed Products Exist in Your SAS Viya Deployment**

1. Select ☐ to view the **Licensed Products** page.
2. Enter SAS Model Manager in the **Product** filter field.

**Create a Custom Repository**

A SAS Administrator can create a custom repository using the SAS Model Manager web application or the %MM_CREATE_REPOSITORY macro. See “Create a New Repository using SAS Model Manager”.

**Secure the Model Repositories Folder and Its Immediate Children**

1. From the applications menu (≡), select **Manage Environment**.
2. In SAS Environment Manager, click ☐ to view the **Content** page.
3. Locate the **Model Repositories** folder. It is a top-level folder in the **Content** tree. Right-click the folder and select **Edit authorization**. Adjust access as follows:
   a. Click ☐.
   b. Select the **Modelers** and **Model Repository Administrators** custom groups as identities. Click **OK**.
      - For the Model Repository Administrators group, grant object permissions of Read, Update, Add, and Remove, as well as grant container (convey) permissions of Read, Update, Delete, Add, and Remove.
      - For the Modelers group, grant object permissions of Read and Add.

Here are the authorizations for the **Model Repositories** folder, after it has been secured:
4 Right-click a repository (any immediate child of the Model Repositories folder) and select **Edit** authorization.

   a Click 🧑.

   b Select the Modelers custom group as an identity. Click **OK**.

   For the Modelers group (or, if you are securing a custom repository, a more specific group), grant object permissions of Read, Add, and Remove, as well as container (convey) grants of Read, Update, Delete, Add, and Remove.

   Here is an example of the authorizations for the custom repository folder:

   ![Table](image)

   Repeat this step for each repository (each immediate child of the Model Repositories folder).

   Whenever a custom repository is added, access to that repository must be adjusted. (For that reason, these instructions allow only Model Repository Administrators to add repositories.) For more information, see “How to Narrow Access to Models in a Custom Repository” on page 9.
Add Model Repository Rules for a Custom Repository

In order to perform certain actions on objects (such as running a scoring test or updating a project property) within a custom repository, additional rules must be added.

1. In SAS Environment Manager, click to display the Rules page.

2. Click  

3. Add additional Model Repository rule for model object access.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object URI</td>
<td>/modelRepository/models/*</td>
</tr>
<tr>
<td>Principal type</td>
<td>Modelers</td>
</tr>
<tr>
<td>Rule type</td>
<td>Grant</td>
</tr>
<tr>
<td>Permissions</td>
<td>Create</td>
</tr>
<tr>
<td>Reason</td>
<td>Provides ability to create models.</td>
</tr>
</tbody>
</table>

4. Add additional Model Repository rule for project object access.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object URI</td>
<td>/modelRepository/projects/*</td>
</tr>
<tr>
<td>Principal type</td>
<td>Modelers</td>
</tr>
<tr>
<td>Rule type</td>
<td>Grant</td>
</tr>
<tr>
<td>Permissions</td>
<td>Create and Update</td>
</tr>
<tr>
<td>Reason</td>
<td>Provides ability to create and update projects.</td>
</tr>
</tbody>
</table>

For more information, see “Add a Rule” in SAS Viya Administration: General Authorization.

Reduce the Availability of Model Applications and Features

The purpose of this task is to enhance usability rather than to increase security.

1. In SAS Environment Manager, select the Content page.

2. In each rule in the following table, replace the original principal (principalType=Authenticated Users) with the _modelers group (principalType=group, name=modelers).
How to Create Repositories

There are a few different ways to create a new repository. If you have a license for SAS Model Manager, the easiest way to create a new custom repository or a standard repository is to use the SAS Model Manager web application or the %MM_CREATE_REPOSITORY macro. If you do not have a license to SAS Model Manager, you can register a model from Model Studio or SAS Visual Analytics to create the standard repositories.

For background information, see “Generated Rules for Each Repository”.

Create a New Repository using SAS Model Manager

To create a new custom or standard repository using the SAS Model Manager web application:

1. From the applications menu (≡), select Manage Models.
2. Click your name in the application bar and select Settings ⇒ SAS Model Manager ⇒ Repository.
3. Click .
4. Enter a name for the repository.
   **Important:** If you are creating a standard repository for use with Model Studio or SAS Visual Analytics, the name must match exactly what is in the Table 1 on page 4. You can name a custom repository anything you want, as long as it does not already exist.
5. (Optional) Enter a description of the repository.
   **Note:** After you save the new repository, the description cannot be edited.
6. Click Save.
7. Click Close.

See Also

“%MM_CREATE_REPOSITORY Macro” in SAS Model Manager: Macro Reference

Create Standard Repositories by Registering Models

If you do not have a license to SAS Model Manager, you can create the standard repositories by registering models from Model Studio and SAS Visual Analytics.

VARespository

1. On the Content page in SAS Environment Manager, verify that a VARespository folder does not already exist directly beneath the Model Repositories root folder.
2. From the applications menu (≡), select Explore and Visualize Data.
3. In a new or existing report, add an object from either the SAS Visual Statistics or SAS Visual Data Mining and Machine Learning headings.

---

<table>
<thead>
<tr>
<th>Application or Feature</th>
<th>Target URI (or URIs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to access SAS Model Manager</td>
<td>/SASModelManager/ and /SASModelManager*</td>
</tr>
</tbody>
</table>

* These are two alternate forms of the same URI. In the current release, because SAS Model Manager uses both forms, both rules are necessary.
Note: If those categories are not listed (and you are signed in as an administrator), a **VARepository** folder cannot be created in your deployment. Skip the remaining steps for creating the repository folder and for securing access to it.

4 Assign data items to the required data roles.

   **TIP** If you need additional information, see *SAS Visual Analytics: Working with SAS Visual Data Mining and Machine Learning* or *SAS Visual Analytics: Working with SAS Visual Statistics*.

5 Right-click the object and select **Register model**.

6 On the **Content** page in SAS Environment Manager (**Manage Environment**), verify that a **VARepository** folder was created beneath the Model Repositories root folder.

**DMRepository**

1 On the **Content** page in SAS Environment Manager, verify that a **DMRepository** folder does not already exist directly beneath the **Model Repositories** folder.

2 From the applications menu (**Ξ**), select **Build Models**.

   **Note:** If your deployment does not include Model Studio, your deployment cannot include a **DMRepository** folder. Skip the remaining steps for creating the repository folder and for securing access to it.

3 In Model Studio, create a project of the type **Data Mining and Machine Learning**. Include at least one model. Run the pipeline.

   **TIP** If you need additional information, see *SAS Visual Data Mining and Machine Learning: User’s Guide*.

4 On the **Pipeline Comparison** tab, select **Register models**.

5 On the **Content** page in SAS Environment Manager (**Manage Environment**), verify that a **DMRepository** folder was created beneath the **Model Repositories** folder.

**VTARepository**

1 On the **Content** page in SAS Environment Manager, verify that a **VTARepository** folder does not already exist directly beneath the **Model Repositories** folder.

2 From the applications menu (**Ξ**), select **Build Models**.

   **Note:** If your deployment does not include Model Studio, your deployment cannot include a **VTARepository** folder. You can skip this step.

3 In Model Studio, create a project of the type **Text Analytics**.

   **Note:** If the project type **Text Analytics** is not available, a **VTARepository** folder cannot be created in your deployment. Skip the remaining steps for creating the repository folder and for securing access to it.

   Include at least one model. Run the pipeline.

   **TIP** If you need additional information, see *SAS Visual Text Analytics: User’s Guide*.

4 Right-click the pipeline and select **Register model**.
On the Content page in SAS Environment Manager (Manage Environment), verify that a VTARepository folder was created beneath the Model Repositories folder.

**Reference: Initial Access to Registered Models**

*Note:* Initially, all authenticated users have Read and Write access to all registered models in the default repository and standard repositories. Restrict access as appropriate for your usage patterns and security goals. See “How to Narrow Access to Models in Standard Repositories”.

**Common Model Repository Predefined Rules**

The predefined rules grant Authenticated Users Read and Add access to the default repository and standard model repositories, model projects, and models as follows:

*Note:* The May 2019 release of SAS Viya 3.4 includes updates to the Model Repository service authorization rules for the common model repository.

**CAUTION! The Common Model Repository predefined rules should not be modified.** Authorization changes should be implemented on a repository folder or folders within a repository folder.

Here are the initial authorization rule permissions for the Model Repository service (/modelRepository) endpoints that were implemented in the May 2019 release of SAS Viya. These rules apply to new installations of the SAS Viya deployment.

**Figure 4  Common Model Repository Rules (/modelRepository)**

<table>
<thead>
<tr>
<th>Object URI</th>
<th>Principal</th>
<th>Setting</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>/modelRepository/</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Read</td>
</tr>
<tr>
<td>/modelRepository</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Read</td>
</tr>
<tr>
<td>/modelRepository/repositories</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Read</td>
</tr>
<tr>
<td>/modelRepository/projects/</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Read</td>
</tr>
<tr>
<td>/modelRepository/models/</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Read</td>
</tr>
<tr>
<td>/modelRepository/models/transfer</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Create</td>
</tr>
<tr>
<td>/modelRepository/models</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Create, Read</td>
</tr>
<tr>
<td>/modelRepository/projects</td>
<td>Authenticated Users</td>
<td>Grant</td>
<td>Create, Read</td>
</tr>
<tr>
<td>/modelRepository/repositories/*</td>
<td>SASAdministrators</td>
<td>Grant</td>
<td>Update, Delete</td>
</tr>
<tr>
<td>/modelRepository/repositories</td>
<td>SASAdministrators</td>
<td>Grant</td>
<td>Create</td>
</tr>
</tbody>
</table>

*Note:* In the previous release, an additional predefined wildcard rule that targets the objectURI /modelRepository/** granted full access to a group that has Administrators as its ID. Because no such group exists, the rule has no effect. The rule is unnecessary, because the SAS Administrators group has a universal grant. You can delete the rule or leave it in place.

Here are the initial authorization rules of the Model Repository service (/modelRepository) endpoints that were implemented in the July 2018 release of SAS Viya 3.4. If you made changes to the authorization rules and you update your SAS Viya deployment to the May 2019 release, your changes to the authorization rules are preserved.
Predefined Rules for the Model Repositories Folder

Predefined rules for the Model Repositories folder grant access to Authenticated Users as follows:

<table>
<thead>
<tr>
<th>Target</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Model Repositories/** (folder)</td>
<td>Add, Read</td>
</tr>
<tr>
<td>Object URI:</td>
<td>/folders/folders/folder-ID/**</td>
</tr>
<tr>
<td>Permissions:</td>
<td>Add, Read, Create</td>
</tr>
</tbody>
</table>

Note: After the May 2019 update to the Model Repository service authorization rules, Authenticated Users no longer have permissions that are associated with the predefined rule for the /Model Repositories container URI.

Generated Rules for Each Repository

Two generated rules on each immediate child of the Model Repositories folder grant access to Authenticated Users. For example, generated rules on the VARRepository folder grant access to Authenticated Users as follows:

<table>
<thead>
<tr>
<th>Target</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Model Repositories/VARepository/** (folder)</td>
<td>Add, Remove, Create, Read</td>
</tr>
<tr>
<td>objectURI:</td>
<td>/folders/folders/folder-ID/**</td>
</tr>
<tr>
<td>Permissions:</td>
<td>Update, Delete, Add, Secure, Remove, Create, Read</td>
</tr>
<tr>
<td>Target</td>
<td>/Model Repositories/VARepository (folder)</td>
</tr>
<tr>
<td>containerURI:</td>
<td>/folders/folders/folder-ID</td>
</tr>
<tr>
<td>Permissions:</td>
<td>Update, Delete, Add, Secure, Remove, Create, Read</td>
</tr>
</tbody>
</table>
Configuring Model Data Libraries

About Configuring Model Data Libraries

During the deployment of the Model Repository service, the ModelPerformanceData and ModelStore caslibs are created on each CAS server (for example, cas-shared-default and cas-shared-mpp). The source type for the caslibs is a file system path. Users must have Read and Write permissions to the source file system directory paths.

The ModelPerformanceData caslib stores the performance results data tables that are created when you run a performance job for a model. This caslib can be used only if you have the SAS Model Manager web application and the Model Management service.

The ModelStore caslib contains analytic store files for models that are created using SAS Visual Analytics or Model Studio. The ModelStore caslib must exist on each CAS server in a cluster. If your model references a train table in a library that exists on a server that does not have a ModelStore library, you cannot register the model from Model Studio to the SAS Model Manager common model repository. An error message is displayed.

SAS Model Manager data sources are managed using SAS Environment Manager. You can create libraries (caslibs) and import tables to be used by SAS Model Manager. For more information, see “Making Data Available to CAS” in SAS Data Explorer: User’s Guide.

SAS Model Manager Predefined Caslibs

The following caslibs are automatically created during the deployment of SAS Model Manager. Each caslib has a default assignment and specifications. For a list of the SAS Viya predefined caslibs, see “Predefined Caslibs” in SAS Viya Administration: Data.

<table>
<thead>
<tr>
<th>Caslib</th>
<th>Default Assignment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelPerformanceData</td>
<td>/opt/sas/viya/config/data/cas/default/modelMonitorLibrary/</td>
<td>Stores application-generated performance results data that is used for reporting.</td>
</tr>
</tbody>
</table>
The file system directory is associated with the ModelStore CAS library. This library stores tables (SASHDAT), and each table contains one analytic store. The table is created when an analytic store model is registered to the common model repository. The analytic store file is created in the associated directory.

When an analytic model is set as the champion or published to SAS Micro Analytic Service, the analytic store is extracted and written to the /opt/sas/viya/config/data/modelsvr/astore directory. For more information, see “Configuring Access to Analytic Store Model Files” in SAS Model Manager: Administrator’s Guide.

Important: Caslibs that have a file system directory path for a source type require specific directory permissions. For more information, see “File System Directory Permissions” on page 18.

### File System Directory Permissions

When defining a caslib where the source type is a file system directory path, the appropriate permissions must be granted. For more information, see “CAS Authorization: Host Access Considerations” in SAS Viya Administration: Cloud Analytic Services Authorization.

By default, CAS sessions run using the cas account. The CASHostAccountRequired custom group is a SAS Viya reserved group name, but it is not created during the deployment of SAS Viya. If you add identities or groups to the custom group with the ID CASHostAccountRequired, members of this group automatically run their CAS sessions under their own host account. Users within this group must have Read and Write permissions to caslib file system directory paths in order to register analytic store models and generate performance results. In addition, users are limited to sharing analytic stores depending on their primary group permissions because the analytic store is created with group ownership by the user’s primary group.

Here is a method to configure permissions for the file system directories used by the ModelStore and ModelPerformanceData caslibs:

1. Create a host system group that contains the cas user account and the user accounts for the users who are in the CASHostAccountRequired custom group.
2. Replace the default sas group with the new host system group for both directories.
3. Grant Read and Write group permissions to the new host system group for the directories.
4. Change directories to /opt/sas/viya/config/etc/cas/default.
5. Add Write permissions for the owner to the file casstartup_usermods.lua.

   ```bash
   sudo chmod +w casstartup_usermods.lua
   ```
6. Edit the casstartup_usermods.lua file and add the following code. Be sure to substitute the name of the new host group where specified.

   ```lua
   --Set permissions on the specified Caslibs to allow a host group containing CASHostAccountRequired users to write data.
   CHARLibraries={"modelStore","modelMonitorLibrary"}
   hostgroup="NewHostGroup"
   ```
for k,v in pairs(CHARLibraries) do
  chgrpcmd = string.format("chgrp -R %s %s/%s",hostgroup,env.CASDATADIR,v)
  chmodcmd = string.format("chmod g+w -R %s/%s",env.CASDATADIR,v)
  print(string.format("Executing '%s'",chgrpcmd))
  print(string.format("Executing '%s'",chmodcmd))
  os.execute(chgrpcmd)
  os.execute(chmodcmd)

Note: The changes to the casstartup_usermods.lua file are necessary in order to preserve the modified directory permissions. Otherwise, the changes to the directory permissions would be overwritten by the initial default directory permissions when the CAS server is restarted.

7 Restart the CAS server to test your changes.

For more information, see “The CASHostAccountRequired Custom Group” in SAS Viya Administration: Identity Management and “User Accounts (Reference)” in SAS Viya for Linux: Deployment Guide.

Add ModelStore and ModelPerformanceData Caslibs to a CAS Server

Note: If you add a new CAS server after the deployment of the Model Repository service on SAS Viya, verify that the ModelStore and ModelPerformanceData caslibs exist. If they do not exist, you must add the caslibs to the new CAS server.

1 Sign in to SAS Environment Manager as an administrator.
   
   Note: If you are already logged in to SAS Model Manager, you can access SAS Environment Manager by clicking ☰ and selecting Manage Environment.

2 Click ☰ in the navigation bar. The SAS Data Explorer page appears.

3 Click the Data Sources tab.

4 Click 📘 on the Data Sources tab. The Connections Settings window appears.

5 Enter the name of the library (for example, ModelStore).

6 Select the CAS server (for example, cas-shared-myserver) where you want the new caslib to reside.

7 Select File system from the Type drop-down list.

8 Select the Persist this connection beyond the current session check box to add a global caslib for this connection.

9 Select PATH as the data source type.

10 Enter a value for Path on the Settings tab.

   Note: Use a similar path to the one specified for the ModelStore caslib on the cas-shared-default server.

11 Click Save.

12 Repeat steps 4 through 11 for the ModelPerformanceData caslib.

   Note: If you have users in the CASHostAccountRequired custom group, you must repeat the steps in the “File System Directory Permissions” section.
Configuring Access to Analytic Store Model Files

Mapping Analytic Store Directories

In order to publish analytic store models or decisions that use analytic store models to the SAS Micro Analytic Service publishing destination, the model's analytic store (ASTORE) file must be accessible from the `/models/astores/viya` directory path.

The Compute service extracts the ASTORE file from the analytic store's CAS table in the ModelStore caslib and copies it to `/opt/sas/viya/config/data/modelsvr/astore`.

The service copies the ASTORE file when you do any of the following:

- run a decision test for a decision that uses the analytic store model
- set the analytic store model as a project champion in SAS Model Manager
- publish the analytic store model to SAS Micro Analytic Service from SAS Model Manager or Model Studio

In order to make the model's ASTORE file accessible, you must map the `/opt/sas/viya/config/data/modelsvr/astore` directory on the Compute server to the `/models/astores/viya` file system directory on each server that hosts either SAS Micro Analytic Service or SAS Event Stream Processing.

If the Compute Server and one of the consuming services are on the same server, the mapping can be a symbolic link. In a clustered environment in which the Compute Server and the consuming services are located on different servers, you can use a Network File System (NFS) mount or another shared directory that points to the `/opt/sas/viya/config/data/modelsvr/astore` directory.

Setting Permissions

Note: Users must have Read and Write permissions to the ModelStore caslib file system directory. For more information, see “File System Directory Permissions” on page 18.

Users who need to work with analytic store models must have Read and Write permissions to both the `/opt/sas/viya/config/data/modelsvr/astore` and `/models/astores/viya` directories. By default, the `sas` group has full permission to these directories. However, this group is intended for administrators. For more information, see “User Accounts (Reference)” in SAS Viya for Linux: Deployment Guide.

As an alternative, you can create a new system group and give members of the new group access to these directories.

1. Create a system group that contains the user accounts for users that need to work with analytic store models.
2. Replace the default `sas` group with the new group for both directory paths.
3. Grant members of the group Read and Write permissions for the directory paths.
4. Set the `setgid` bit on the directories.

For more information, see Understanding Linux File Permissions and How to use special permissions: the `setuid`, `setgid` and `sticky` bits.
Creating Analytic Store Directories in a Multi-Tenant Deployment

In a multi-tenant deployment, the steps for mapping analytic store directories and setting permissions on page 20 on the analytic store directories must be performed for each tenant after the tenant is onboarded. However, the directory paths must reflect the tenant name in place of viya.

For example, if the onboarded tenant is named “companyA”, use the following directory paths:

/models/astores/companyA
/opt/sas/companyA/config/data/modelsvr/astore