



SAS[®] Viya[®] 3.5 Administration: Models

Administering Models: Overview	1
Modeling Integration Points	3
Access to Models	5
Overview	5
Initial Access to Registered Models	8
How to Narrow Access to Models in Standard Repositories	10
How to Narrow Access to Models in a Custom Repository	15
How to Create Repositories	20
Configuring Model Data Libraries	23
About Configuring Model Data Libraries	23
SAS Model Manager Predefined Caslibs	23
File System Directory Permissions	24
Add ModelStore and ModelPerformanceData Caslibs to a CAS Server	26
Configuring Access to Analytic Store Model Files	27
Mapping Analytic Store Directories	27
Setting Permissions	27
Configuring Analytic Store Directories in a Multi-Tenant Deployment	28
Promoting Analytic Store Models	28
Configuring Support for Python Code Files	29
Accessing Python Resource Files	29

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

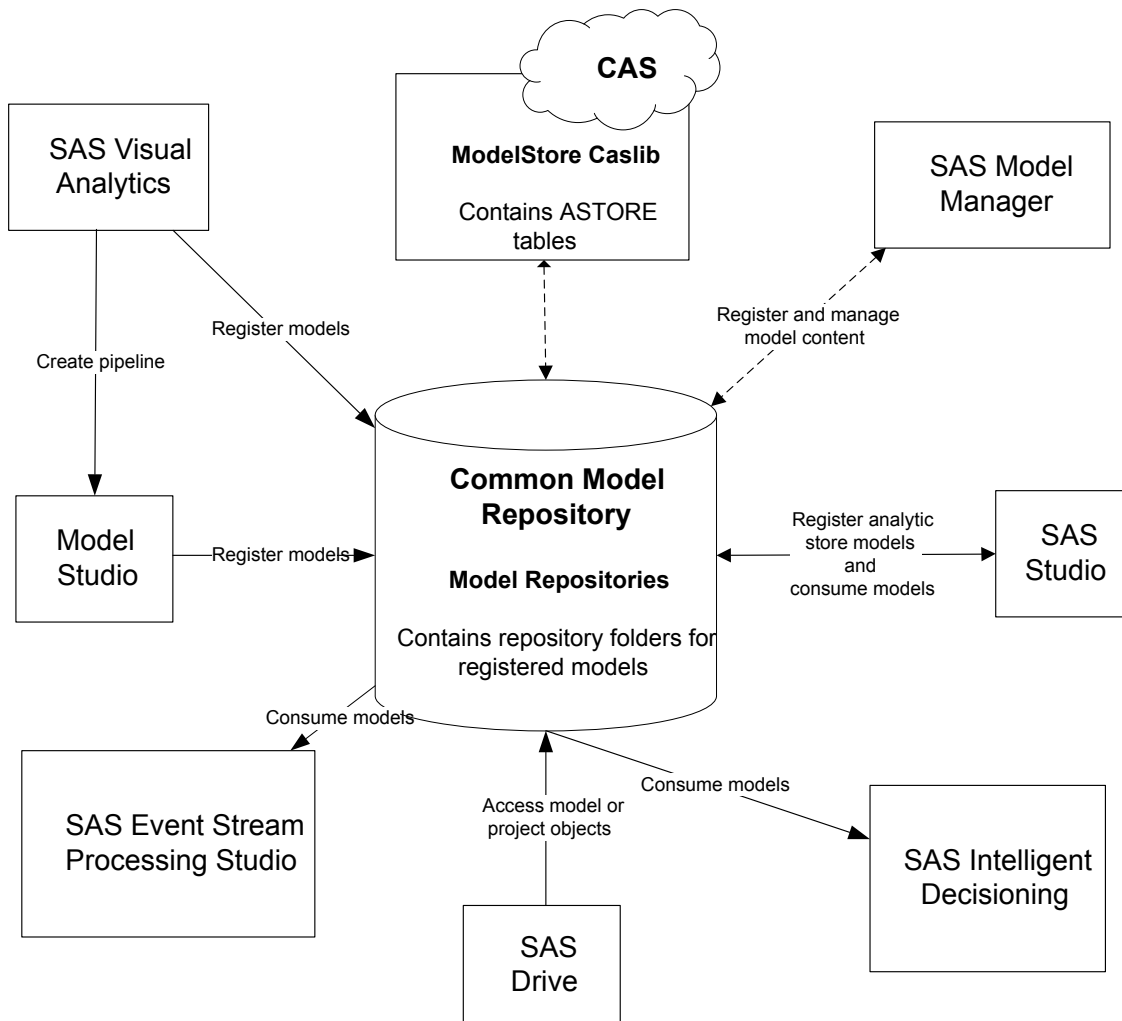
2

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 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.

See ["Data" in SAS Cloud Analytic Services: Fundamentals](#).

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.

See [“Managing Model Repositories” in SAS Model Manager: User’s Guide](#).

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.

See [SAS Visual Data Mining and Machine Learning: User’s Guide](#) and [SAS Visual Text Analytics: User’s Guide](#).

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: Documentation](#)

SAS Event Stream Processing Studio

A SAS web application that enables you to create, edit, upload, publish, and test event stream processing models using SAS Event Stream Processing Studio Modeler. SAS Event Stream Processing Studio Modeler displays a model as a data flow diagram, which enables you to see and control how windows relate and flow to one another. Projects can reference models that are stored in the common model repository. When a project is deployed, the model is retrieved from the common model repository and written to the ESP server. SAS Micro Analytic Service modules are used to accommodate the imported content that was created in SAS Model Manager. The module is uploaded and then referenced from the Calculate window’s input handler. See [SAS Event Stream Processing: Using SAS Event Stream Processing Studio](#)

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, open-source programming languages, and PMML. 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 and monitored for performance. Models within folders or projects can also be published to a configured publishing destination that can be defined for CAS, Hadoop, SAS Micro Analytic Service, and Teradata, as well as Amazon Web Services (AWS) and Private Docker containers.

See [SAS Model Manager: User’s Guide](#).

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.

See [“Register a Model” in SAS Studio: Task Reference Guide](#).

SAS Visual Analytics

A 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. SAS Visual Data Mining and Machine Learning cannot be licensed without SAS Visual Statistics. 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.

See [SAS Visual Analytics: Working with SAS Visual Data Mining and Machine Learning](#) and [SAS Visual Analytics: Working with SAS Visual Statistics](#).

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 [“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:

- [“Managing Projects” in SAS Visual Data Mining and Machine Learning: User’s Guide](#)
- [“Managing Projects” in SAS Visual Text Analytics: User’s Guide](#)

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 Only SAS Administrators and other authorized users can create, update, or delete repository folders. In addition, custom repositories by default cannot be accessed by authenticated users. A SAS Administrator must grant user’s access to custom repositories.

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 1 Initial Default and Standard Repositories

Repository Name	Description
Public	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 .
DMRepository	SAS Visual Data Mining and Machine Learning models registered from Model Studio to the common model repository are stored within the DMRepository. The DMRepository is made available with SAS Visual Data Mining and Machine Learning as part of the Model Studio installation process.
VARRepository	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 VARRepository. The VARRepository folder is created the first time a SAS Visual Analytics report object is registered as a model to the common model repository.
VTARRepository	SAS Visual Text Analytics models registered from Model Studio to the common model repository are stored within the VTARRepository. The VTARRepository folder is created the first time a SAS Visual Text Analytics model is registered from Model Studio to the common model repository.

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

Principal	Read	Update	Delete	Secure	Add	Remove	Read (convey)	Update (convey)	Delete (convey)	Secure (convey)	Add (convey)	Remove (convey)
Authenticated Users	✓ ♦	✗	✗	✗	✓ ♦	✗	✗	✗	✗	✗	✗	✗
SAS Administrators	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
SAS Demo User	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

The following figure is an example of the initial authorization for the default repository (Public) and standard repository (DMRepository, VARRepository, and VTARRepository) folders.

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

Principal	Read	Update	Delete	Secure	Add	Remove	Read (convey)	Update (convey)	Delete (convey)	Secure (convey)	Add (convey)	Remove (convey)
Authenticated Users	✓ ♦	✗	✗	✗	✓ ♦	✓ ♦	✓ ♦	✓ ♦	✓ ♦	✓ ♦	✓ ♦	✓ ♦
SAS Administrators	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SAS Demo User	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

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

Figure 3 Initial Authorization for a Custom Repository Folder

Principal	Read	Update	Delete	Secure	Add	Remove	Read (convey)	Update (convey)	Delete (convey)	Secure (convey)	Add (convey)	Remove (convey)
Authenticated Users	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
SAS Administrators	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
SAS Demo User	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

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:

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 for the new SAS Viya deployment.

Figure 4 Common Model Repository Rules (/modelRepository)

/modelRepository			
Object URI	Principal	Setting	Permissions
/modelRepository/	Authenticated Users	Grant	Read
/modelRepository/projects	Authenticated Users	Grant	Read, Create
/modelRepository	Authenticated Users	Grant	Read
/modelRepository/repositories	Authenticated Users	Grant	Read
/modelRepository/projects/	Authenticated Users	Grant	Read
/modelRepository/models/	Authenticated Users	Grant	Read
/modelRepository/models/transfer	Authenticated Users	Grant	Create
/modelRepository/repositories	sasapp	Grant	Create
/modelRepository/models/userProperties	Authenticated Users	Grant	Read
/modelRepository/projects/userProperties	Authenticated Users	Grant	Read
/modelRepository/jobs	Authenticated Users	Grant	Create
/modelRepository/models	Authenticated Users	Grant	Read, Create
/modelRepository/repositories/*	SASAdministrators	Grant	Update, Delete
/modelRepository/repositories	SASAdministrators	Grant	Create

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.

Predefined Rules for the Model Repositories Folder

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

Target:	/Model Repositories/** (folder)
Object URI:	/folders/folders/folder-ID/**
Permission s:	Add, Read
Target:	/Model Repositories (folder)
Container URI:	/folders/folders/folder-ID
Permission s:	Add, Read, Create

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 **VARepository** folder grant access to Authenticated Users as follows:

Target:	/Model Repositories/VARepository/** (folder)
objectURI:	/folders/folders/folder-ID/**
Permission s:	Add, Read, Create, Remove
Target:	/Model Repositories/VARepository (folder)
containerURI:	/folders/folders/folder-ID

Permission Add, Secure, Read, Update, Create, Remove, Delete
s:

See Also

- [“Managing Permissions” in SAS Model Manager: Administrator’s Guide](#)
- [“General Authorization: Concepts” in SAS Viya Administration: General Authorization](#)
- [“General Authorization: How To \(Rules Page\)” in SAS Viya Administration: General Authorization](#)
- [“General Authorization: How To \(Authorization Window\)” in SAS Viya Administration: General Authorization](#)

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.](#)
- 4 [Secure the Model Repositories folder and its immediate children.](#)
- 5 [Reduce the availability of model applications and features.](#)

Create Custom Groups

Create two new [custom groups](#) as follows:

Name	ID	Membership Description
Modelers	_modelers	Anyone who uses model functionality.
Model Repository Administrators	_modelRepoAdmins	Anyone who should be able to add, update, and delete repositories.

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 From the applications menu (☰), select **Manage Environment**.
- 2 In SAS Environment Manager, click  to view the **Licensed Products** page.
- 3 Enter **SAS Model Manager** in the **Product** filter field.
- 4 Enter the following licensed products names in the **Product** filter field.

Licensed Product	Standard Repository
SAS Data Mining and Machine Learning	DMRepository
SAS Visual Analytics Explorer	VARepository
Text Analytics UI Setinit	VTARespository

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
- VTARespository

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, SAS Model Repository CLI, 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 using the SAS Model Repository CLI or 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 Click the arrow to the right of the **SAS Content** folder to locate the **Model Repositories** folder. Right-click the folder and select **Edit authorization**.

Note: Only users with SECURE permission to the selected folder can edit its authorization.

Adjust access as follows:

- a Click  and select **Add identities**. The Add Identities window appears.

Note: If guest access is not configured, the Add Identities window appears when you click .

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

Note: Both groups inherit Read and Add object permissions.

- c For the Model Repository Administrators group, click  to set a direct grant for object permissions of Remove, as well as grant container (convey) permissions of Read, Update, Delete, Secure, Add, and Remove.
- d For the Modelers group, click  to set a direct grant for object permissions of Read and Add.
- e Click **Preview**.

After the **Model Repositories** folder has been secured, here are its authorizations.

Principal	Read	Update	Delete	Secure	Add	Remove	Read (convey)	Update (convey)	Delete (convey)	Secure (convey)	Add (convey)	Remove (convey)
 Authenticated Users	✓ ♦	✗	✗	✗	✓ ♦	✗	✗	✗	✗	✗	✗	✗
 Model Repository Administrators	✓	✗	✗	✗	✓	✓ ♦	✓ ♦	✓ ♦	✓ ♦	✓ ♦	✓ ♦	✓ ♦
 Modelers	✓ ♦	✗	✗	✗	✓ ♦	✗	✗	✗	✗	✗	✗	✗
 SAS Administrators	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
 SAS Demo User	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

f Click **Save**.

- 3 Right-click a repository (any immediate child of the **Model Repositories** folder) and select **Edit authorization**.

Note: Only users with SECURE permission to the selected folder can edit its authorization.

- a Click  and select **Add identities**. The Add Identities window appears.
- b Select the **Modelers** custom group as an identity. Click **OK**.
- c Click  in each cell to remove the current grants that are set for all object permissions for Authenticated Users. Change the **Direct setting** value to **None**.

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

- d For the Modelers group (or if you are securing a custom repository for a more specific group), click  in each cell to add direct object grants of Read, Add, and Remove and container (convey) grants of Read, Update, Delete, Secure, Add, and Remove.
- e Click **Preview**.

Here is an example of the authorizations for the **VARespository** folder.

VARespository													
☐ Show horizontal column headings 👤 ↺ ?													
Principal	Read	Update	Delete	Secure	Add	Remove	Read (convey)	Update (convey)	Delete (convey)	Secure (convey)	Add (convey)	Remove (convey)	
👤 Authenticated Users	🚫	🚫	🚫	🚫	🚫	🚫	🚫	🚫	🚫	🚫	🚫	🚫	
👤 Model Repository Administrators	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	
👤 Modelers	✅ ♦	🚫	🚫	🚫	✅ ♦	✅ ♦	✅ ♦	✅ ♦	✅ ♦	✅ ♦	✅ ♦	✅ ♦	
👤 SAS Administrators	✅	✅	✅	✅	✅	✅	🚫	🚫	🚫	🚫	🚫	🚫	
👤 SAS Demo User	✅	✅	✅	✅	✅	✅	🚫	🚫	🚫	🚫	🚫	🚫	

f Click **Save**.

Repeat this step for each repository (each immediate child of the **Model Repositories** folder). Here are key points:

- The DMRepository model repository is created during the installation process. The VARespository and VTARespository 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 15.

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, click  to view the **Rules** page.
- 2 For each rule in the following table, replace the original principal (principalType=Authenticated Users) with the custom modelers group (for example, principalType=group, name=Modelers, and group ID=_modelers) that you created earlier. For more information, see [“Create Custom Groups”](#) on page 10.

Application or Feature	Target URI (or URIs)
Ability to use model objects in SAS Visual Analytics	/SASVisualAnalytics_capabilities/buildAnalyticalModel
Ability to access SAS Model Manager ¹	/SASModelManager/ and /SASModelManager

Application or Feature	Target URI (or URIs)
Ability to access Model Studio ²	/ModelStudio/**

- 1 These are two alternate forms of the same URI. In the current release, because SAS Model Manager uses both forms, both rules are necessary.
- 2 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 [Add model repository rules for a custom repository.](#)
- 4 [Create a custom repository.](#)
- 5 [Secure the Model Repositories folder and its immediate children.](#)
- 6 [Reduce the availability of model applications and features.](#)

Create Custom Groups

Create two new [custom groups](#) as follows:

Name	ID	Description
Modelers	_modelers	Anyone who uses model functionality.
Model Repository Administrators	_modelRepoAdmins	Anyone who should be able to add,update, and delete repositories.

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 From the applications menu () select **Manage Environment**.
- 2 In SAS Environment Manager, click  to view the **Licensed Products** page.
- 3 Enter **SAS Model Manager** in the **Product** filter field.

Add Model Repository Rules for a Custom Repository

In order to create, delete, and update custom repositories as well as perform certain actions on objects (such as deleting model variables) within a custom repository, additional rules must be added.

- 1 In SAS Environment Manager, click  to view the **Rules** page.
- 2 Click .
- 3 By default, only SAS Administrators can create, update, or delete repository folders. You can add additional Model Repository rules for repository object access to enable other users such as the Model Repository Administrators to perform these actions.

- Here is the rule to add for granting users the ability to read and create a repository:

Field Name	Value
Object URI	/modelRepository/repositories
Principal type	Group
Principal	Model Repository Administrators
Rule type	Grant
Permissions	Read, Create
Reason	Provides the ability to create a model repository.

- Here is the rule to add for granting users the ability read, delete, and update repositories:

Field Name	Value
Object URI	/modelRepository/repositories/*
Principal type	Group
Principal	Model Repository Administrators
Rule type	Grant
Permissions	Read, Delete, Update
Reason	Provides the ability to delete, read, and update repositories.

- 4 In order for non-administrator users to be able to create, delete, and update model variables, you must add additional Model Repository rule for model object access.

Field Name	Value
Object URI	/modelRepository/models/**
Principal type	Group
Principal	Modelers
Rule type	Grant
Permissions	Create, Delete, Read, Update
Reason	Provides ability to create, update, and delete model variables.

For more information, see [“Add a Rule” in SAS Viya Administration: General Authorization](#).

Create a Custom Repository

A SAS Administrator or another authorized user 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”](#).

Note: You can authorize other users or groups the ability to create and update custom repositories using rules. See [“Add Model Repository Rules for a Custom Repository” on page 16](#).

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 Click the arrow to the right of the **SAS Content** folder to locate the **Model Repositories** folder. Right-click the folder and select **Edit authorization**.

Note: Only users with SECURE permission to the selected folder can edit its authorization.

Adjust access as follows:

- a Click  and select **Add identities**. The Add Identities window appears.
- b Select the **Modelers** and **Model Repository Administrators** custom groups as identities. Click **OK**.

Note: Both groups inherit Read and Add object permissions.

- c For the Model Repository Administrators group, click  to set a direct grant for object permissions of Remove, as well as grant container (convey) permissions of Read, Update, Delete, Secure, Add, and Remove.
- d For the Modelers group, click  to set a direct grant for object permissions of Read and Add.

Note: If you have updated your installation from a previous release, the Authenticated Users principal group might have container (convey) permissions for the **Model Repositories** folder. Remove container access for Authenticated Users by changing the direct setting to **(none)** for each convey permission. Authenticated users should have only Read and Add object permissions.

- e Click **Preview**.

After the **Model Repositories** folder has been secured, here are its authorizations:

- e Click **Save**.

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 15](#).

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, click  to view the **Rules** page.
- 2 In each rule in the following table, replace the original principal (principalType=Authenticated Users) with the custom modelers group (for example, principalType=group, name=Modelers, and group ID=_modelers) that you created earlier. For more information, see [“Create Custom Groups” on page 10](#).

Application or Feature	Target URI (or URIs)
Ability to access SAS Model Manager ¹	/SASModelManager/ and /SASModelManager

- ¹ These are two alternate forms of the same URI. In the current release, because SAS Model Manager uses both forms, both rules are necessary.

How to Create Repositories

You can create a new repository using several different methods. 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. You can also use the SAS Model Repository CLI or the %MM_CREATE_REPOSITORY macro. If you do not have a license to SAS Model Manager, you can use the SAS Model Repository CLI or register a model from Model Studio or SAS Visual Analytics to create the standard repositories. You can also use the SAS Model Repository CLI. For information about the SAS Model Repository CLI, see [SAS Model Manager: Command-Line Interfaces](#) and for information about repository authorization, see [“Generated Rules for Each Repository”](#).

Note: The **DMRepository** is made available with SAS Visual Data Mining and Machine Learning as part of the Model Studio installation process.

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** ⇒ **Repositories**.
- 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 6](#). 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.

Note: The **DMRepository** is made available with SAS Visual Data Mining and Machine Learning as part of the Model Studio installation process.

VARepository

- 1 On the **Content** page in SAS Environment Manager (**Manage Environment**), select **SAS Content** and verify that a **VARepository** folder does not already exist directly beneath the **Model Repositories** root folder.
- 2 Click ≡ and select **Explore and Visualize**.
- 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.

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 Click **Assign data** and use the form to 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 model canvas and select **Register model**.
- 6 On the **Content** page in SAS Environment Manager (**Manage Environment**), select **SAS Content** and verify that a **VARepository** folder was created under the Model Repositories root folder.

For more information, see “[Register Models and Access the Model Repository](#)” in [SAS Visual Analytics: Working with SAS Visual Statistics](#).

VTARepository

- 1 On the **Content** page in SAS Environment Manager (**Manage Environment**), select **SAS Content** and verify that a **VTARepository** folder does not already exist directly under the **Model Repositories** root folder.
- 2 Click  and 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 model node and select **Register model**.
- 5 On the **Content** page in SAS Environment Manager (**Manage Environment**), select **SAS Content**, and verify that a **VTARepository** folder was created under the **Model Repositories** folder.

For more information, see “[Registering Models](#)” in [SAS Visual Text Analytics: User's Guide](#).

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, Model Studio, or SAS Studio. The ModelStore caslib must exist on each CAS server that is in use. If your model references a train table in a library that exists on a CAS server, but the CAS server 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.

Note: When you transfer analytic store models from one system to another, you must manually copy the content that is stored in your file system from the source system to the target system. For more information, see [“Promoting Analytic Store Models” on page 28](#).

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](#).

Note: The CAS data directory path is configured during installation. The default value for the CASDATADIR environment variable is `/opt/sas/viya/config/data/cas/default/`.

Table 2 Predefined Caslibs

Caslib	Default Assignment
ModelPerformanceData	<code>/opt/sas/viya/config/data/cas/default/modelMonitorLibrary/</code> Stores application-generated performance results data that is used for reporting. The directory assignment is on the CAS controller node.
ModelStore	<code>/opt/sas/viya/config/data/cas/default/modelStore/</code> The file system directory is associated with the ModelStore CAS library. The directory assignment is on the CAS controller node. 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 <code>/opt/sas/viya/config/data/modelsvr/astore</code> 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 24](#).

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” in SAS® Viya Administration: CAS 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`.

```
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.

```
--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)
end
```

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”](#).

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` 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 multi-host machine 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 24](#).

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.

Here are examples of the commands that you can use:

```
sudo chmod 777 /opt/sas/viya/config/data/modelsvr/astore/
```

Multi-tenant deployment directory permissions:

```
sudo chmod 755 /opt/sas/tenant/config/data/
```

Note: The permissions for the `/opt/sas/tenant/config/data/modelsvr/astore/` directory for each tenant are set to 777 by default during the deployment process.

- 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](#).

Configuring Analytic Store Directories in a Multi-Tenant Deployment

In a multi-tenant deployment, verify that the analytic store (ASTORE) directories for each tenant exist after the tenant is onboarded. The directory paths must reflect the tenant name. This includes the `/opt/sas/tenant/config/data/modelsvr/astore` directory on the Compute server and the `/models/astores/tenant` directory on each server that hosts either SAS Micro Analytic Service or SAS Event Stream Processing. Then you must perform the steps for [mapping analytic store directories](#) and [setting permissions on page 27](#) on the analytic store directories for each onboarded tenant. 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
```

Promoting Analytic Store Models

When you promote analytic store models from one system to another using SAS Environment Manager or the `sas-admin` CLI transfer plug-in, you must manually copy the ModelStore caslib content that is stored in your file system from the source system to the target system.

Note: When you export an analytic store model using the SAS Model Manager 15.3 web application, the analytic store file is included in the ZIP file. If you re-import the analytic store model to the same system or another system, the analytic store file is copied to the ModelStore caslib.

The analytic store files are located in the `/models/astores/viya` directory. In a multi-tenant deployment, the analytic store files are located in the `/models/astores/tenant` directory.

For more information, see [“Promoting Content” in SAS Model Manager: Administrator’s Guide](#).

Configuring Support for Python Code Files

Here are the requirements for configuring support for Python code files:

- To support models that contain custom Python code files, you must enable PyMAS package support.

Note: The file system directory path for Python code files is `/models/resources/viya`. This directory and its permissions are set during the installation and configuration process for the SAS Micro Analytic Service.

For more information, see [“Enabling PyMAS Package Support” in SAS Micro Analytic Service: Programming and Administration Guide](#).

- Users who are developing and testing Python code files must be added to the CASHostAccountRequired custom group. For more information, see [“The CASHostAccountRequired Custom Group” in SAS Viya Administration: Identity Management](#) and [“Add or Remove Custom Group Members” in SAS Viya Administration: Identity Management](#).
- The resources directory must be granted Read and Write permissions.
For information, see [“Setting Permissions” on page 30](#).

Accessing Python Resource Files

Mapping Resource Directories

In order to publish Python models or decisions that use Python models to the SAS Micro Analytic Service publishing destination, the model’s resource files must be accessible from the `/models/resources/viya` directory path.

To make the Python model’s resource files accessible, you must map the `/opt/sas/viya/config/data/modelsvr/resources` directory on the Compute server to the `/models/resources/viya`

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 multi-host machine 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/resources` directory.

Setting Permissions

Users who need to work with Python models must have Read and Write permissions to both the `/opt/sas/viya/config/data/modelsvr/resources` and `/models/resources/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 Python 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.

Here are examples of the commands that you can use:

```
sudo chmod 777 /opt/sas/viya/config/data/modelsvr/resources/
```

Multi-tenant deployment directory permissions:

```
sudo chmod 755 /opt/sas/tenant/config/data/
```

Note: The permissions for the `/opt/sas/tenant/config/data/modelsvr/resources/` directory for each tenant are set to 777 by default during the deployment process.

- 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](#).

Configuring Resource Directories in a Multi-Tenant Deployment

In a multi-tenant deployment, verify that the resource directories for each tenant exist after the tenant is onboarded. The directory paths must reflect the tenant name. This includes the `/opt/sas/tenant/config/data/modelsvr/resources` directory on the Compute server and the `/models/resources/tenant` directory on each server that hosts either SAS Micro Analytic Service or SAS Event Stream Processing. Then you must perform the steps for [mapping resources directories](#) and [setting permissions](#) on the resources directories for each onboarded tenant.

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

```
/models/resources/companyA
```

```
/opt/sas/companyA/config/data/modelsvr/resources
```

See Also

[“Python Models” in SAS Model Manager: User's Guide](#)

