Overview of SAS Model Manager Administration

This guide provides post-installation configuration tasks for SAS Model Manager 15.2 on SAS Viya, and explains both how to prepare SAS Model Manager for use and how to manage information that is associated with SAS Model Manager. The administrator uses SAS Environment Manager to define data sources, configure publish destinations, and manage user access and permissions.

Here are the tasks that are included:

- Manage permissions
- Manage content
- Promote content
- Configure data libraries
- Configure publishing destinations
- Configure access to analytic store model files
- Configure support for Python code files
- Restart the Model Repository service

For information about deploying SAS Model Manager 15.2 on the SAS Viya platform, see the Deployment and Administration section of the SAS Viya documentation page.

Managing Permissions

How to Manage Permissions

SAS Environment Manager is used to manage identities and authorization for SAS Viya. Information is available in the SAS Viya administration documentation:

- “Two Authorization Systems” in SAS Viya Administration: Orientation to Authorization
- “Identity Management: Overview” in SAS Viya Administration: Identity Management
The default permissions for SAS Model Manager are described in “Default Permissions” on page 2. To modify the default permissions, you can do the following:

- Modify permissions for specific folders or objects by using the Edit Authorization window. For more information, see “General Authorization: How to (Authorization Window)” in SAS Viya Administration: General Authorization.

- Modify the existing groups or create new ones. For more information, see the following topics:
  - “Manage Custom Groups” in SAS Viya Administration: Identity Management
  - Granting Access to Test Results

- Modify the existing rules or create new rules. For more information, see “General Authorization: How to (Rules Page)” in SAS Viya Administration: General Authorization.

- Modify file system directory permissions for caslibs. For more information, see “File System Directory Permissions” in SAS Viya Administration: Models.

Note: If the Rules page or the Authorization window displays a warning icon next to a principal’s name, that principal does not exist in the identities service. For more information, see “Unrecognized Principals” in SAS Viya Administration: General Authorization.

**Default Permissions**

Note: 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, authenticated users cannot initially access new custom repositories. A SAS Administrator must grant access for a user or group to a new custom repository. Authorization for existing repositories is not modified during an upgrade. For more information, see “Access to Models” in SAS Viya Administration: Models.

By default, all authenticated users have permission to do the following in the default repository and standard repositories:

- Read or view a list of all models and projects.
- Create a standard repository folder when registering models from Model Studio or SAS Visual Analytics.
- Create a model or project.
- Copy a model from another project or a folder.
- Move a model from a folder to another folder or project version.
- Update and delete any model or project.
- Publish any model, including a project champion and challenger models.
- Create a test definition for any model that they have access to read.
- Create a performance definition, run performance, and view performance results and history.

By default, only the user that created a test definition can do the following:

- View, update, or delete the test definition.
- Run the test and view the test results.

You can grant access to test definitions and test results to users other than the user that created the definition either by adding the users to the SASScoreUsers group or by granting the users access to specific test results. For more information, see “Granting Access to Test Results” on page 3.

Note: Specific permissions are required for the following tasks:

- To start a workflow, you must be in the Application Administrators group and have permissions to access the workflow definition. For more information, see SAS Workflow Manager: Administrator’s Guide.
To move a model, you must have the appropriate permissions for the source folder, object, and target folder. For more information, see “Move an Object or Folder” in SAS Viya Administration: Content Management.

Granting Access to Test Results

By default, only the user who creates a test can view, update, or delete the test definition or run the test. Only a user who runs a test can view the test results. Other users do not have permission to access the test definition or test results unless the user is a member of the SASScoreUsers group.

SAS Model Manager configures the SASScoreUsers group automatically. Members of this group have full access to test definitions and results. These permissions enable access through the user interface, the Score Definitions service, and the Score Execution service. For instructions on adding users to a group, see “Manage Custom Groups” in SAS Viya Administration: Identity Management.

Managing Content

Information that you or other users save is stored and organized in folders. A folder is a virtual container rather than a representation of a physical file system. A folder contains members that are URIs for other folders, SAS resources, or resources outside of SAS. The Content page in SAS Environment Manager contains objects (such as SAS content and reports) that you save and that are organized into folders.

The Model Repositories folder is a common model repository for SAS applications. The Model Repositories folder can contain one or more repository folders. A repository folder can contain folders, models, and projects. A folder within a repository folder can contain models or projects. A project contains project versions, and a project version can contain one or more models. When a user creates a model or project, both a folder and an object are created. The folder and the object have the same name, and the object appears within the folder.

CAUTION! Do not rename folders or objects. The name of the Model Repositories folder, as well as repository folders, project folders, project version folders, model folders, and objects within the Model Repositories folder should not be modified using SAS Environment Manager. However, user-defined folders within a repository folder can be renamed.
Promoting Content

About Promoting Content

Promotion is the process of capturing content and moving it to a different location. For SAS Model Manager 15.2 on SAS Viya, promotion is performed using the transfer plug-in to the admin command-line interface (CLI) or by using SAS Environment Manager. You can promote content from SAS Model Manager 15.1 on SAS Viya 3.3 or SAS Model Manager 15.2 on SAS Viya 3.4 to SAS Model Manager 15.2 on SAS Viya 3.4.

Workflow content cannot be promoted or migrated. Promoting content from SAS Model Manager 14.2 on SAS 9.4 to SAS Model Manager 15.2 on SAS Viya is not currently supported using the SAS Viya 3.4 transfer plug-in.

Important: If you are promoting content from SAS Model Manager 15.1 on SAS Viya 3.3 to SAS Model Manager 15.2 on SAS Viya 3.4, you must use the sas-admin command-line interface.

For more information, see the following documentation:

- “Promotion within SAS Viya: Instructions” in SAS Viya Administration: Promotion (Import and Export)
- “Import SAS Viya Resources” in SAS Viya Administration: Promotion (Import and Export)
- “Promotion: How to Import (Command-Line Interface)” in SAS Viya Administration: Promotion (Import and Export)
Folder and Object Types

The following folder and object types can be exported from the Content page of SAS Environment Manager. If you are using the sas-admin command-line interface, you can copy the URIs from SAS Environment Manager.

Here are a few points to be aware of:

- The Model Repositories folder, a repository folder, a folder or subfolder, and a project folder can be transferred. The transfer package contains all of the associated folders and objects within the exported folder.
- When a project object is exported, it does not contain the associated model folders and model objects in the transfer package. The model folder or model object URIs must be included in the transfer package.
- The project version folder does not contain the project object or model objects. Only the folders are transferred. Therefore, it is not recommended to promote content at the project version folder level.

Table 1 Examples of Folder and Object Types

<table>
<thead>
<tr>
<th>Folder or Object Type</th>
<th>Location Example</th>
<th>URI Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Repositories folder</td>
<td>/Model Repositories</td>
<td>/folders/folders/02c3ede2-0a41-4d56-997b-703aee31b329</td>
</tr>
<tr>
<td>repository folder</td>
<td>/Model Repositories/QS_Repository</td>
<td>/folders/folders/3b709148-29c0-4be5-a1f1-4109b9b040c6</td>
</tr>
<tr>
<td>folder</td>
<td>/Model Repositories/QS_Repository/sasdemo</td>
<td>/folders/folders/870af9c9-9074-4ee0-84a6-bc3cd4954cfcf</td>
</tr>
<tr>
<td>project folder</td>
<td>/Model Repositories/QS_Repository/sasdemo/QS_HMEQ</td>
<td>/folders/folders/de9c31c8-881f-40c0-86ef-029256c5fa5</td>
</tr>
<tr>
<td>project version folder</td>
<td>/Model Repositories/QS_Repository/sasdemo/QS_HMEQ/Version 1</td>
<td>/folders/folders/aa948017-7fe2-405f-abfc-1c7245f29e9f</td>
</tr>
<tr>
<td>project object</td>
<td>/Model Repositories/QS_Repository/sasdemo/QS_HMEQ/QS_HMEQ</td>
<td>/modelRepository/projects/a0792aac-1f20-4f70-8a0c-72baeb29523c</td>
</tr>
<tr>
<td>model folder</td>
<td>/Model Repositories/QS_Repository/sasdemo/QS_HMEQ/Version 1/QS_Tree1</td>
<td>/folders/folders/3a7b1268-f9a4-4e7b-86f9-64df9acd4f99</td>
</tr>
<tr>
<td>model object</td>
<td>/Model Repositories/QS_Repository/sasdemo/QS_HMEQ/Version 1/QS_Tree1/QS_Tree1</td>
<td>/modelRepository/models/79b160f3-5fc4-4b66-9a82-baa98b2b3060</td>
</tr>
</tbody>
</table>

Important Considerations

Here are a few points to be aware of:

- When transferring content using the project object, model folder, or model object, you should import the projects first, followed by the models.
When you are transferring content using the project object or project version folder, not all of the content is included in the transfer package. Therefore, it is not recommended to transfer content at these levels. You should transfer content from the project folder level instead.

When project folders or project objects are transferred, the scoring tests, test results, performance definition, and performance results are not included. Scoring tests can be transferred using the sas-admin command-line interface with the score definition URI and the score execution URI. However, it is recommended that you re-create and rerun the test in the target environment rather than transfer the test information.

When you are transferring analytic store models, you must manually copy the content that is stored on your file system from the source system to the target system. For more information, see “Configuring Access to Analytic Store Model Files” in SAS Viya Administration: Models.

When objects are transferred, the following properties change based on who performed the transfer and when it was done:

- Created by
- Modified by
- Date created
- Date modified

Transferring of data from the Model Publish service and Model Management service is not supported.

Transfer Request Body Examples

Folders and objects can be transferred at different levels from the Model Repository service using SAS Environment Manager or the transfer plug-in to the sas-admin command-line interface. Here are some examples of the request body that would appear within the JSON files that are used to transfer folder and object content using the sas-admin command-line interface.

Model Repositories Folder

If you specify the folder URI for the /Model Repositories directory, all of the repository folders, subfolders, authorization rules, projects, models, and model files are transferred as one package. Sample request body:

```json
{
   "name": "ModelRepositoriesFolderID",
   "items": [
      "/folders/folders/0fec5575-2ee2-4b18-ac3b-5afdd32b4412"
   ]
}
```

Repository Folder

If you specify the folder URL for a repository folder (for example, /Model Repositories/myRepository), all subfolders, authorization rules, projects, models, and model files are transferred as one package. Sample request body:

```json
{
   "name": "RepositoryFolderID",
   "items": [
      "/folders/folders/b5d59f4c-2346-4de8-bb0d-e3714cdf5594"
   ]
}
```
Folder
If you specify a folder URI for a folder within a repository folder (for example, `/Model Repositories/myRepository/myFolder`), all subfolders, authorization rules, projects, models, and model files are transferred as one package. Sample request body:

```json
{
  "name": "FolderID",
  "items": [
    "/folders/folders/9c33b3dd-746e-4ebb-947a-c596f36d96d6"
  ]
}
```

Project Folder
If you specify the URIs for one or more projects, the specified projects and the associated models are transferred.

```json
{
  "name": "Projects",
  "items": [
    "/folders/folders/de9c31c8-881f-40c0-86ef-029256c5f0a5",
    "/folders/folders/18f2c85c-68e3-418a-904e-a405aff8eb50"
  ]
}
```

Project Object
If you specify the URIs for one or more projects, the specified projects are transferred.

```json
{
  "name": "Projects",
  "items": [
    "/modelRepository/projects/4f29e89c-bc93-42f5-8491-f338025d75e3",
    "/modelRepository/projects/15103b67-586c-4a3a-8ffdb840c0921734"
  ]
}
```

Model Object
If you specify the URIs for one or more models, the specified models and their content are transferred.

```json
{
  "name": "Models",
  "items": [
    "/modelRepository/models/c8f5694d-9717-45de-98cc-14d08edf7e10",
    "/modelRepository/models/85c66c5e-58ef-4ae6-9df3-3d709da980f",
    "/modelRepository/models/d62ea249-5bea-4fc1-911d-966a387f4758"
  ]
}
```
sas-admin Command Line Interface Example

```bash
./sas-admin --profile mySource profile init
./sas-admin --profile myTarget profile init

./sas-admin --profile mySource auth login
./sas-admin --profile myTarget auth login

/opt/sas/viya/home/bin/sas-admin --profile mySource transfer export
   --request @request_myrepository.json

/* Replace the value of the id below with the id taken from */
/* the command output above in the console */
/opt/sas/viya/home/bin/sas-admin --profile mySource transfer download
   --id 9b613274-ccf2-48b6-9567-85f46adb1430 --file myPackage_repository.json

/opt/sas/viya/home/bin/sas-admin --profile myTarget transfer upload
   --file myPackage_repository.json

/* Replace the value of the id below with the id taken from */
/* the command output above in the console. */

/opt/sas/viya/home/bin/sas-admin --profile myTarget transfer import
   --request "{"packageUri":":"/transfer/packages/6b62ddfc-0a00-4b30-987e-1bb7c428e9a1"}"
```

**Configuring Data Libraries**

**About Configuring Data Libraries**

During the deployment of SAS Model Manager, 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. For more information, see “Configuring Model Data Libraries” in SAS Viya Administration: Models.

**File System Directory Permissions**

When defining a caslib where the source type is a file system directory path, the appropriate permissions must be granted. 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 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’ ability to share analytic stores depends on their primary group permissions. An analytic store is created with group ownership by each user’s primary group. For more information, see “File System Directory Permissions” in SAS Viya Administration: Models.
Configuring Publishing Destinations

You can publish content to destinations on SAS Cloud Analytic Services (CAS), Apache Hadoop, SAS Micro Analytic Service, and Teradata. By default, a SAS Micro Analytic Service destination named maslocal is defined for you. You must configure all other publishing destinations. Use SAS Environment Manager to manage publishing destinations. For more information, see SAS Viya Administration: Publishing Destinations.

You can use the SAS Model Manager tutorial to verify that your publishing destinations have been configured properly. For more information, see SAS Model Manager: Quick Start Tutorial.

Configuring Access to Analytic Store Model Files

In order to publish analytic store models to a SAS Micro Analytic Service destination, you must configure access to the location where the analytic store (ASTORE) files are located. Also, users who work with analytic store models must have Read and Write access to analytic store directories. For more information, see “Configuring Access to Analytic Store Model Files” in SAS Viya Administration: Models.

Configuring Support for Python Code Files

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

Restarting the Model Repository Service

If the default repository (for example, Public or Repository 1) has been renamed or deleted, the Model Repository service must be restarted by a system administrator in order to re-create the default repository. The default repository can be deleted only by a SAS Administrator.

Here is the command to restart the Model Repository service:

```
sudo service sas-viya-modelrepository-default restart
```

For more information, see “General Servers and Services: Operate (Linux)” in SAS Viya Administration: General Servers and Services.