SAS® Decision Manager 3.2: User's Guide

SAS® Documentation
November 26, 2018

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Decision Manager 3.2: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Decision Manager 3.2: User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

3.2-P2:edmug

Contents

What's New in SAS Decision Manager 3.2

Overview

SAS Decision Manager 3.2 runs on the sixth maintenance release of SAS 9.4. Major features for this release include the ability to set permissions on top-level business rules folders and support for SAS analytic store models. New features and enhancements in this release enable you to perform the following tasks:

- define an administrator role for folders and set permissions on top-level business rules folders
- control the ability to import and export business rules content
- set Read-Only privileges for vocabularies and lookup tables
- set default application servers for testing rule flows and decisions
- delete published rule flows
- generate SAS DATA step (DS1) code for rule flows
- use Oracle Database 12c for the SAS Decision Manager Common Data Server
- duplicate decisions
- import SAS analytic store models
- publish, score, monitor performance, or run reports for SAS analytic store models
- publish models to Hadoop and Teradata using single sign-on authentication by Kerberos
- run scoring tests in a High-Performance Analytics production environment
- score SAS Factory Miner models within a SAS Decision Manager portfolio by using macros
- publish SAS Factory Miner models from within a SAS Decision Manager portfolio to a database or to Hadoop by using macros

SAS Decision Manager 3.2M1 on SAS 9.4M6 enables you to do the following:

- compute the feature contribution indices for interval and nominal predictors by using the feature contribution index (FCI) macros
- convert and export a model's DS2 score code by using the %MM_GetModelDS2Code macro
- configure support for executable attachments

Generate SAS DATA Step Code for Rule Flows

The brm.runtime.codetype property controls whether SAS Decision Manager generates DS2 code or SAS DATA step (DS1) code for rule flows. See "Business Rules Manager Web Advanced Properties" in *SAS Decision Manager: Administrator's Guide* for more information.

Define a Business Rules Folder Administrator

The brm.folder.config.enabled property and the Folder Administration capability in SAS Management Console enable you to define a role for a business rules folder administrator. Users assigned to this role control which groups have access to business rules folders and the location of tests and test data for that folder. See "Enable Business Rules Folder Administration" in *SAS Decision Manager: Administrator's Guide* for more information.

Control Permission for Importing and Exporting Business Rules Content

SAS Decision Manager provides several macros for importing and exporting vocabularies, terms, lookup tables, rule sets, and rule flows to and from the Decision Manager database. You can limit the ability to run these macros by setting the correct properties and configuring identities in SAS Management Console. See "Configure the Ability to Import and Export Content" in *SAS Decision Manager: Administrator's Guide* for more information.

Import SAS Analytic Store Models

You can import SAS analytic store models that are created using SAS Factory Miner or by using the HPFOREST and HPSVM procedures in Base SAS. You can use a SAS analytic store model (SASAST) file to import an analytic store model into SAS Decision Manager. However, SAS analytic store models that are created by using SAS Enterprise Miner and that are in SAS package file (SPK) format are not supported in this release. For more information, see "Import a Model from a SAS Analytic Store File" on page 152.

Publish, Score, Monitor Performance, or Run Reports for SAS Analytic Store Models

You can publish, score, monitor performance, or run reports for SAS analytic store models that are located within a project. SAS analytic store models can be published only to Hadoop and Teradata. Retraining of SAS analytic store models is not supported. For more information, see the following:

- publish models to a database or Hadoop
- score models
- monitor performance
- run model comparison reports

Publish Models to Hadoop and Teradata Using Single Sign-On Authentication by Kerberos

Users who are configured to use single sign-on web authentication by Kerberos can publish models to Hadoop and Teradata. For more information, see "Configure Users Authenticated by Kerberos for Publishing Models" in *SAS Decision Manager: Administrator's Guide*.

Run Scoring Tests in a High-Performance Analytics Production Environment

When you are creating a scoring test for a model, you can choose to process the scoring test data in a High-Performance Analytics production environment. For more information, see "Create a Scoring Test" on page 196.

Score SAS Factory Miner Models within a SAS Decision Manager Portfolio

SAS Model Manager portfolio macros enable you to score SAS Factory Miner models that are within a SAS Decision Manager portfolio. For more information, see "%MM_ScorePortfolioModels Macro" in *SAS Model Manager: Macro Reference*.

Publish SAS Factory Miner Models from within a SAS Decision Manager Portfolio to a Database or Hadoop

SAS Model Manager portfolio macros enable you to publish SAS Factory Miner models that are within a SAS Decision Manager portfolio to a database or Hadoop. For more information, see "%MM_PublishPortfolioModelsDB" in *SAS Model Manager: Macro Reference* and "%MM_PublishPortfolioModelsHadoop" in *SAS Model Manager: Macro Reference*.

Feature Contribution Index Macros

Feature contribution index (FCI) macros enable you to compute the feature contribution indices for interval and nominal predictors, and create an ad hoc report. For more information, see "Feature Contribution Index Macros" in *SAS Model Manager: Macro Reference*.

%MM_GetModelDS2Code Macro

The %MM_GetModelDS2Code macro enables you to convert and export a model's DS2 score code from the SAS Model Manager model repository. This macro simplifies the integration of analytical models with SAS applications that support SAS DATA step, DS2, and analytic store score code execution.

Using the %MM_GetModelDS2Code macro provides these benefits:

- Provides an out-of-the-box macro to convert and export a model's DS2 score code and its attributes for use by external applications such as SAS Event Stream Processing Studio.
- Reduces errors that could result from manual steps that are required to obtain model DS2 score code and translate it to a format that is compatible with SAS Event Stream Processing.

For more information, see "Macro for Converting and Exporting Model DS2 Code" in *SAS Model Manager: Macro Reference*.

Configure Support for Executable Attachments

By default, users cannot attach executable files to objects within SAS Decision Manager. However, in SAS 9.4M6, you can configure SAS Decision Manager to allow executable attachments for models, projects, and portfolios. For more information, see "Configure Support for Executable Attachments" in *SAS Decision Manager: Administrator's Guide*.

Part 1

Introduction to SAS Decision Manager

Chapter 1

Introduction to SAS Decision Manager

Enterprise Decision Management Systems

Enterprise decision management systems can transform the way businesses make decisions. They enable businesses to use the information they already have to make better decisions—decisions that are based on predictive analytics rather than on past history. Decision management systems automate the process of making decisions, particularly day-to-day operational decisions. They improve the speed, efficiency, and accuracy of routine business processes, in part by reducing the need for human intervention. By automating decisions, organizations in every industry can improve interactions with customers, partners, suppliers, and employees. In addition, organizations that are highly regulated, such as financial services, health care, and insurance, can more easily achieve compliance as a result of repeatable, traceable decisions.

SAS Decision Manager helps organizations manage data, business rules, analytical models, and optimization techniques. Rule management, model management, and data management are integrated into a consistent interface for easier accessibility.

Data Management

SAS Decision Manager enables you to manage your list of data tables from within the application. You can create new Base SAS libraries, add and remove tables, view table data and metadata, create and delete table summaries, and associate attachments and comments with tables. The application uses data tables when it needs to access data, such as for rule discovery, rule flow testing, and testing, scoring, training, and performance monitoring of models.

Business Rule Management

About Business Rules

Business rules capture the logic of business decisions and are one of the core components of decision management systems. Business rules make the decision-making process transparent and adaptable, allowing organizations to respond quickly to new information about customers and markets. They allow organizations to identify and deal with fraud, avoid unnecessary risk, and find opportunities hidden in customer data.

You can use SAS Decision Manager to create a database of business rules, connect those rules together into rules flows, and publish the rule flows for use by other applications. SAS Decision Manager provides the following capabilities:

vocabulary management
: A business vocabulary defines entities and terms. Terms are the building blocks that you use to construct business rules. SAS Decision Manager enables you to easily create and edit entities and terms. For individual terms, you can create a list of allowable values, which makes creating rules even easier.

business rule authoring
: A business rule specifies conditions to be evaluated and action to be taken if those conditions are satisfied. For example, you can create a rule that determines whether a customer has a mortgage. That same rule can then add the outstanding balance of the mortgage to a running total of the customer's debt. With SAS Decision Manager, you define the conditions and actions for each rule. You can use the Equation Editor to create the expressions for the rule.

  The rule authoring features of SAS Decision Manager make creating rules easier and more accurate. For example, the list of allowable values for a term help avoid incorrect rules. The lists of allowable values can be updated as needed, and the lists do not prevent you from providing new values manually.

rule set management
: A rule set is a logical collection of rules. A single rule set can have many rules. For example, you might have a rule set that determines a customer's asset balance and another rule set that determines a customer's debt level. SAS Decision Manager displays rules sets in decision tables. Each row of the decision table defines the conditions and actions for one rule. By using SAS Decision Manager, you can easily create new rule sets, reorder the rules in a rule set, add new rules to existing rule sets, and more.

You can also manage rule sets and rule flows. When a rule set or rule flow is published, the versioning features of SAS Decision Manager create a static version of the rule set or rule flow. This static version helps you to enforce integrity and governance over the rule sets and rule flows that are put into production.

rule flow authoring and publishing
A rule flow is a logical collection of rule sets. A rule flow defines a set of rule sets and the order in which they will be executed. A single rule flow frequently corresponds to a single decision. For example, a rule flow can initially execute the rule set that determines a customer's asset balance. Next, the rule set that determine a customer's debt level is executed. Finally, the rule set that assign's a customer's loan application status is executed.

SAS Decision Manager makes it easy to combine rules sets into a rule flow and to publish those rule flows to the metadata server. After a rule flow has been published, it is available for use by other applications.

Create and Publish Business Rules

To create and publish business rules using SAS Decision Manager:

1. Add data tables to your list of data sources.
2. Create business rule folders where you want to save the business rules.
3. Create vocabularies.
4. Create entities and terms.
5. Create rule sets and rules.
6. Create rule flows.
7. (Optional) Test rule flows.
8. Publish rule flows.

After a rule flow has been published, it is available for use by other applications such as SAS Data Integration Studio. These applications map objects in the SAS Decision Manager database to objects in the input data. For example, terms are mapped to table columns or to data set variables. The output generated when a rule flow is executed is written to a data set. The location of the data set is specified by the application.

Model Management

About Managing Models

Using SAS Decision Manager, you can store models, and organize them within projects or folders, validate candidate models, assess candidate models for champion model selection, and publish and monitor champion models in a production environment, and retrain models. All model development and model maintenance personnel, including data modelers, validation testers, scoring officers, and analysts, can use SAS Decision Manager.

Here are some of the services SAS Decision Manager provides:

- Use a single interface to access all of your business modeling projects. All models are stored in a central, secure model repository. Models can also be accessed in one place using the model inventory list in the Inventory category.
- Track the progress of your project's version by creating processes, definitions, and tests. You create custom processes, definitions, and tests to meet your business requirements and to match your business processes.
- Use data tables that are registered in the SAS Metadata Repository.
- Import SAS Enterprise Miner models, SAS/STAT linear models, SAS/ETS COUNTREG and SEVERITY models, models that you develop using SAS code, PMML models, and R models. You can also import a generic model and the model's files in to a folder. You can create custom model templates for SAS code models so that SAS Decision Manager knows exactly what files and metadata are associated with a model.
- You can schedule and run scoring tests, performance monitoring, and retraining to validate models.
- Run several reports to compare and assess candidate models. You can also write your own SAS reporting programs to run and assess candidate models. The aggregated reporting facility enables you to combine multiple reports into a single report. Dashboard reports enable you to monitor the state of projects using performance monitoring reports and can be viewed in a web browser.
- Publish models to the SAS Metadata Repository or a SAS channel. You can also publish the champion model and challenger models to a database for scoring. The SAS Scoring Accelerator is used by SAS Decision Manager to publish models to a database or Hadoop.

Data tables are an integral part of the modeling process. You can use project input and output prototype tables, as well as scoring input and output prototype tables to define variables. Data tables are used for scoring, testing, and performance monitoring. Performance data can be created from your operational data, provided that it has the required structure (for example, the data contains a target variable).

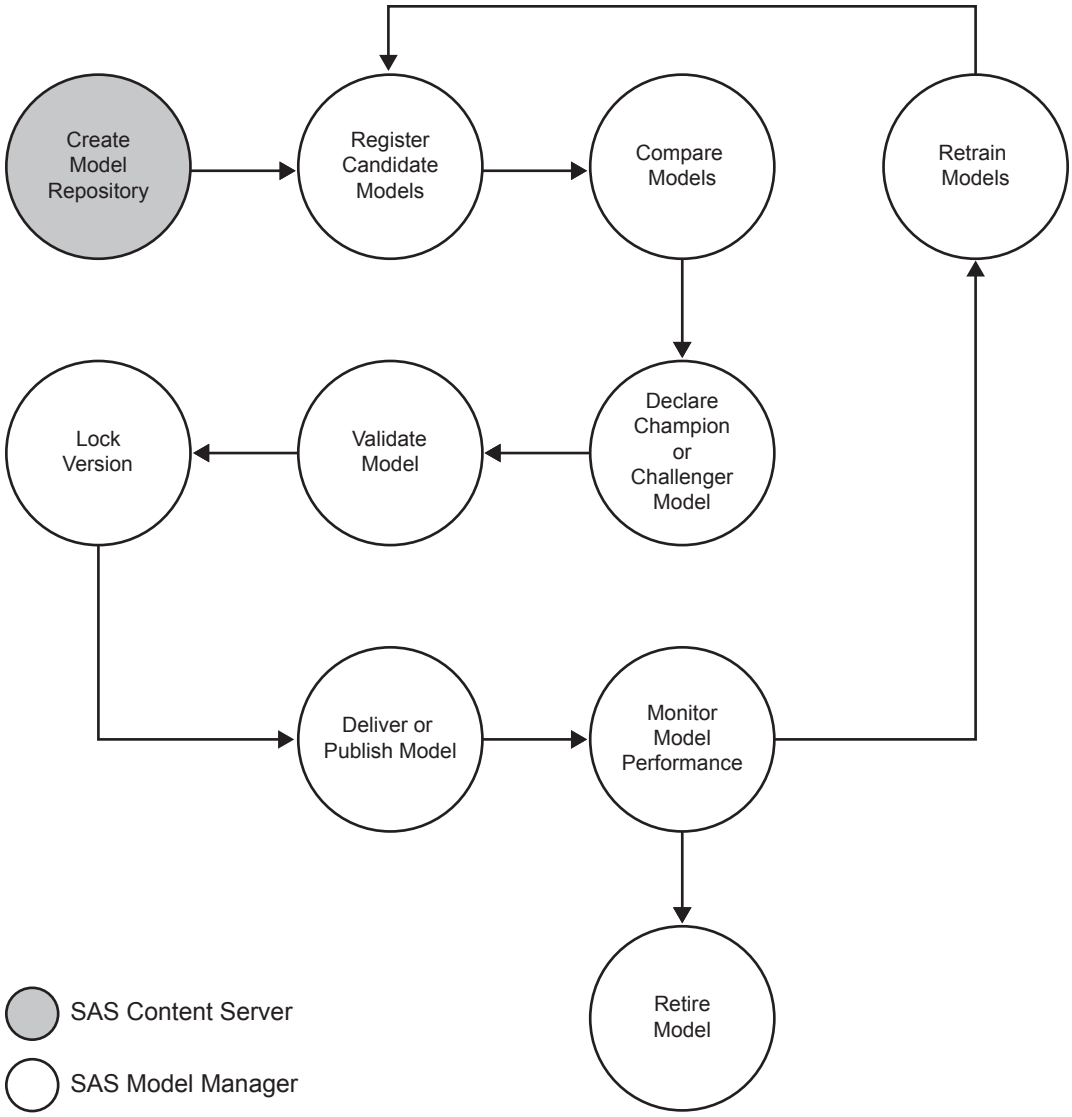
You can also create multiple projects in a portfolio. Additional versions can then be created for all projects within the portfolio. Champion models for all projects within the portfolio can be monitored for performance, and published to the SAS Metadata Repository. SAS Factory Miner models can also be registered to the SAS Decision Manager model repository. The SAS Factory Miner projects are managed as portfolios in SAS Decision Manager. The project segments and models are available within a portfolio and can be managed from the Portfolios category within SAS Decision Manager.

Any user who is registered in SAS Management Console can be assigned to a SAS Decision Manager group, and can then work in SAS Decision Manager. For more information, see "Configuring Users, Groups, and Roles" in *SAS Decision Manager: Administrator's Guide*.

Model Management Process

The following diagram illustrates the model management process:

Figure 1.1 *The Model Management Process*



Here is a summary of the model management process:

- **Create Model Repository**: create a secure model repository on the SAS Content Server where SAS code, input and output files, and metadata that is associated with a model can be stored.
- **Register Candidate Models**: register input and output files, and then import and configure a model.
- **Compare Models**: perform scoring tests and create comparison reports for the models by using test data sources.
- **Declare Champion or Challenger Model**: declare the model as champion or challenger to use for testing and production phases of the workflow.
- **Validate Model**: perform scoring tests and create validation reports for the champion model and challenger models by using test data sources.
- **Lock Version**: lock a version when the champion model is approved for production.
- **Deliver or Publish Model**: publish a champion or challenger models to a SAS publish channel, to a database, or to the SAS Metadata Repository.

- **Monitor Model Performance**: provide comparative model performance benchmarking.
- **Retrain Models**: select models to retrain in response to data or market changes.
- **Retire Model**: retire a model from production.

Here is an example of the model management process for comparing a challenger model to the champion model to determine the best champion model:

1. Register candidate models in the version that is under development.
2. Create a Dynamic Lift report and compare the model to the champion model. Flag the model as a challenger based on the results of the Dynamic Lift report.
3. Perform scoring tests with the champion and challenger models in real time or in batch. This step can be performed outside SAS Decision Manager.
4. Publish the challenger model to a database or to the SAS Metadata Repository.
5. Prepare performance data sources, which include both the actual outcome variable and predicted variable.
6. Create and execute the performance monitoring for the champion and challenger models to create reports to compare and validate the champion model and challenger models. One of the reports that is available for this comparison is the Champion and Challenger Performance report.
7. Set the challenger model as the project champion if the challenger is good enough to be promoted. Go to step 3, or consider building another model as a challenger with existing or a new input training data source.
8. Publish the new project champion model with or without a new challenger model.

About Decision Management

SAS Decision Manager includes Decision Builder. Decision Builder enables you to combine analytical models, rule flows, 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. You can automate the decision-making process by operationalizing your analytics and business rules. Automating decisions with Decision Builder provides a streamlined mechanism for controlling and monitoring the rules and processes used by your organization.

Managing Preferences

About Setting Preferences

Preferences provide a way for you to customize the user interface. Preferences for each user are stored in metadata and are retained if your deployment is migrated or reconfigured.

You can set preferences in two ways:

by using the Preferences window
: To open the Preferences window, select **File** ⇨ **Preferences**. There are two general categories of preferences: Global and Decision Manager preferences. See "Global Preferences" and "Decision Manager Preferences" on page 9 for more information.

by using SAS Preferences Manager
: SAS Preferences Manager is a web application that provides a central facility for users to manage their preferences and settings. See "SAS Preferences Manager" on page 10 for more information.

Global Preferences

Global preferences apply to all SAS web applications that are displayed with the Adobe Flash Player. When you set a global preference, it applies only to the user that you are logged on as.

To set global preferences, select the **Global Preferences** page. The following global preferences are available:

User locale
: specifies the geographic region whose language and conventions are used in the applications. This setting might also apply to some SAS web applications that are not displayed with the Adobe Flash Player. The default is the browser locale. Locale changes take effect after you log off and log back on.

Note: You can also set the **User locale** setting by using the SAS Preferences Manager. Select the **Regional** menu option in SAS Preferences Manager. For more information, see "SAS Preferences Manager" on page 10.

Note: If the user locale that you specify in the preferences is different from the user locale for the SAS Workspace Server, you might receive an error when you try to sign in to the application. You might also receive encoding errors when executing tasks in SAS Decision Manager. If you receive an error, change the updated locale back to the original locale.

Theme
: specifies the collection of colors, graphics, and fonts that appear in the applications. Your site administrator can change the default theme. A theme change might take a few seconds to apply if many items and features are open in the application.

Invert application colors
: inverts all of the colors in the application window, including both text and graphical elements. You can also temporarily invert or revert the colors for an individual application session by pressing Ctrl+~.

Override settings for focus indicator
: controls the appearance of the highlighting that surrounds the currently selected field in the SAS Decision Manager interface.

Decision Manager Preferences

Decision Manager preferences apply to SAS Decision Manager only. To set these preferences, select **Decision Manager** ⇨ **General**.

Show this number of recent items
: controls the number of items that are listed in the **Recent Work** menu. To display this menu, select **File** ⇨ **Recent Work**.

SAS Preferences Manager

SAS Preferences Manager is a web application that provides a common mechanism for managing preferences for SAS web applications. The application enables users to manage their preferences and administrators to set default preferences for locale, theme, alert notification, time, date, and currency.

To launch the SAS Preferences Manager, enter the URL http://*host-name*:*port*/SASPreferences in your browser window. Replace the values for host-name and port based on the location of the configured SAS Web Infrastructure Platform. For more information, see "SAS Preferences Manager" in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Change the Delivery Type for Alert Notifications

The default delivery type for notifications is specified in the properties for the SAS Application Infrastructure by using the Configuration Manager plug-in to SAS Management Console. For SAS 9.4, the default delivery type is **`My alerts portlet`**. You can use SAS Preferences Manager to change your default delivery type.

Note: A SAS administrator can modify the default notification type for all users. For information about modifying the default delivery type for all users, see "Configure Alert Notifications for SAS Workflow" in *SAS Decision Manager: Administrator's Guide*.

To specify the notification delivery preference for an individual user:

1. Enter the URL http://*host-name*:*port*/SASPreferences in your browser window to launch the SAS Preferences Manager. Replace the values for host-name and port based on the location of the configured SAS Web Infrastructure Platform.
2. Enter the user ID and password for the user account that you use to access SAS web applications.
3. Select **General** ⇨ **Notifications**.
4. Select a format type for the e-mail notifications. The options are **`HTML-formatted e-mail`** and **`Plain-text e-mail`**.
5. Select the notification types from the **Available** list and click to add the selected notification types.

   TIP To remove a notification type, select the type from the list and click .
6. Click **Apply** to update the notification settings, and click **OK** to save the changes.

For more information, see "SAS Preferences Manager" in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Viewing Help and Documentation

SAS Decision Manager provides the following types of Help and documentation:

How-to Help
: How-to Help provides quick instructions or tips to help you complete some tasks in the application. To access how-to Help, select **Help** ⇨ **How To**.

Embedded Help

Help pop-up menus and tooltips provide brief descriptions of various fields.

To access a Help pop-up menu for a field, click the Help icon (?) when it appears next to a field. You can also place the mouse pointer over an element in the SAS Decision Manager windows to view the associated tooltip.

SAS Decision Manager: User's Guide

This document provides detailed information about the concepts and tasks that are related to using SAS Decision Manager. This document is available at http://support.sas.com/documentation/onlinedoc/edm.

SAS Decision Manager: Administrator's Guide

This document contains information about the administration tasks that are required to set up and configure the SAS Decision Manager and is available at http://support.sas.com/documentation/onlinedoc/edm.

Additional resources are available from the **Help** menu. To access these resources, select **Help** ⇨ **SAS on the Web**.

Part 2

Getting Started

Chapter 2

Quick Start Tutorial

Overview of the Quick Start Tutorial

This Quick Start tutorial is an introduction to some of the primary features of SAS Decision Manager. The tutorial covers basic tasks for creating and publishing rule flows, managing models, and creating and deploying decisions. The tutorial also shows you how to validate the installation and configuration of SAS Decision Manager at your site.

In this tutorial, you import a vocabulary, create a rule set, create and publish a rule flow, import models, set and publish a champion model, score models, monitor model performance, and create and deploy a decision.

Note: In order to complete the tasks in this tutorial, your user ID must be a member of the Decision Manager Users group or must be granted equivalent permissions. Also, to complete the steps related to model management, your ID must be a member of either the Model Manager Advanced Users group or the Model Manager Administrator Users group. See "Configuring Users, Groups, and Roles" in *SAS Decision Manager: Administrator's Guide* for more information.

With the exception of Step 1, the steps in this tutorial are basic steps that are required to add content to the SAS Decision Manager database and model repository. In this tutorial, you complete the following steps:

1. Make the tutorial files available on the SAS Application Server.

   Note: The QuickStartTutorial.zip file contains data and model files for several tutorials, including this one.

2. Sign in to SAS Decision Manager.
3. Define the data source needed for the tutorial.
4. Define business rule folders.
5. Create a vocabulary and import the terms.
6. Create a rule set and define business rules.
7. Create and publish a rule flow.
8. Define and create the components of the model hierarchy.
9. Import models.
10. Compare models using reports.
11. Set a champion model.
12. Create a scoring test and run model score code.
13. Monitor the performance of the champion model.
14. Publish the champion model to the SAS Metadata Repository.
15. Create and publish a decision.
16. Deploy the decision in SAS Data Integration Studio.

Make the Tutorial Files Available

About Making the Tutorial Files Available

The tutorial is designed to use the SAS Metadata Repository. Before you use tables in the SAS Metadata Repository, the tutorial data sets and models must be on the SAS Application Server. An administrator who has Write access to the server and a valid SASApp user ID and password can put the tables there.

Some parts of this tutorial require files other than data sets and models, such as score code and templates. These files do not need to be registered in the SAS Metadata Repository. The drive where you extract the tutorial ZIP file must be accessible to the SAS Metadata Repository and to tutorial users. Tutorial users can also extract tutorial ZIP files to their local computers in order to access the other files.

You can define a data library and register the tables in the SAS Metadata Repository using the Data category view in SAS Decision Manager.

Download the Tutorial Files

The ZIP file QuickStartTutorial.zip contains the tutorial's data sets, models, and score code, and is available at http://support.sas.com/documentation/onlinedoc/edm/. Before you begin the tutorial, extract the tutorial files to a computer that is accessible to the SAS Metadata Server and to SAS Decision Manager users. If your SAS Metadata Server is separate from the SAS Application Server, the files must be placed on the SAS Application Server. Use WinZip to extract the files. If you are using a different extraction program, follow that program's instructions for extracting the files.

To download the files:

1. Create a folder on the server machine or a network drive to store the tutorial files. The instructions refer to this folder as **`<drive>`**.

   Note: Users must have Read, Write, and Execute permissions to this folder and subfolders. You can create a group and add the tutorial users to that group to grant the permissions. For more information, see "Creating Operating System Accounts for Product Administrators and Users" in *SAS Decision Manager: Administrator's Guide*.

2. Save the QuickStartTutorial.zip to ***`<drive>`***.

3. Open Windows Explorer to ***`<drive>`***. Right-click **QuickStartTutorial.zip** and select **Open**. Click **Open**.

4. Click the arrow on the **Unzip** button to open the Unzip from WinZip File Folder window.

   Note: If you are using a previous release of Windows, from the WinZip window, click the **Extract** button. The Extract dialog box appears.

5. Select **`<drive>`** from the Unzip to WinZip File Folder window.

   Note: If you are using a previous release of Windows, in the **Extract to** box, select **`<drive>`** and click **Extract**.

You can find the data and models files for each tutorial in the respective tutorial folder (for example, **`<drive>\QuickStartTutorial\Data`** or **`<drive>\QuickStartTutorial\Models`**).

6. UNIX only: To complete the tutorial in a UNIX environment, locate the CPORT file. Files that you use to import the data sets into UNIX are located in the QuickStartTutorial.zip file. Instructions and the sample code for performing an import are provided in the Readme.txt file. In order for a transport file to be imported successfully, the encodings of the source and target SAS sessions must be compatible. Use either latin1 or UTF-8 for your SAS sessions. For more information, see "CIMPORT Problems: Importing Transport Files " in *Base SAS Procedures Guide*.

Sign In

To sign in to SAS Decision Manager:

1. In the address bar of your web browser, enter the URL for SAS Decision Manager and press **Enter**. The Sign In page appears.

   Note: Contact your system administrator if you need the URL for SAS Decision Manager. The default URL is http://*host_name*:*port*/SASDecisionManager.

   Note: The default URL for Decision Builder is http://*host_name*:*port*/SASDecisionManager/DecisionBuilder.

2. Enter a user ID and password. Your user ID might be case sensitive, depending on the operating system that is used to host the application server. Your password is case sensitive.

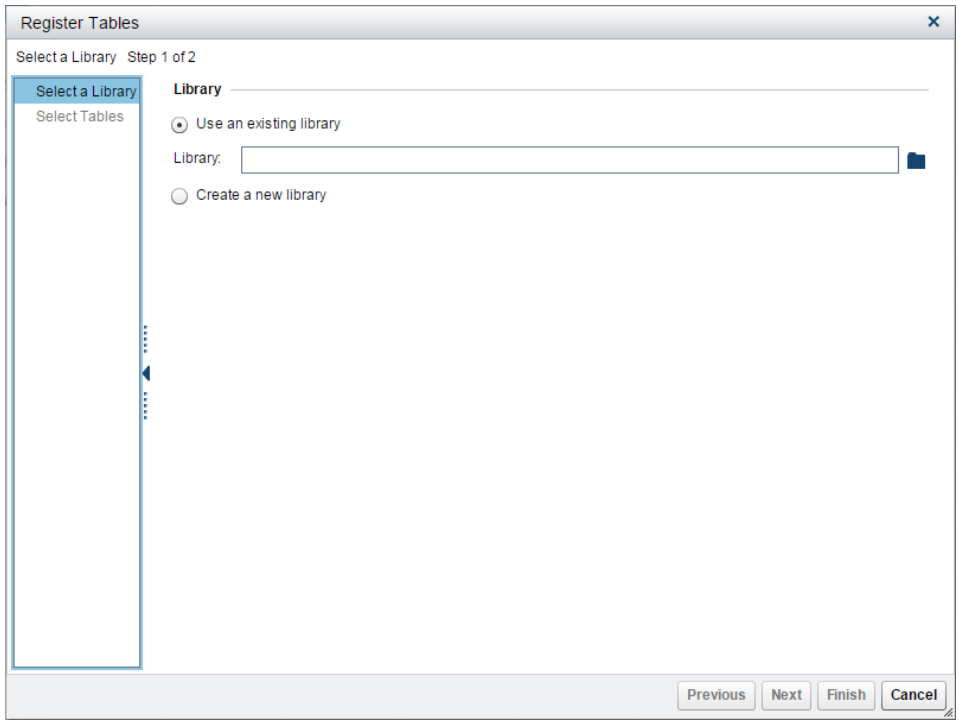
   Note: To schedule jobs in a Windows environment, you must include the domain name when entering your user ID (for example, *domain**myuserID*).

3. Click **Sign In**.

Define Data Sources

To register new tables in the SAS Metadata Repository and add them to the list of data sources:

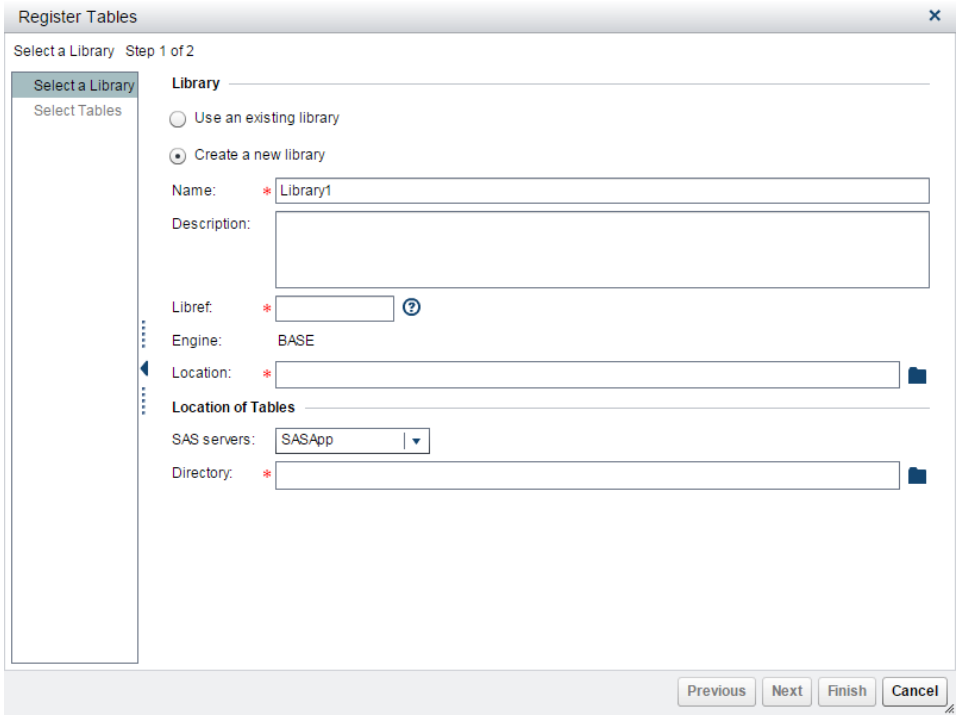
1. Select **Data** ⇨ **Tables**.
2. Click + and select **Register Tables**. The Register Tables window appears.



Note: You cannot use the **Register Tables** option to add a table that has already been registered in the SAS Metadata Repository using the SAS Management Console. You must select **Add Registered Tables** instead. See "Add Tables That Are Registered in Metadata" on page 48.

3. Create a new Base SAS library.

   a. Select **Create a new library**.



   b. Specify **QSTutorial** for the name of the new library. The name cannot exceed 60 characters.

c. (Optional) Specify a description for the library.

d. Specify **QSTut** for the libref.

e. Specify the location for the new library. This location is the folder in the SAS Metadata Repository where the library is stored.

f. Select the server and the directory where the data tables for the quick start tutorial reside (for example, `C:\QuickStartTutorial\Data`).

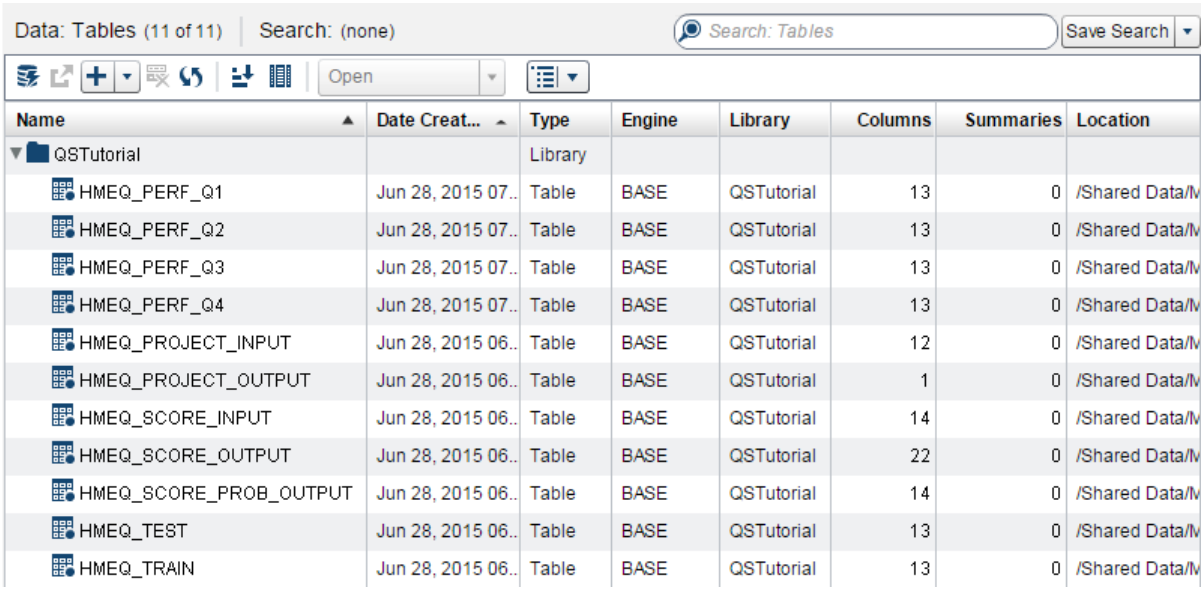
g. Click **Next**.

Note: If you click **Cancel** at this point, a folder for the library is created in the SAS Metadata Repository, but the folder does not appear in the list of data tables.

4. Click to add all of the tables to the **Selected tables** list.

5. Click **Finish**. The new library is now available in the list of data tables.

Figure 2.1 *SAS Decision Manager Data: Tables List*



Name	Date Creat...	Type	Engine	Library	Columns	Summaries	Location
QSTutorial		Library					
HMEQ_PERF_Q1	Jun 28, 2015 07..	Table	BASE	QSTutorial	13	0	/Shared Data/M
HMEQ_PERF_Q2	Jun 28, 2015 07..	Table	BASE	QSTutorial	13	0	/Shared Data/M
HMEQ_PERF_Q3	Jun 28, 2015 07..	Table	BASE	QSTutorial	13	0	/Shared Data/M
HMEQ_PERF_Q4	Jun 28, 2015 07..	Table	BASE	QSTutorial	13	0	/Shared Data/M
HMEQ_PROJECT_INPUT	Jun 28, 2015 06..	Table	BASE	QSTutorial	12	0	/Shared Data/M
HMEQ_PROJECT_OUTPUT	Jun 28, 2015 06..	Table	BASE	QSTutorial	1	0	/Shared Data/M
HMEQ_SCORE_INPUT	Jun 28, 2015 06..	Table	BASE	QSTutorial	14	0	/Shared Data/M
HMEQ_SCORE_OUTPUT	Jun 28, 2015 06..	Table	BASE	QSTutorial	22	0	/Shared Data/M
HMEQ_SCORE_PROB_OUTPUT	Jun 28, 2015 06..	Table	BASE	QSTutorial	14	0	/Shared Data/M
HMEQ_TEST	Jun 28, 2015 06..	Table	BASE	QSTutorial	13	0	/Shared Data/M
HMEQ_TRAIN	Jun 28, 2015 06..	Table	BASE	QSTutorial	13	0	/Shared Data/M

Define Business Rule Folders

All of the content in your business rules database is contained within business rules folders. You must define at least one top-level folder. To define a top-level folder:

1. Select any category under **Business Rules**, such as **Business Rules** ⇨ **Vocabularies**.

2. Click , and select **New Top-Level Folder**.

3. Enter `Tutorials` for the folder name.

4. If you are a business rules folder administrator, you must also select the group that needs access to this folder. Select **Decision Manager Users**.

5. Click **OK**.

Because multiple users might want to perform the tasks in the tutorial, each user should create a separate folder in the **Tutorials** folder. To create a new folder:

1. Right-click the **Tutorials** folder and select **New Folder**.
2. Enter a name for the folder such as **`myUserID`**. The examples in this tutorial use the ID **sasdemo**.
3. Click **OK**.

For more information, see Chapter 4, "Managing Business Rule Folders," on page 59.

Create a Vocabulary, Entities, and Terms

Vocabularies, entities, and terms are the basic building blocks of a business rules database. Vocabularies contain entities, and entities contain terms. In this tutorial, you use the data table HMEQ_SCORE_PROB_OUTPUT, which you registered in "Define Data Sources" to import the terms for a new vocabulary. Then, you create a new entity and term manually in SAS Decision Manager and rename two of the imported terms.

For more information about vocabularies, entities, and terms, see Chapter 5, "Managing Vocabularies," on page 61.

Create a New Vocabulary

To create a vocabulary:

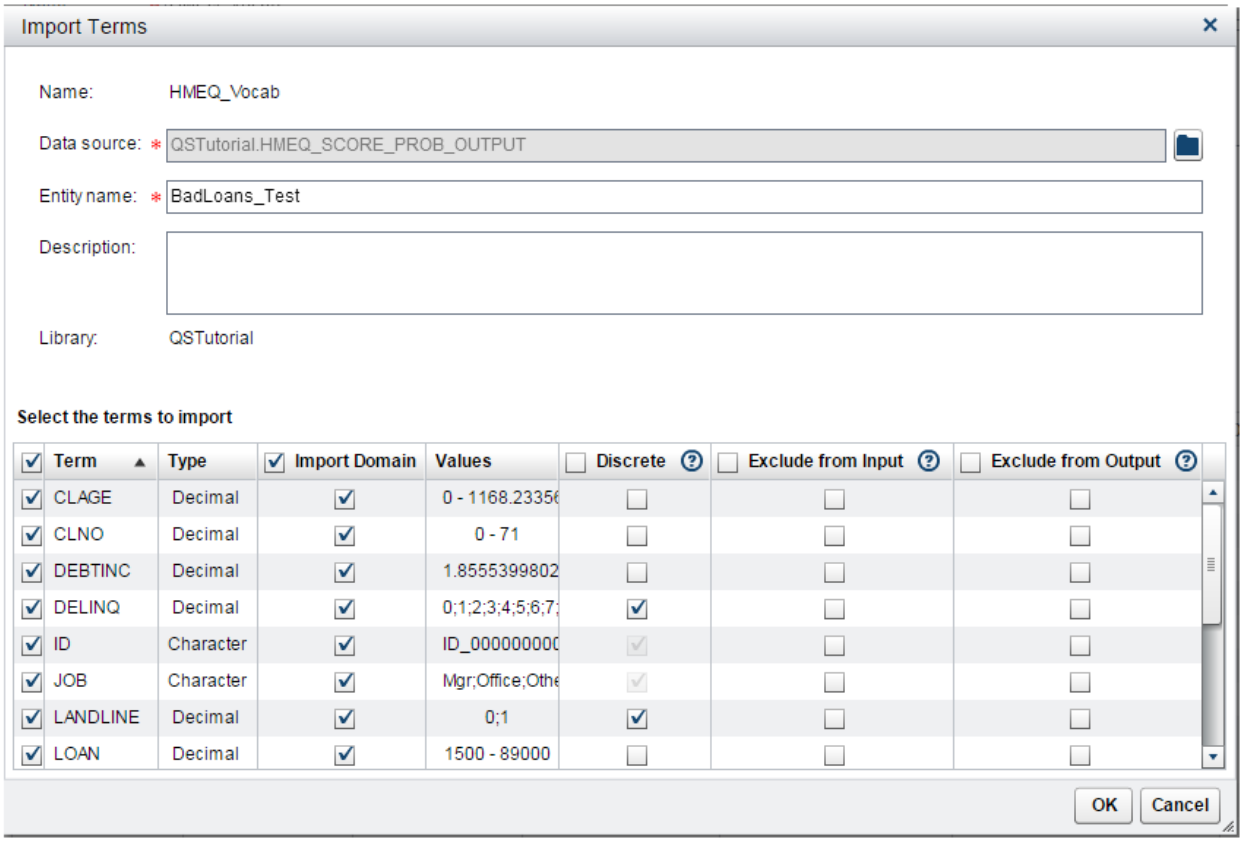
1. Select **Business Rules** ⇨ **Vocabularies**.
2. Right-click on your folder in the **Tutorials** folder, and select **New Vocabulary**.
3. Enter **`HMEQ_Vocab`** for the vocabulary name, and click **OK**.

Import Terms from an Input Data Set

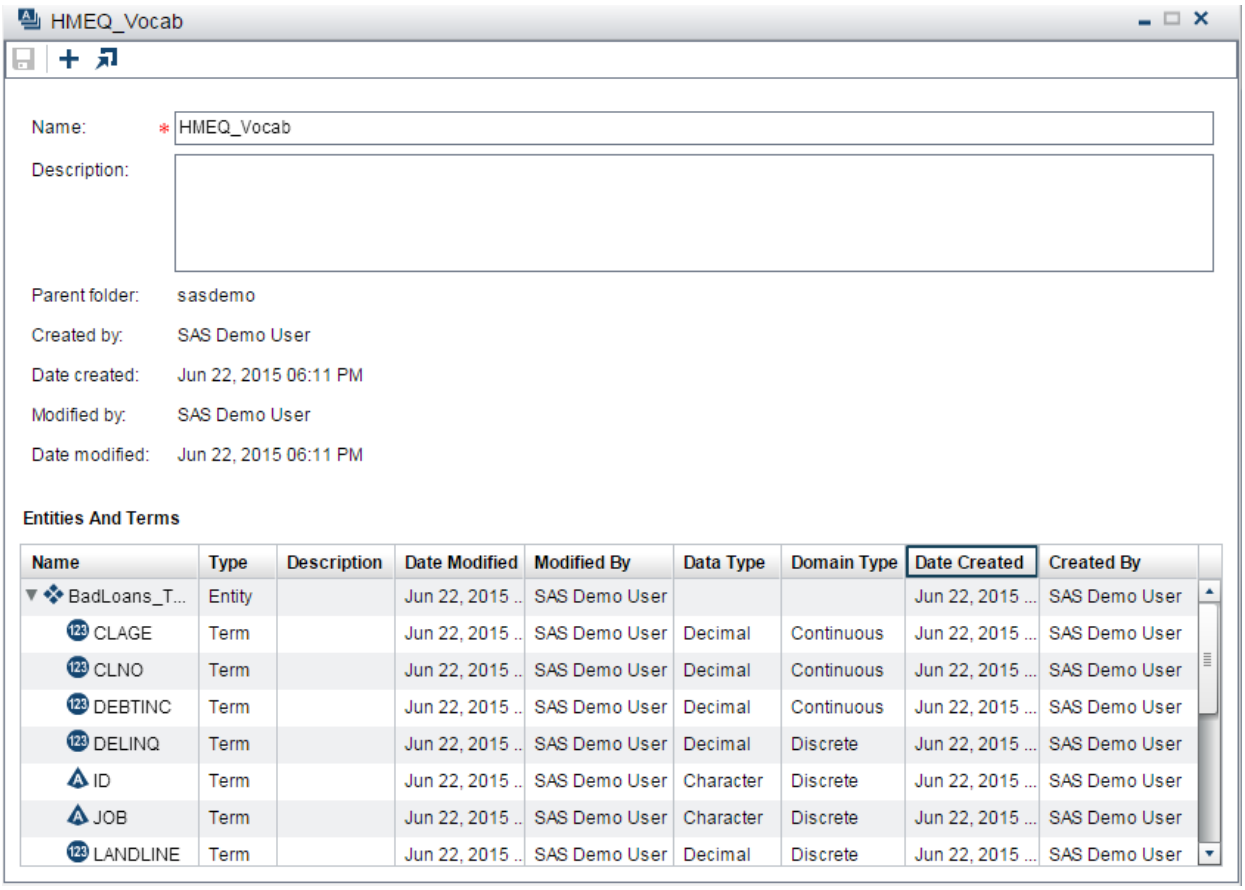
The easiest way to create new entities and terms is to import them from an input data set. The following steps import all of the terms in the data table HMEQ_SCORE_PROB_OUTPUT, which you registered in the SAS Metadata Repository in "Define Data Sources" on page 18.

4. Double-click **HMEQ_Vocab** to open the vocabulary.
5. Click to import the vocabulary entities and terms. The Import Terms window appears.
6. Select **HMEQ_SCORE_PROB_OUTPUT** as the data source, and enter **`BadLoans_Test`** as the entity name.

7. Select the check box to the left of the **Term** table header to select all of the terms, and click **OK**.



SAS Decision Manager imports the terms and adds them to the entity named BadLoans_Test. If you expand the list, you see all of the terms that were imported.



Name	Type	Description	Date Modified	Modified By	Data Type	Domain Type	Date Created	Created By
BadLoans_T...	Entity		Jun 22, 2015 ..	SAS Demo User			Jun 22, 2015 ...	SAS Demo User
CLAGE	Term		Jun 22, 2015 ..	SAS Demo User	Decimal	Continuous	Jun 22, 2015 ...	SAS Demo User
CLNO	Term		Jun 22, 2015 ..	SAS Demo User	Decimal	Continuous	Jun 22, 2015 ...	SAS Demo User
DEBTINC	Term		Jun 22, 2015 ..	SAS Demo User	Decimal	Continuous	Jun 22, 2015 ...	SAS Demo User
DELINQ	Term		Jun 22, 2015 ..	SAS Demo User	Decimal	Discrete	Jun 22, 2015 ...	SAS Demo User
ID	Term		Jun 22, 2015 ..	SAS Demo User	Character	Discrete	Jun 22, 2015 ...	SAS Demo User
JOB	Term		Jun 22, 2015 ..	SAS Demo User	Character	Discrete	Jun 22, 2015 ...	SAS Demo User
LANDLINE	Term		Jun 22, 2015 ..	SAS Demo User	Decimal	Discrete	Jun 22, 2015 ...	SAS Demo User

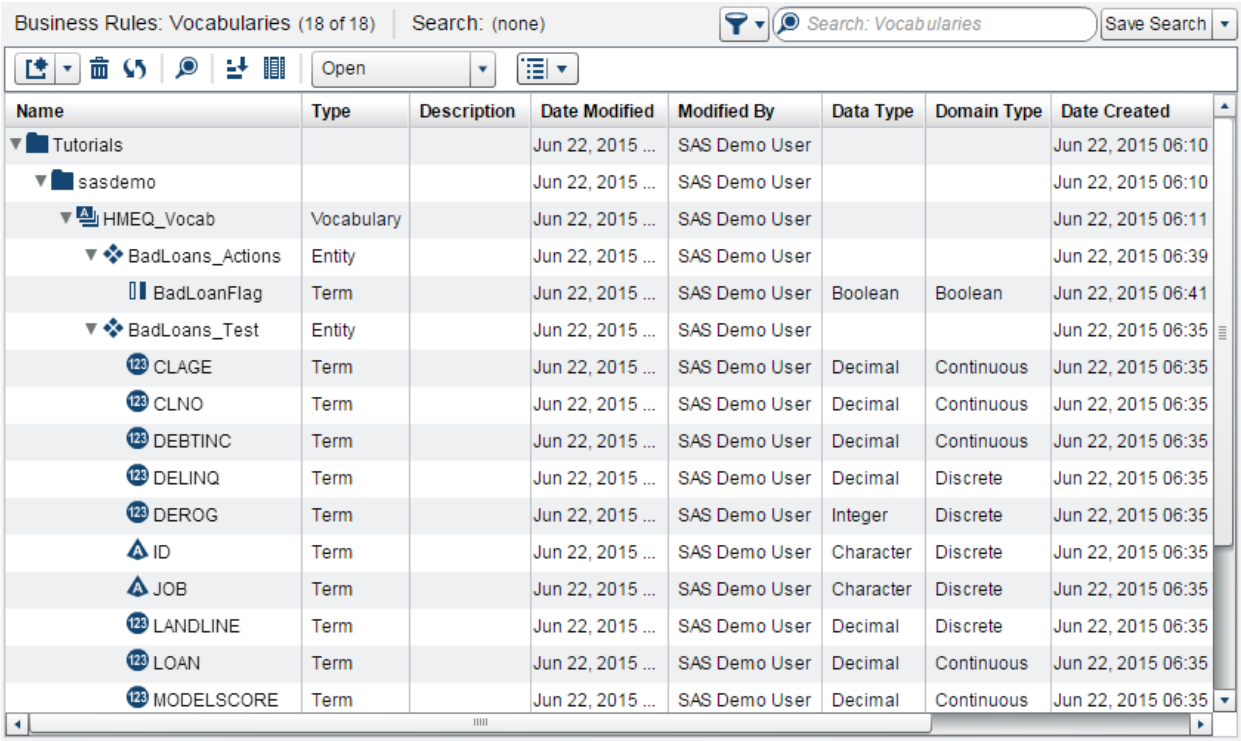
8. Click ✕ to close the vocabulary.

Create a New Entity and New Terms

In addition to importing terms, you can create entities and terms manually. The following steps create a new entity and two new terms.

9. In the Business Rules: Vocabularies category view, expand **HMEQ_Vocab**, right-click the **BadLoans_Test** entity, and select **New Term**. The New Term window appears.
10. Enter **`DEROG`** as the term name, select **Integer** as the data type, and click **OK**.
11. Right-click **HMEQ_Vocab** and select **New Entity**. The New Entity window appears.
12. Enter **`BadLoans_Actions`** as the entity name, and click **OK**.
13. Right-click **BadLoans_Actions** and select **New Term**. The New Term window appears.
14. Enter **`BadLoanFlag`** for the term name. Select **Boolean** for the data type, select **Exclude from input**, and click **OK**.

The **HMEQ_Vocab** vocabulary now contains two entities and several terms. You can display all of the terms in the vocabulary in the category view.



Name	Type	Description	Date Modified	Modified By	Data Type	Domain Type	Date Created
Tutorials			Jun 22, 2015 ...	SAS Demo User			Jun 22, 2015 06:10
sasdemo			Jun 22, 2015 ...	SAS Demo User			Jun 22, 2015 06:10
HMEQ_Vocab	Vocabulary		Jun 22, 2015 ...	SAS Demo User			Jun 22, 2015 06:11
BadLoans_Actions	Entity		Jun 22, 2015 ...	SAS Demo User			Jun 22, 2015 06:39
BadLoanFlag	Term		Jun 22, 2015 ...	SAS Demo User	Boolean	Boolean	Jun 22, 2015 06:41
BadLoans_Test	Entity		Jun 22, 2015 ...	SAS Demo User			Jun 22, 2015 06:35
CLAGE	Term		Jun 22, 2015 ...	SAS Demo User	Decimal	Continuous	Jun 22, 2015 06:35
CLNO	Term		Jun 22, 2015 ...	SAS Demo User	Decimal	Continuous	Jun 22, 2015 06:35
DEBTINC	Term		Jun 22, 2015 ...	SAS Demo User	Decimal	Continuous	Jun 22, 2015 06:35
DELINQ	Term		Jun 22, 2015 ...	SAS Demo User	Decimal	Discrete	Jun 22, 2015 06:35
DEROG	Term		Jun 22, 2015 ...	SAS Demo User	Integer	Discrete	Jun 22, 2015 06:35
ID	Term		Jun 22, 2015 ...	SAS Demo User	Character	Discrete	Jun 22, 2015 06:35
JOB	Term		Jun 22, 2015 ...	SAS Demo User	Character	Discrete	Jun 22, 2015 06:35
LANDLINE	Term		Jun 22, 2015 ...	SAS Demo User	Decimal	Discrete	Jun 22, 2015 06:35
LOAN	Term		Jun 22, 2015 ...	SAS Demo User	Decimal	Continuous	Jun 22, 2015 06:35
MODELSCORE	Term		Jun 22, 2015 ...	SAS Demo User	Decimal	Continuous	Jun 22, 2015 06:35

Create a Rule Set and Define Business Rules

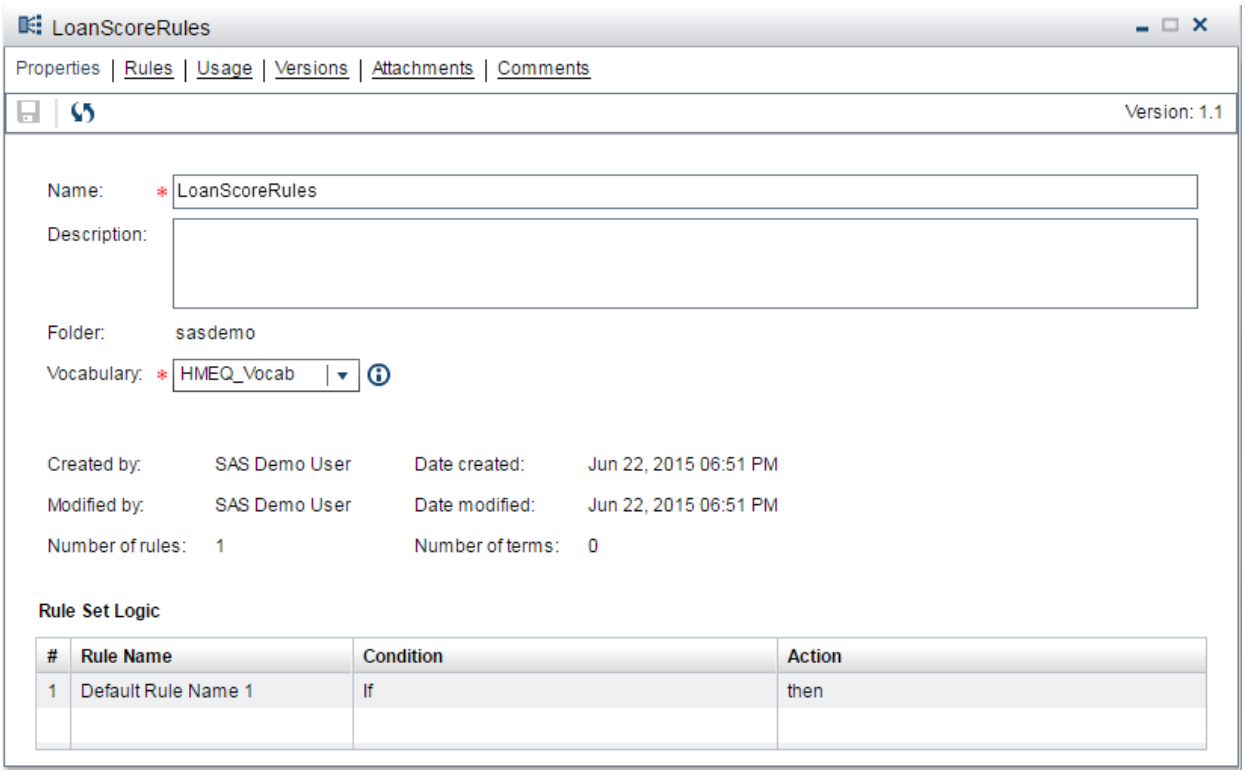
A business rule specifies conditions to be evaluated and actions to be taken if those conditions are satisfied. Rules are grouped together into rule sets. In this tutorial, you create a single rule set with three rules.

For more information, see Chapter 7, “Managing Rules and Rule Sets,” on page 75.

Create a New Rule Set

1. Select **Business Rules** ⇨ **Rule Sets**.
2. Select your folder in the **Tutorials** folder.
3. Click and select **New Rule Set**. The New Rule Set window appears.
4. Enter `LoanScoreRules` for the rule set name.
5. Click , select **HMEQ_Vocab**, and then click **OK**.

6. Click **Create** in the New Rule Set window. SAS Decision Manager opens the rule set and displays the **Properties** page.



Define Business Rules

To define the business rules for the new rule set, you enter condition and action expressions for the terms in the rule set into the rule set editor. The following steps define three simple rules that determine whether a loan application is a bad risk. The determination is based on the model score that was generated for the loan, among other things. For example, if the model score is less than or equal to 0.6, the loan is considered a bad risk, and the **BadLoanFlag** term is set to `True`.

7. Select the **Rules** page.
8. Expand the **BadLoans_Test** entity, and select the **LANDLINE**, **MODELSCORE**, and **REASON** terms.
9. Right-click on one of the highlighted terms, and select **Use as Condition Term**.
10. Right-click the **BadLoanFlag** term, and select **Use as Action Term**.
11. Enter the rule expressions into the rule set editor. Each row in the table represents a different rule. Enter the expressions for each term into the column for that term. You can enter expressions directly into the table cells, or you can use the equation editor. Click to open the equation editor.

    Enter the expressions in the following table into the rule set editor.

	Condition Terms			Action Term
Rule	**LANDLINE**	**MODELSCORE**	**REASON**	**BadLoanFlag**
Rule 1		<=.6		True

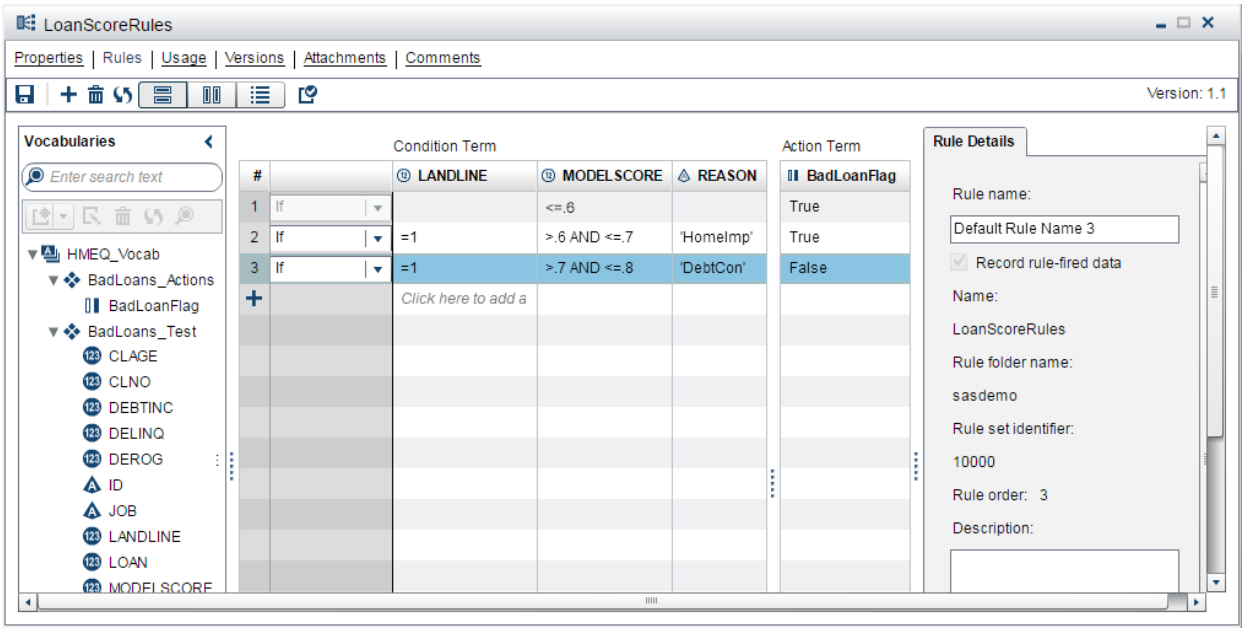
Rule	Condition Terms			Action Term
	LANDLINE	MODELSCORE	REASON	BadLoanFlag
Rule 2	= 1	>.6 AND <=.7	'HomeImp'	True
Rule 3	= 1	>.7 AND <=.8	'DebtCon'	False

TIP If you do not specify an operator at the beginning of an expression, SAS Decision Manager adds an equal sign to the beginning of the expression.

TIP In condition expressions, when an AND or OR operator is followed immediately by another operator, SAS Decision Manager inserts the column term between the AND or OR operator and the operator that follows it.

See "Define Expressions for a Rule" on page 80 and "Terms and Operators Added by SAS Decision Manager" on page 89 for more information.

The following display shows the decision table with the rules that are defined for the **LoanScoreRules** rule set.



Validate, Version, and Save the Rule Set

When you save a rule set, SAS Decision Manager validates the content of the rule set before saving it. However, you can validate the rule set at any time while you are creating the rule set.

12. Click to validate the rule set.
13. Click to save the rule set.
14. On the **Versions** tab, select version 1.1, and then click .
15. Click to close the rule set.

Create and Publish a Rule Flow

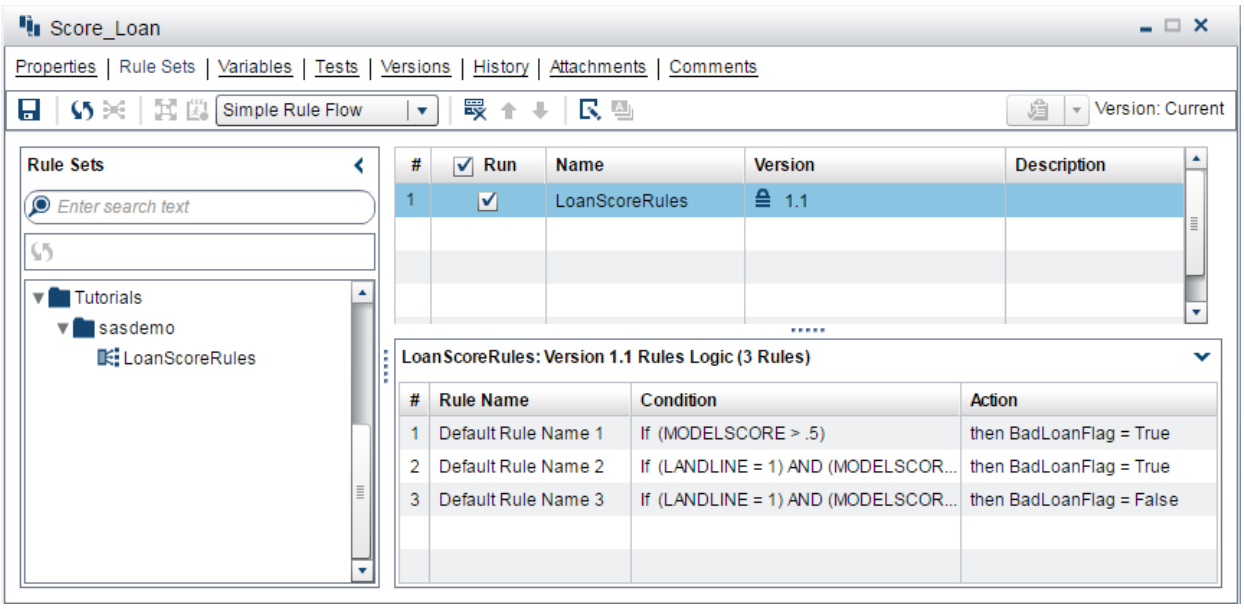
The last major step in adding business rules content to the SAS Decision Manager database is to create and publish rule flows. A business rule flow is a logical collection of rule sets that define multiple conditions and actions. The rule flow that you create in this tutorial contains only one rule set, but most rule flows contain multiple rule sets.

For more information, see Chapter 8, "Creating and Publishing Rule Flows," on page 101.

To create and publish a rule flow:

1. Select **Business Rules** ⇨ **Rule Flows**.
2. Right-click on your folder in the **Tutorials** folder, and select **New Rule Flow**.
3. Enter `Score_Loan` for the rule flow name and click **Create**. SAS Decision Manager opens the rule flow and displays the **Properties** page.
4. Select the **Rule Sets** page.
5. In the rule sets list, right-click **LoanScoreRules** and select **Add To Rule Flow**.
6. Click on **Use latest** in the **Version** column, and select **1.1**.

   TIP Before you can publish a rule flow, all of the rule sets in the rule flow must be locked.



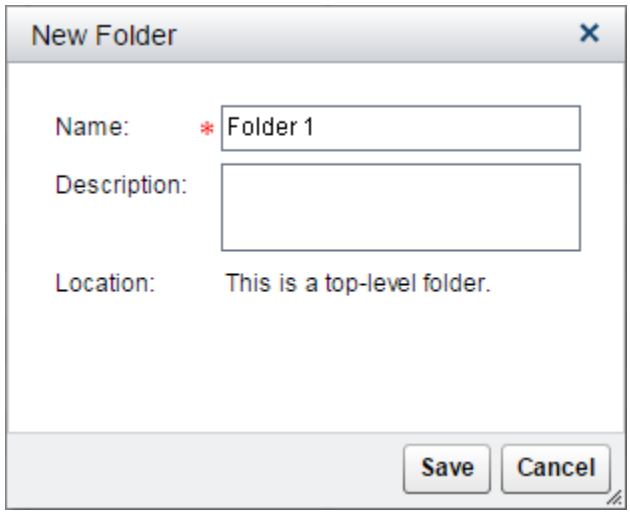
7. Click to save the rule flow.
8. Click to publish the rule flow. The Choose a Location window appears.
9. Select the location where you want to publish the rule flow, and click **OK**. SAS Decision Manager confirms that the rule flow has been published and displays the published name and identification number of the published rule flow.
10. Click **Close** to close the message, and then click to close the rule flow.

Organize the Model Hierarchy

Create Folders

Create a top-level folder for the quick start tutorials:

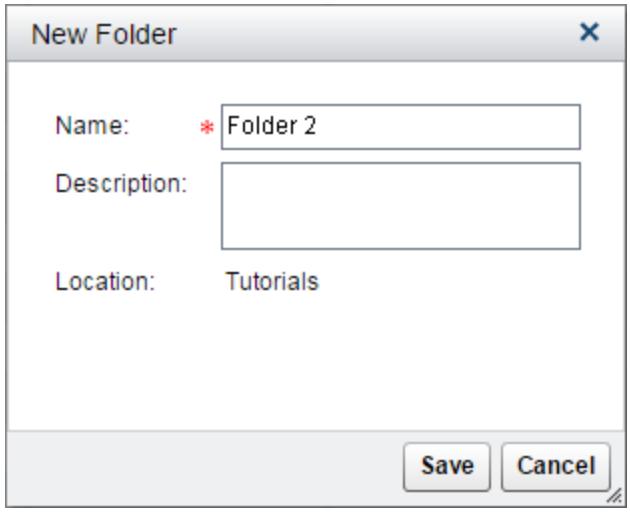
1. Select **Models** ⇨ **Projects**.
2. Click and select **New Top-Level Folder**. The New Folder window appears.



3. Enter **Tutorials** for the name of the folder.
4. (Optional) Enter a description for the folder.
5. Click **Save**.

Because multiple users might want to perform the tasks in the tutorial, it is recommended that each user create their own folder in the **Tutorials** folder. To create a new folder:

1. Select **Tutorials**, click , and select **New Folder**. The New Folder window appears.



2. Enter a name for the folder, such as **myUserID**. The examples in this tutorial use the ID **sasdemo**.
3. (Optional) Enter a description for the folder.
4. Click **Save**.

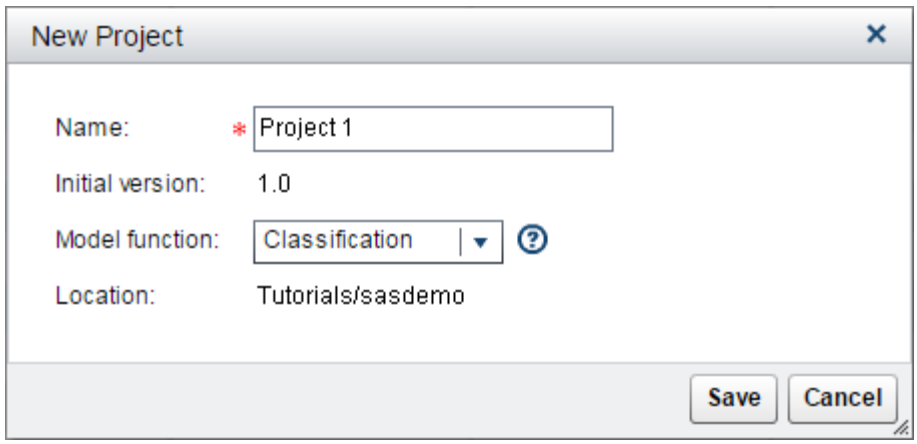
See Also

"Overview of Managing Folders" on page 121

Create a Project

To create a project:

1. Select a folder to store the new project (for example, **myUserID**).
2. Click and select **New Project**. The New Project window appears.



3. Enter **HMEQ** for the name of the project.

   The initial version is displayed and reflects the level for sequential versions.
4. Select **Classification** for the model function.

   Note: The model function (Classification, Prediction, Segmentation, or Analytical) indicates the type of models that should be imported into the project.
5. Click **Save**.

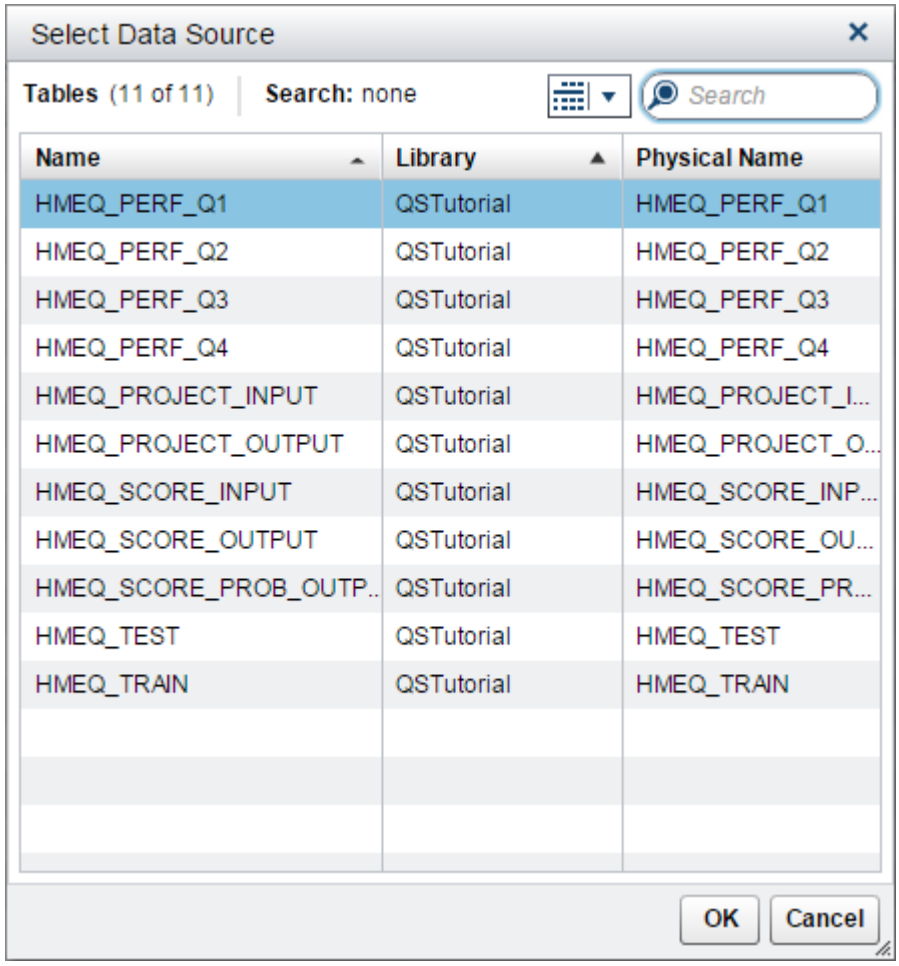
See Also

"Overview of Projects" on page 126

Import Project Variables

To import project variables:

1. Select **Variables** ⇨ **Input** and click . The Select Data Source window appears.



2. Select HMEQ_PROJECT_INPUT as the data source from the **QSTutorial** library. Click **OK**.
3. Select the **Output** tab and click .
4. Select the HMEQ_PROJECT_OUTPUT as the data source from the **QSTutorial** library and click **OK**.
5. Click to make the changes effective for other pages.
6. Click **Yes** in the warning message since you have not set a champion or challenger model yet.

See Also

"Import Variables" on page 134

Set the Project Properties

To define the properties that SAS Decision Manager uses to create reports, score, publish, and monitor models:

1. Select **Properties** ⇨ **Specific**.
2. Click **Browse** to select the default data tables from the **QSTutorial** library and specify values for the other properties:

   Default test table
   select **HMEQ_TEST**.

   Default scoring input table
   select **HMEQ_SCORE_INPUT**.

   Default scoring output table
   select **HMEQ_SCORE_OUTPUT**.

   Default train table
   select **HMEQ_TRAIN**.

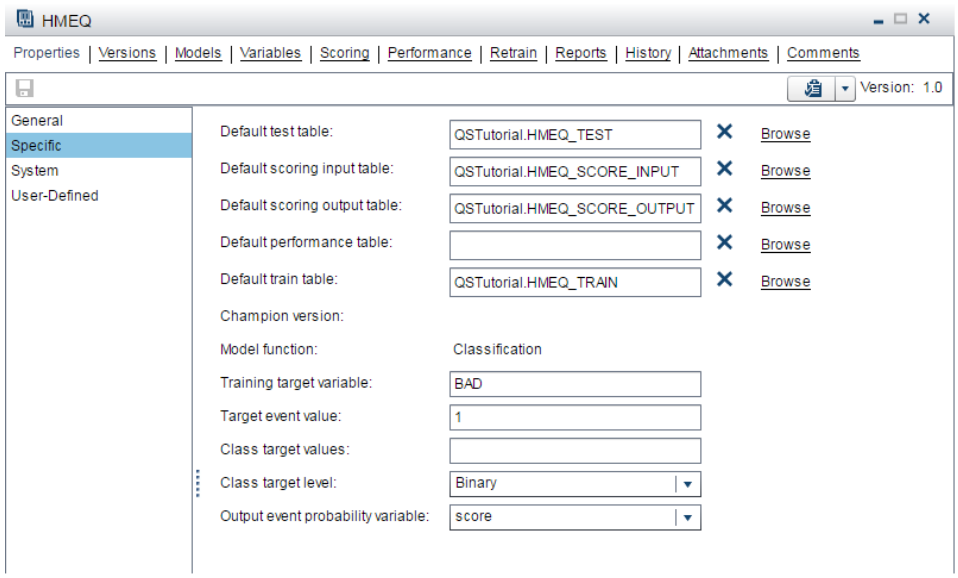
   Training target variable
   enter **BAD**.

   Target event value
   enter 1.

   Class target level
   select **Binary**.

   Output event probability variable
   select **score**.
3. Click .

Here is an example of the **HMEQ** project-specific properties:



See Also

"Project Properties" on page 129

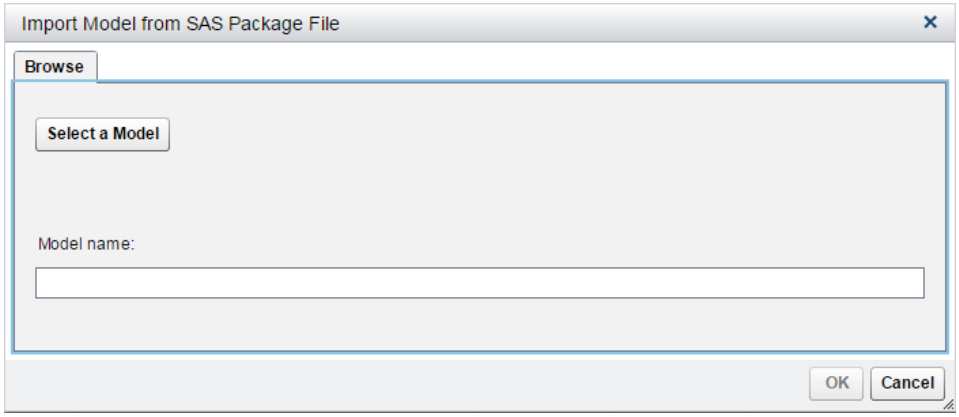
Import Models

Import a SAS Package File

Note: Before you import a model, verify that the model type matches the **Model function** property setting on the project's **Properties** page.

To import a model from a SAS Package File:

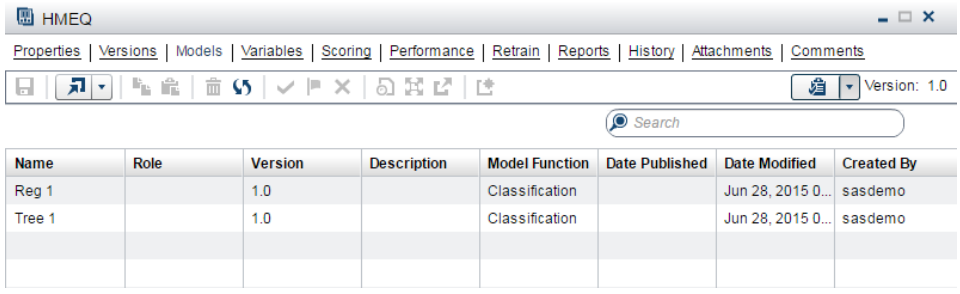
1. Select the **Models** page.
2. Click and select **from a SAS package file**.



3. On the **Browse** tab, click **Select a Model** and navigate to the location of the file (for example, use`<drive:>\QuickStartTutorial\Models\Reg1)`.

   Select the miningResult.spk file to import and click **Open**.
4. Enter **Reg 1** for the name of the model.
5. Click **OK**.
6. Click **Close**.
7. Repeat steps 2 through 6 to import the model package file located in `<drive:>\QuickStartTutorial\Models\Tree1`. Name the model **Tree 1**.

Here is an example of the **Models** page, after the models have been imported:



Name	Role	Version	Description	Model Function	Date Published	Date Modified	Created By
Reg 1		1.0		Classification		Jun 28, 2015 0...	sasdemo
Tree 1		1.0		Classification		Jun 28, 2015 0...	sasdemo

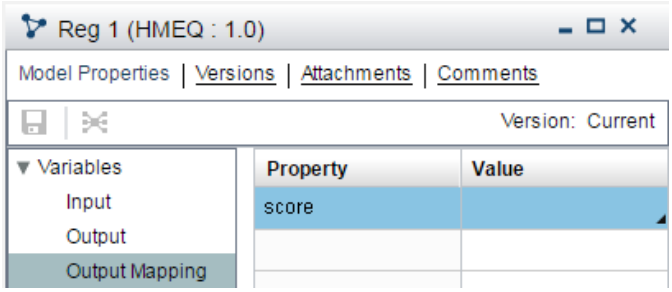
See Also

"Overview of Importing Models" on page 145

Map Model Variables to Project Variables

To map model variables to the project variables:

1. Select and open the **Reg 1** model.
2. On the **Model Properties** page, select **Variables** ⇨ **Output Mapping**.



3. Select **EM_EVENTPROBABILITY** from the **Value** column beside the **score** variable in the **Property** column.
4. Click .
5. Repeat steps 1 through 4 for the **Tree 1** model.

See Also

"Map Model Variables to Project Variables" on page 164

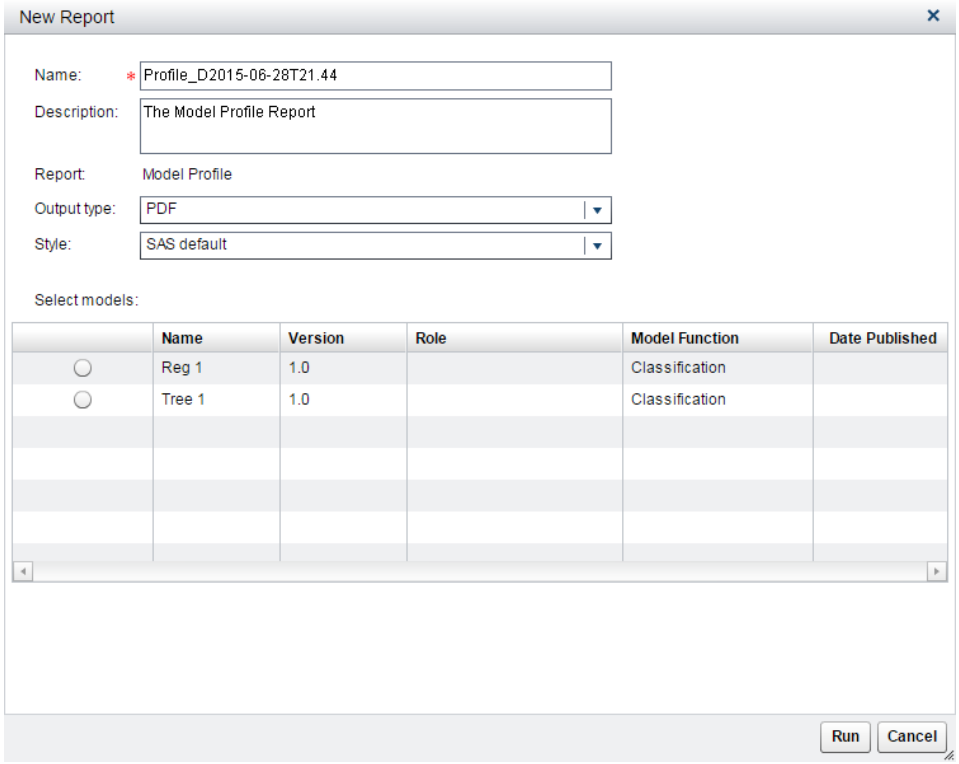
Create Model Comparison Reports

Create a Model Profile Report

The Model Profile report creates three tables to display the profile data that is associated with the model input variables, output variables, and target variables.

To create a Model Profile report:

1. Select the **Reports** page.
2. Click and select **Model Profile**. The New Report window appears.



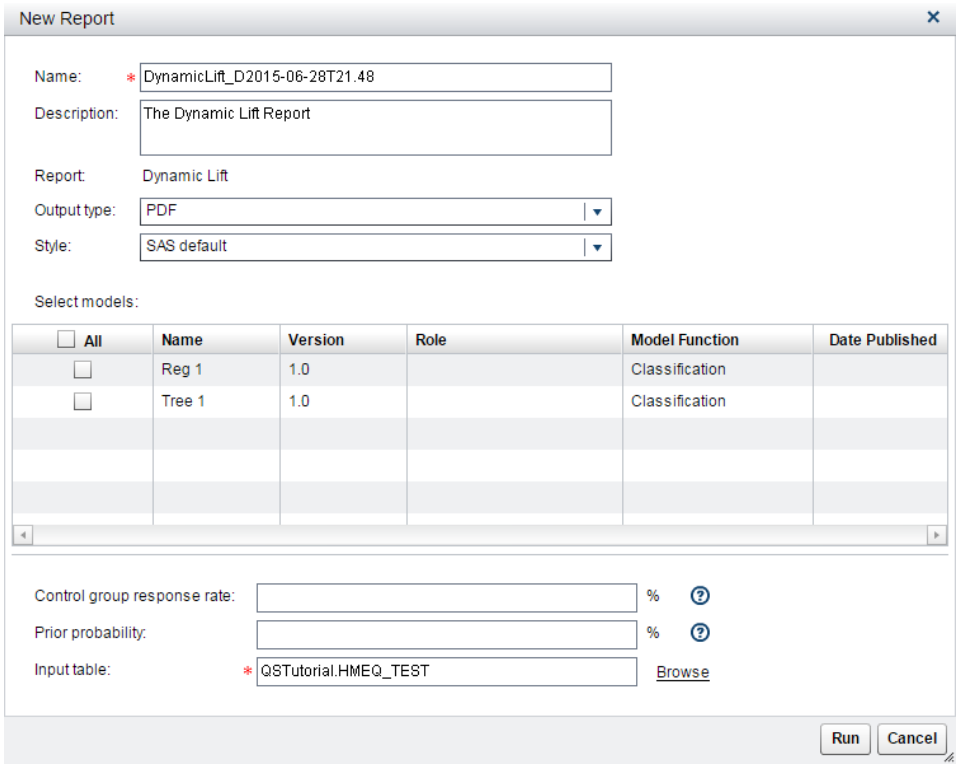
3. Enter **profile_tree1** for the name of the report.
4. Select **PDF** for the output type.
5. Select **Seaside** for the style of the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
6. Select the model **Tree 1** from the list.
7. Click **Run**. The report is generated and appears in the default viewer based on the selected output type.

Create a Dynamic Lift Report

The Dynamic Lift report provides visual summaries of the performance of one or more models for predicting a binary outcome variable.

To create a Dynamic Lift report:

1. Click and select **Dynamic Lift**. The New Report window appears.



2. Enter **lift_reg1tree1** for the report name.
3. Select **HTML** for the output type.
4. Select **Seaside** for the style of the report.
5. Select the models **Reg 1** and **Tree 1** from the list.
6. (Optional) Specify the **Control group response rate**.
7. (Optional) Specify the **Prior probability**.
8. Accept the default input table of **QSTutorial.HMEQ_TEST**.
9. Click **Run**. The report is generated and appears in the default viewer based on the selected output type.

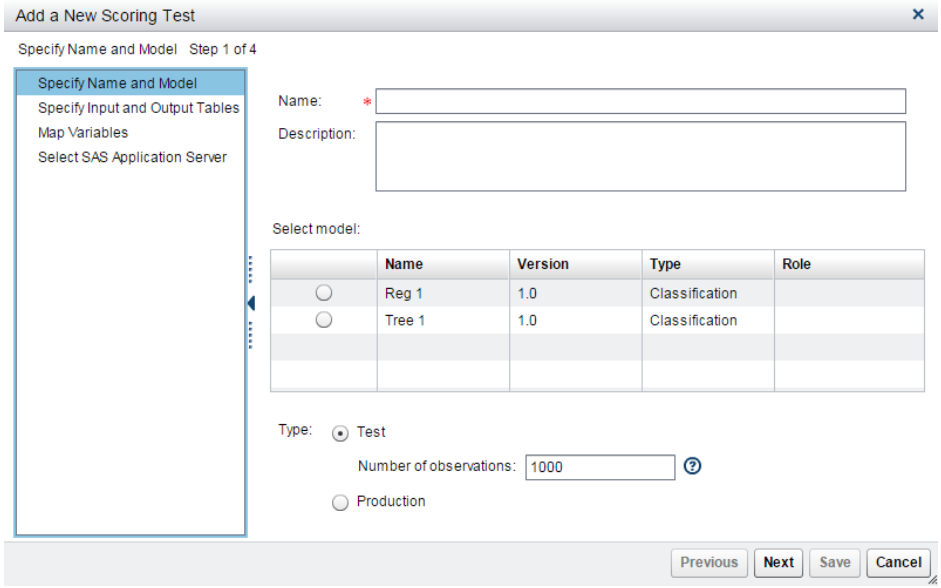
See Also

"Overview of Model Comparison, Validation, and Summary Reports" on page 202

Create a Scoring Test

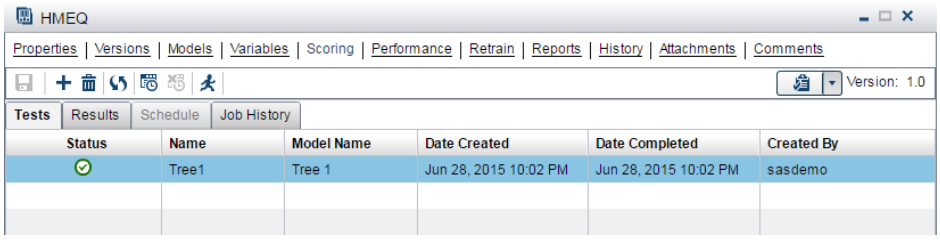
To create a scoring test:

1. Select the **Scoring** page.
2. Click +. The Add a New Scoring Test window appears.



3. Enter **Tree1** for the name.
4. (Optional) Enter **test1** for the description.
5. Select the **Tree 1** model from the list.
6. Select **Test** for the type of scoring test. Accept the default number of observations (1000 rows) to be read from the scoring input table.

   Note: A best practice is to select **Test** before beginning all scoring tests. Later, when you are satisfied with the results of running the scoring test and you are ready to put the test into production, you can change the type to **Production**.
7. Click **Next**.
8. Verify that the value you previously specified for the **Default score input table** project property appears in the **Input table** box. To select a table, click **Browse** and select the table **QSTutorial.HMEQ_SCORE_INPUT**. Click **OK**.
9. Verify that the value you previously specified for the **Default score output table** project property appears in the **Output table** box. To select a table, click **Browse** and select the table **QSTutorial.HMEQ_SCORE_OUTPUT**. Click **OK**.
10. Click **Next**.
11. Verify that all of the scoring output table variables are mapped to the available variables.
12. Click **Next**.
13. Select the configured SAS Application Server (for example, **SASApp**).
14. Click **Save**.
15. Select the **Tree1** scoring test from the list and click .



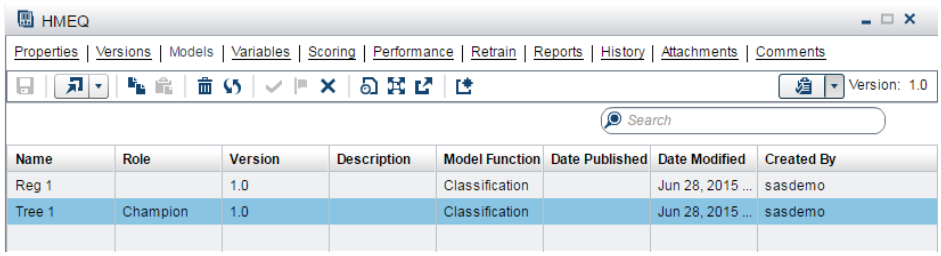
16. Click the **Results** tab or double-click the scoring test to view the scoring test results. When the job has completed without errors, a appears in the **Status** column.

    Note: You can check the status of a job by clicking in the **Job History** tab. A new record appears after the job has completed.

Set the Champion Model

To set the champion model:

1. Select the **Models** page of the project.
2. Select the **Tree 1** model and click . The value in the **Role** column changes to **Champion**.



See Also

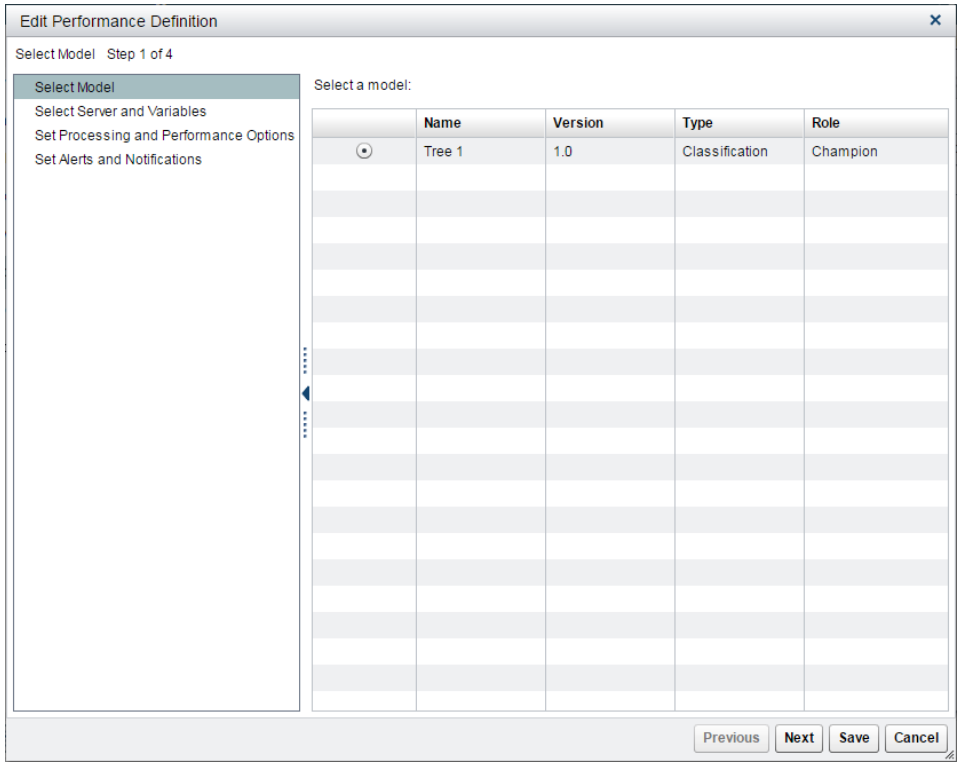
"Overview of Scoring Tests" on page 193

Monitor the Performance of a Champion Model

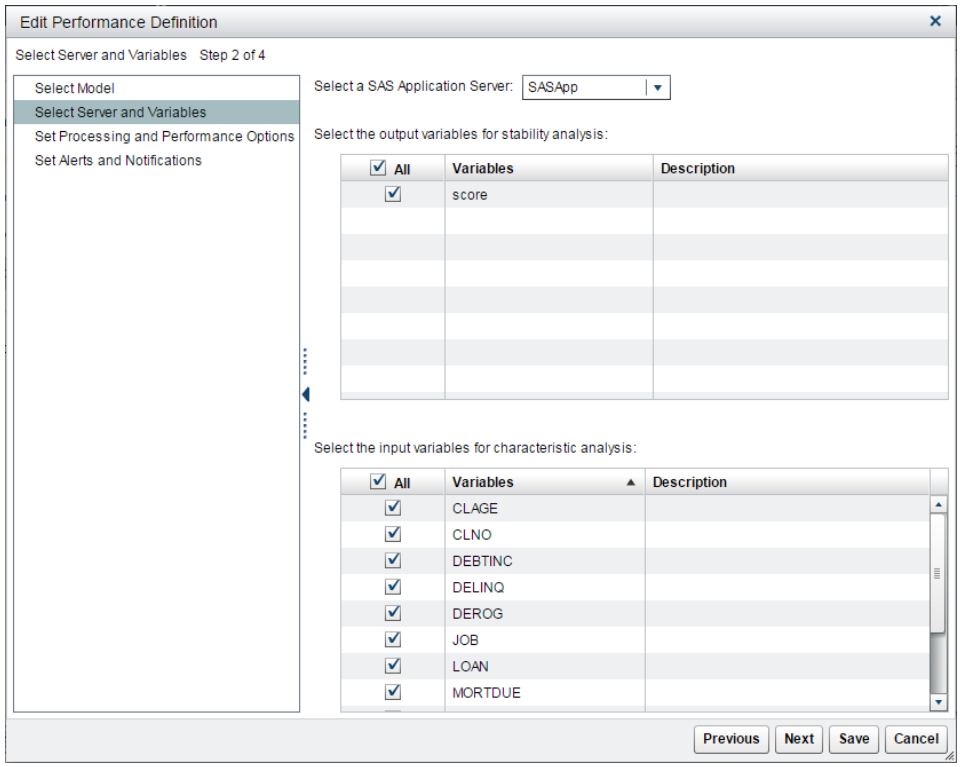
In this tutorial you use the Edit Performance Definition wizard to generate SAS code. You then execute the generated code.

To edit the performance definition:

1. Select the **Performance** page.
2. Click and select the champion model **Tree 1**. Click **Next**.



3. Select a SAS Application Server (for example, the default is SASApp).
4. Click **All** to select all output variables for stability analysis.
5. Click **All** to select all input variables for characteristic analysis.



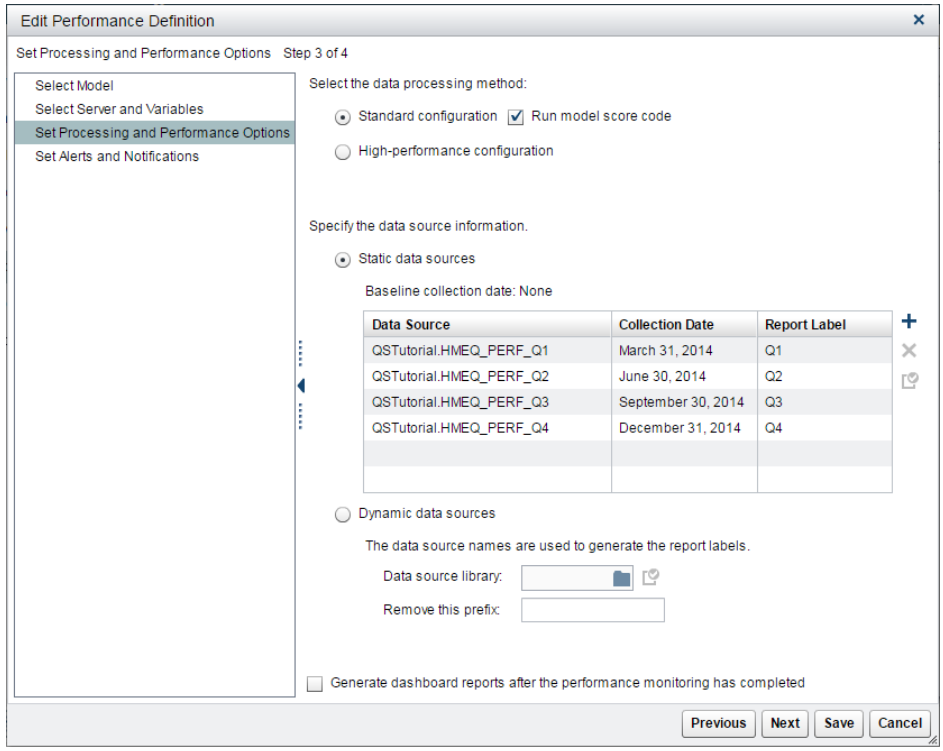
Click **Next**.

6. Select **Standard configuration** as the data processing method and select **Run model score code** to run the score code in the performance monitor job.
7. Specify the data source information.
   a. Select **Static data sources**.
   b. Click +.

      Note: The data table whose collection date is the earliest is set as the baseline performance data table.
   c. Click the empty cell in the **Data Source** column.
   d. Click **Browse** and select the **HMEQ_PERF_Q1** performance data source from the **QSTutorial** library. Click **OK**.
   e. Click the empty cell in the **Collection Date** column and click . Select the date of **March 31, 2014**. The date can be any date in the time period when the performance data was collected.
   f. Enter the label **Q1** in the **Report Label** column. The report label represents the time point of the performance data source. Because the report label appears in the performance charts, use a label that has not been used for another time period, is short, and is understandable.

      Note: If you duplicate report labels, the results in the previous performance results are overwritten.
   g. (Optional) Select a data source and click to verify that the selected input variables and target variable are included in the performance data source.
   h. (Optional) Repeat the above steps to add the following performance data sources from the **QSTutorial** library to the performance definition.

Data Source	Collection Date	Report Label
HMEQ_PERF_Q2	June 30, 2014	Q2
HMEQ_PERF_Q3	September 30, 2014	Q3
HMEQ_PERF_Q4	December 31, 2014	Q4



i. (Optional) To delete a data source from the performance definition, select the data source and click ✕.

8. Click **Next**.

   Note: When prompted to replace the existing performance data, click **Yes**.

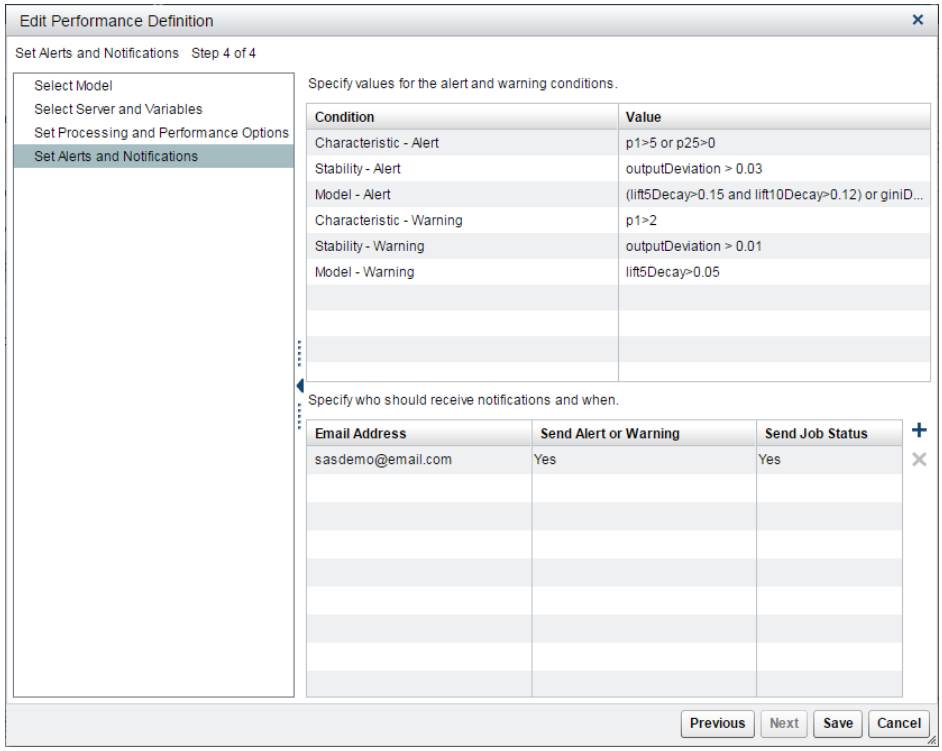
9. (Optional) Either specify values for the alert and warning conditions or accept the defaults. Click **Next**.

10. (Optional) To send the results by email, click +. A new row is added to the table.

    a. Enter an email address.

    b. Select either **Yes** or **No** if you want an alert or warning to be sent by email when alert or warning thresholds have been exceeded.

    c. Select either **Yes** or **No** if you want a completion notice with the job status to be sent by email every time the report runs.



11. Click **Save**.
12. Click .
13. After the performance monitoring has been completed, a confirmation message appears. Click **Close**.
14. Click the **Results** tab to view the performance results.

    Note: You can check the status of a job by clicking in the **Job History** tab. A new record appears after the job has completed.

See Also

"Edit and Execute a Performance Definition" on page 259

Publish a Champion Model to the SAS Metadata Repository

To publish a model to the SAS Metadata Repository:

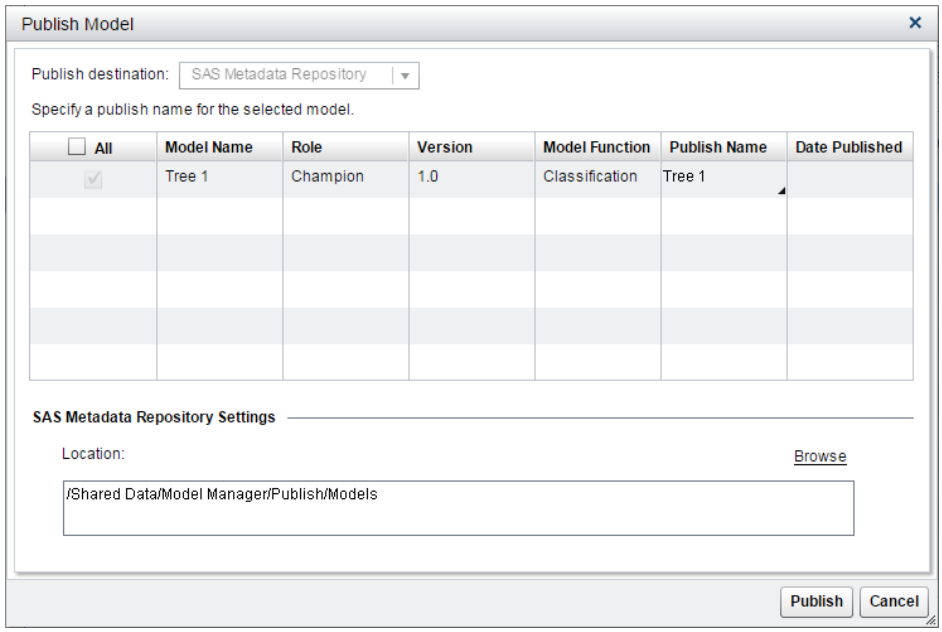
1. Select the **Models** page.
2. Select the champion model **Tree 1** and click .

   Note: Alternatively, you can select a project from the Projects category view and click .

3. To specify a new publish name, click in the **Publish Name** column for a selected model.

Note: By default, the publish name for the champion model is the project name. You cannot modify the publish name for a champion model when publishing from the Projects category view.

4. Click **Browse** to select the location to publish the model to. You must have Write permission to this location.



5. Click **Publish**.
6. Click **Close**.
7. (Optional) Select **History** ⇨ **Published** to view a list of the models that have been published.

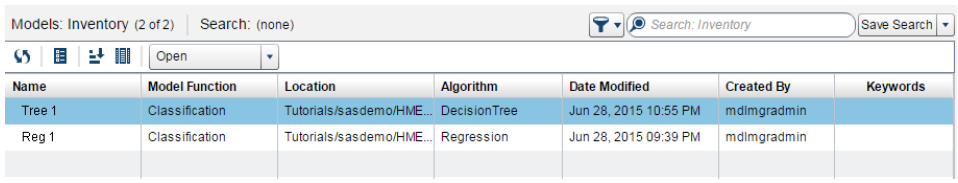
See Also

"Publishing Models to the SAS Metadata Repository" on page 293

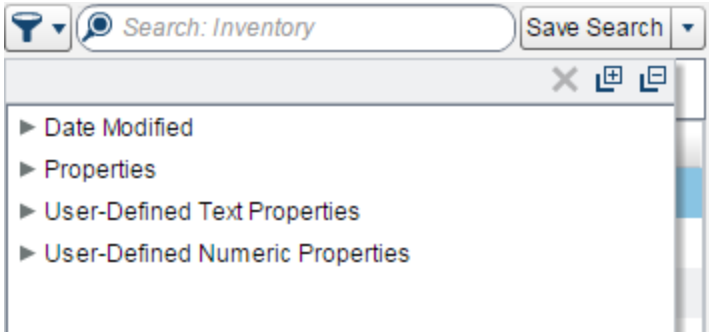
View and Search Model Inventory

In this tutorial, you view the model inventory that you imported in the previous tutorials. You then search the model inventory to filter the results, and open a model.

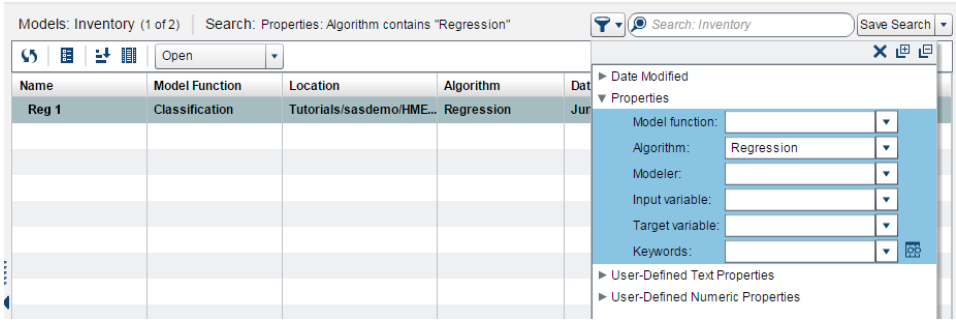
1. Select **Models** ⇨ **Inventory**.



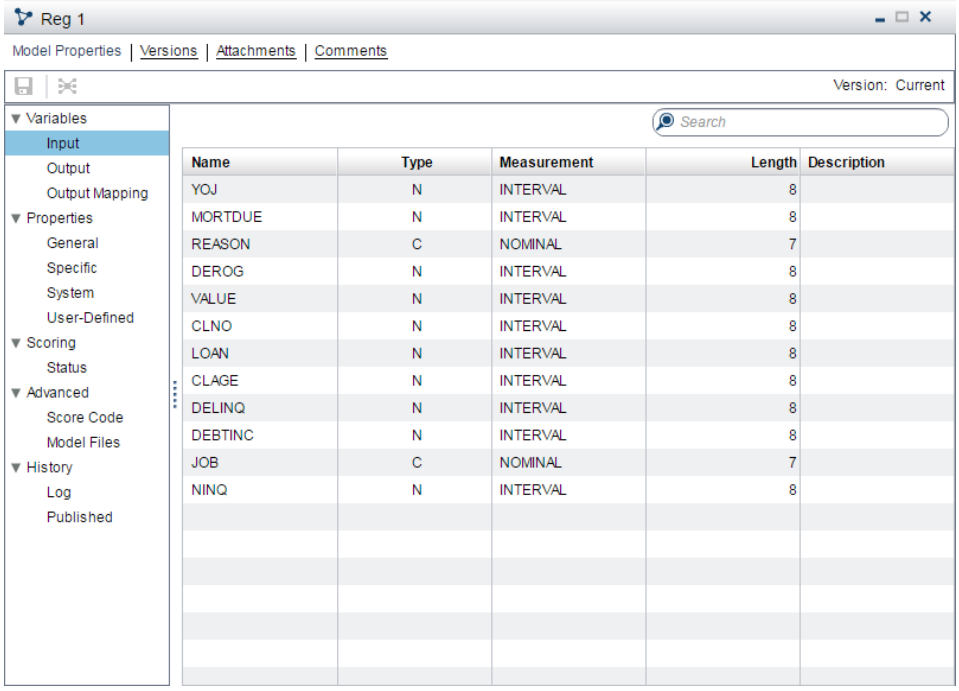
2. Click . The additional search options appear.



3. Expand the **Properties** search option.
4. Select **Regression** from the **Algorithm** property drop-down list. The model inventory list is filtered automatically.



5. Double-click the **Reg 1** model to view its model content.



6. Minimize or close the model object window to return to the list of models.

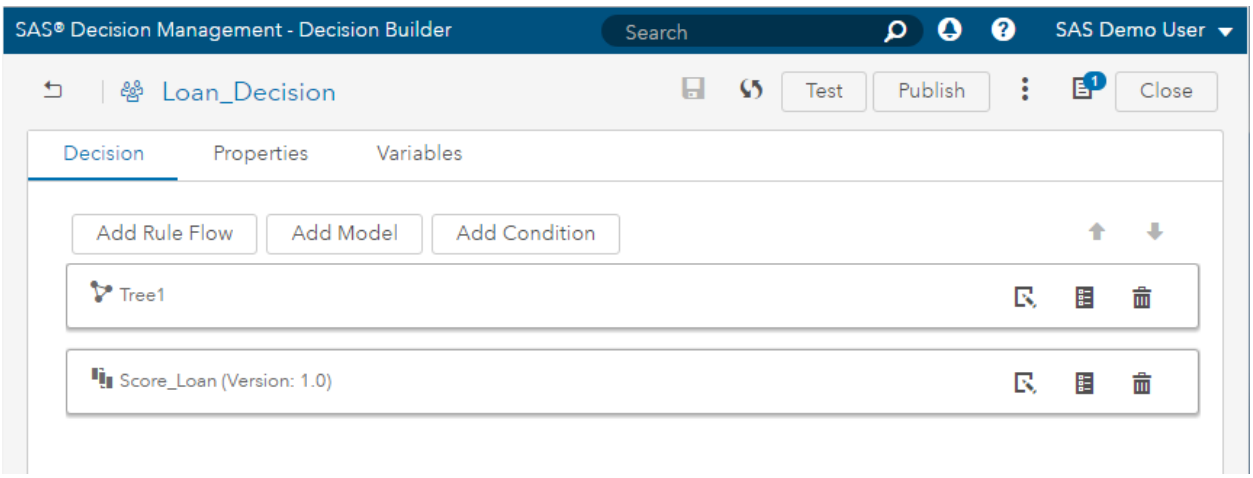
Create and Publish a Decision

After you have published models and rule flows, you can use the Decision Builder to combine these models and rule flows into decisions. In this step, you combine the Score_Loan rule flow and the Tree1 champion model into a decision.

1. Open Decision Builder. The default URL is http://*host_name*:*port*/SASDecisionManager/DecisionBuilder.

   Note: To run Decision Builder in Internet Explorer, you must use Internet Explorer 11.

2. Click . The New Decision window appears.
3. Enter **`Loan_Decision`** for the decision name.
4. Click , select the **HMEQ_Vocab** vocabulary, click **Open**, then click **Save**. The application opens the new decision.
5. Click **Add Rule Flow**. The Open window appears.
6. Select the **Score_Loan** rule flow, and click **Open**. The Select Rule Flow Version window appears.
7. Select **1.0** for the version, and click **OK**.
8. Click **Add Model**. The Open window appears.
9. Select the **Champion - Tree 1** model, and click **Open**.
10. Select the **Variables** page.
11. In the **Input** column, clear the check box for the DEROG variable.
12. Click to save the decision.
13. Click **Publish**. The Publish Decision window appears.
14. Select **Use a Decision Logic object in batch applications**.
15. Verify that **Publish a new object** is selected.
16. Select the location where you want to publish the decision, and click **OK**.
17. Click **Publish** in the Publish Decision window.



Deploy the Decision in SAS Data Integration Studio

When you publish decisions as Decision Logic objects, they are available for use by other applications such as SAS Data Integration Studio. Because decisions incorporate both rule flows and models, the process of adding rule flows and models to SAS Data Integration Studio jobs is simplified. You can add decisions to the job instead of individual models and rule flows.

In this tutorial, you create a SAS Data Integration Studio job that uses the decision that you created in "Create and Publish a Decision" on page 44.

To create and run a job in SAS Data Integration Studio:

Create a new job.

1. Right-click on the folder where you want to create the new job, and select **New** ⇨ **Job**. The New Job window appears.

   This tutorial creates the new job in **My Folder**.

2. Enter `ScoreLoanDecision` for the job name, and click **OK**.
3. If you are prompted to select a default application server, select **SASApp**, and click **OK**.

Add the data source for the decision.

4. On the **Folders** tab, expand **Shared Data** ⇨ **Model Manager** ⇨ **QSTutorial**.
5. Drag the **HMEQ_SCORE_PROB_OUTPUT** data source to the Job Editor window.

Add the decision to the job.

6. On the **Folders** tab, navigate to the location where you published the **Loan_Decision** decision.
7. Drag the **Loan_Decision** decision to the Job Editor window.
8. Drag the cursor from the input table to the decision.

Add a table loader for the decision results table.

9. On the **Transformations** tab, expand **Access**.
10. Drag the **Table Loader** to the Job Editor window.

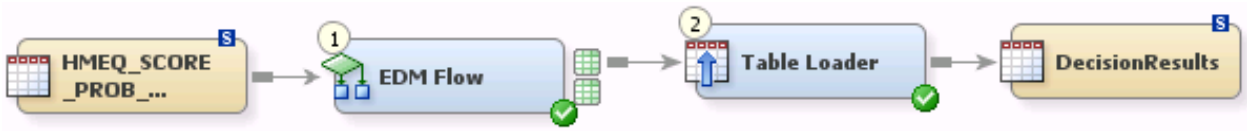
Create a decision results table.

11. On the **Folders** tab, right-click on **My Folder**, and select **New** ⇨ **Table**. The New Table wizard opens.
12. Enter `DecisionResults` for the table name, and click **Next**. The Table Storage window appears.
13. Select **QSTutorial** for the **Library**, and click **Next**. (Use the default values for the remaining fields.) The Select Columns window appears.
14. Select the **Inventory** tab.

15. Expand **Library**, expand **QSTutorial**, select **HMEQ_SCORE_PROB_OUTPUT**, and click ▷.
16. Click **Next**. The Change Columns/Indexes window appears.
17. Click **Next**, and then click **Finish**.
18. Drag the **DecisionResults** table to the Job Editor window.

Connect the output table and run the decision.

19. Drag the cursor from the EDM Results output port of the decision to the table loader. To determine which port is the correct port, position your mouse over the port icons to display the description of the port.
20. Drag the cursor from the table loader to the **DecisionResults** output table.



21. Click **Run** to run the job.

Chapter 3

Managing Data Tables

About Managing Data Tables

The Data category enables you to manage your list of data tables from within SAS Decision Manager. You can create new Base SAS libraries, add and remove tables, view table data and metadata, create and delete table summaries, and associate attachments and comments with tables. The application uses these data tables whenever it needs to access data, such as for importing terms, rule discovery, rule flow testing, decision testing, and for testing, scoring, retraining, and performance monitoring of models.

You can view the list of tables by selecting **Data** ⇨ **Tables**. There are three ways to add tables to the list.

- You can use SAS Visual Data Builder to create new tables and add them to the list. See "Adding Tables Using SAS Visual Data Builder" on page 48 for more information.
- If the table is already registered in the SAS Metadata Repository, you can add the table to the list as described in "Add Tables That Are Registered in Metadata" on page 48.

- If the table is not already registered in the SAS Metadata Repository, you can add a new table as described in "Register and Add New Tables" on page 49.

Note:

- SAS Decision Manager cannot access tables in a SAS LASR Analytic Server instance.
- If you do not have the appropriate permissions to access a folder, then the tables and libraries are not listed in the Data category view.
- You can view tables with double-byte characters (DBCS) in the table name or in variable names, and you can use them to create summary tables. You can use them in models, model projects, and model portfolios, as well as their associated tasks. You cannot use them in business rules or decisions.To use these tables, support for DBCS characters must be enabled as described in "Extend Support for Double-Byte Characters in Table and Variable Names" in *SAS Decision Manager: Administrator's Guide*.
- Special characters and blanks are not supported in table names or in variable names.

Adding Tables Using SAS Visual Data Builder

SAS Visual Data Builder enables analysts and data administrators to perform data preparation for analytics. You can design queries to perform joins, add calculated columns, and subset and sort data. Several productivity features speed the creation of columns based on common aggregation functions.

Once you design your queries, you can reuse them as subqueries for more sophisticated queries, export them as jobs for scheduling, or schedule them directly from the user interface.

The application has data import features that enable you to access data from spreadsheets, delimited files, and SAS data sets. Once you import the data, you can prepare it for analysis or join it with existing data.

The application provides a series of features that you can use to extract and transform data from multiple sources and create new data tables.

To access SAS Visual Data Builder, select **Data** ⇨ **Tables**. Click to start SAS Visual Data Builder. For more information about SAS Visual Data Builder, click to access *SAS Visual Analytics: User's Guide* and videos about using SAS Visual Data Builder.

Add Tables That Are Registered in Metadata

If a data table has already been registered in the SAS Metadata Repository, you can add it to the list of data sources. To add one or more tables:

1. Select **Data** ⇨ **Tables**.
2. Click + and select **Add Registered Tables**. The Choose an Item window appears.
3. Select the tables that you want to add, and click **OK**.

Register and Add New Tables

You can create new Base SAS libraries and register tables by using SAS Decision Manager. To register new tables in the SAS Metadata Repository and add them to the list of data sources:

1. Select **Data** ⇨ **Tables**.
2. Click + and select **Register Tables**. The Register Tables window appears.

   Note: You cannot use the **Register Tables** option to add a table that is already registered. You must select **Add Registered Tables** instead. See "Add Tables That Are Registered in Metadata" on page 48.

3. Select an existing library, or create a new Base SAS library.

   To use an existing library:

   a. Select **Use an existing library**.

   b. Click and select the library.

   c. Click **Next**.

   To create a new Base SAS library:

   a. Select **Create a new library**.

   b. Specify a name for the new library. The name cannot exceed 60 characters.

   c. (Optional) Specify a description for the library.

   d. Specify a libref. A *libref* is a name that SAS uses to refer to the library. Enter a unique name of eight characters or less.

   e. Select the location for the new library. This location is the folder in the SAS Metadata Repository where the library is stored.

   f. Select the server and the directory where the data tables reside.

   g. Click **Next**.

   Note: If you click **Cancel** at this point, a folder for the library is created in the SAS Metadata Repository, but the folder does not appear in the list of data tables.

4. Select the tables that you want to add to the library, and click to add the tables to the **Selected tables** list. Click to add all of the tables to the **Selected tables** list.
5. Click **Finish**.

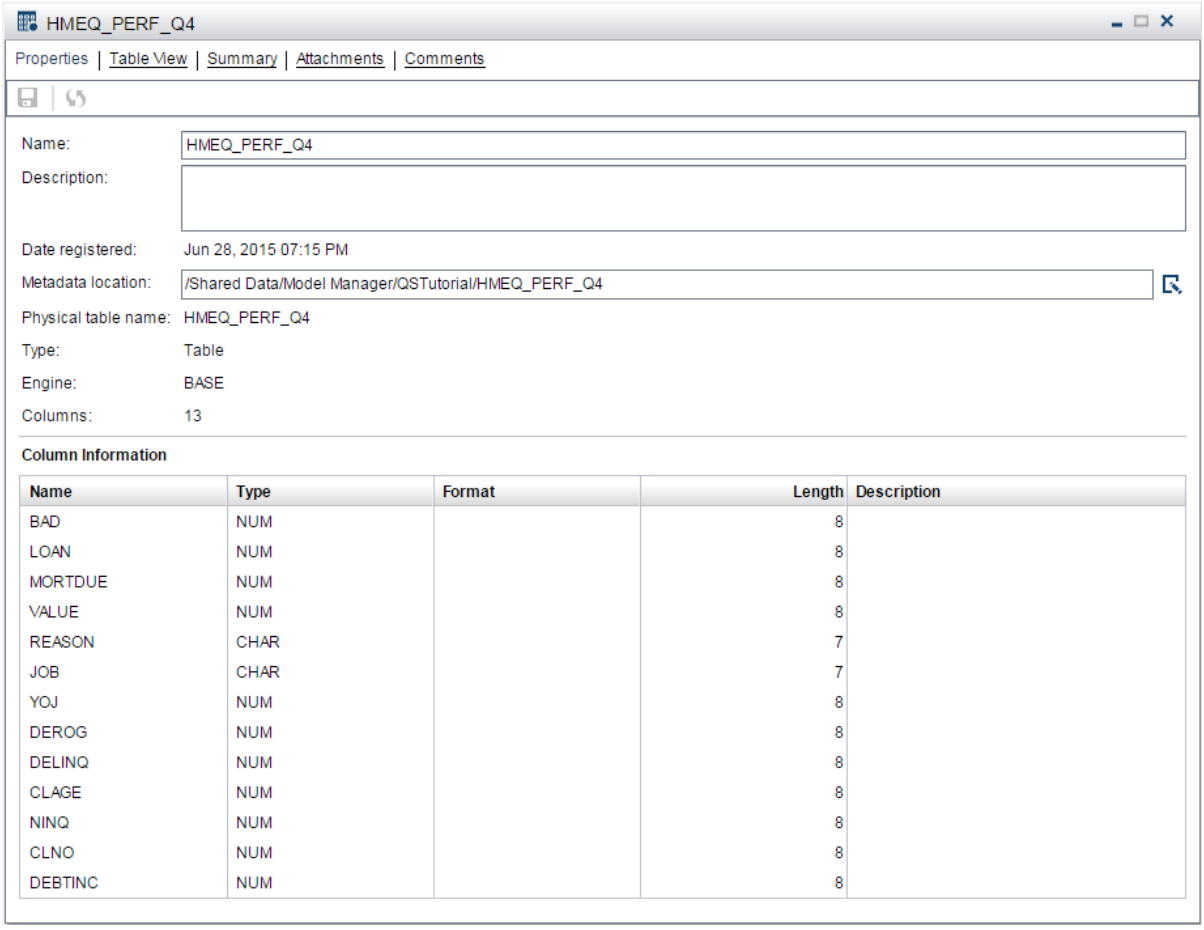
Edit Table Properties and View Table Metadata

The **Properties** page displays table metadata. On this page, you can edit the data source name and description, and change the table associated with the data source name.

1. Select **Data** ⇨ **Tables**.

2. Double-click on the table whose properties you want to edit. The **Properties** page appears.

   The **Properties** page displays table metadata such as the number of columns, the table location, and information about each column in the table.



Name	Type	Format	Length	Description
BAD	NUM		8	
LOAN	NUM		8	
MORTDUE	NUM		8	
VALUE	NUM		8	
REASON	CHAR		7	
JOB	CHAR		7	
YOJ	NUM		8	
DEROG	NUM		8	
DELINQ	NUM		8	
CLAGE	NUM		8	
NINQ	NUM		8	
CLNO	NUM		8	
DEBTINC	NUM		8	

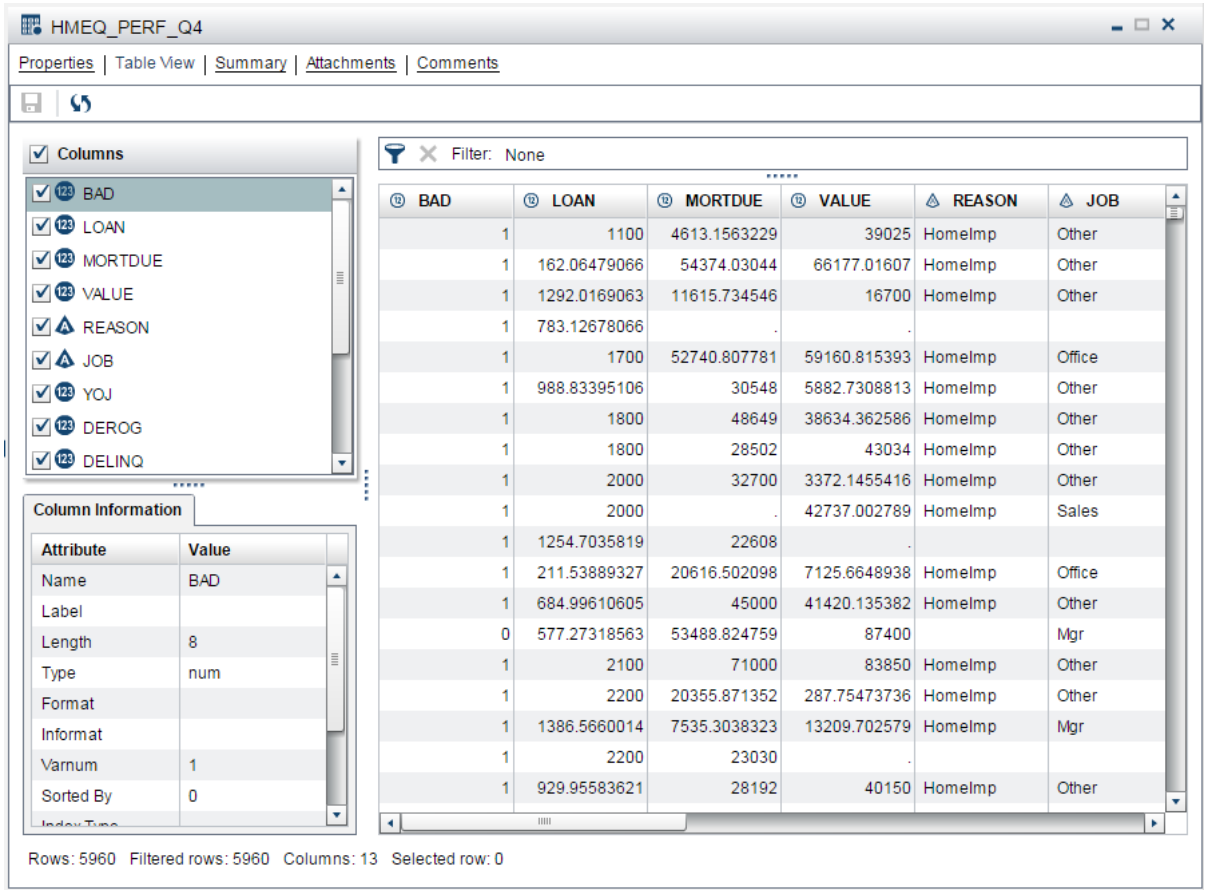
3. Edit the data source name and description, or click to select a different table as the data source.
4. Click to save the changes.

View Table Data

To view table data:

1. Select **Data** ⇨ **Tables**.
2. Double-click on the table that you want to view.
3. Select the **Table View** page.

On the **Table View** page, you can control the display by selecting specific columns in the **Columns** section. The **Column Information** section displays information about the currently selected column.



Note: If the name of the selected column begins with a blank space, the table cannot be displayed.

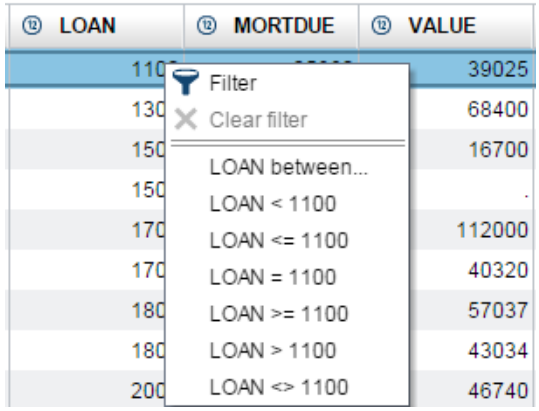
Note: The row count might not be displayed, depending on the database with which the table was created.

To sort the table based on the values in a particular column, click on the column heading. If the column is sorted in ascending order, a ▲ appears beside the column heading. When the column is sorted in descending order, a ▼ appears.

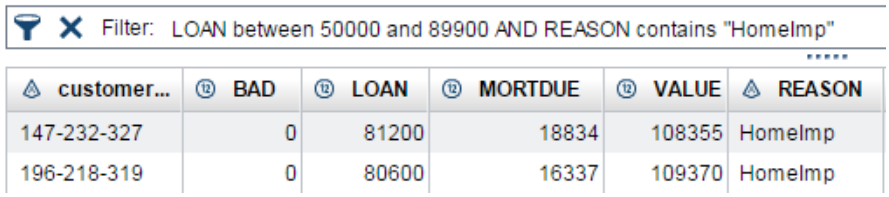
Filter Data in the Table View

You can filter the rows that are shown on the **Table View** page in either of the following ways:

- Click above the table. The Filter window appears. Enter a valid SQL expression, and click **Apply**.
- Right-click on a value in the table. SAS Decision Manager displays several predefined filter options. You can select any of these options. Depending on which option you select, you might be prompted to enter data values for the query.



The expression that you enter is displayed above the data table, and the table is filtered accordingly.



To clear the filter and display the entire table, click ✕.

For more information about SQL expressions, see *SAS FedSQL Language Reference*.

Create a Table Summary

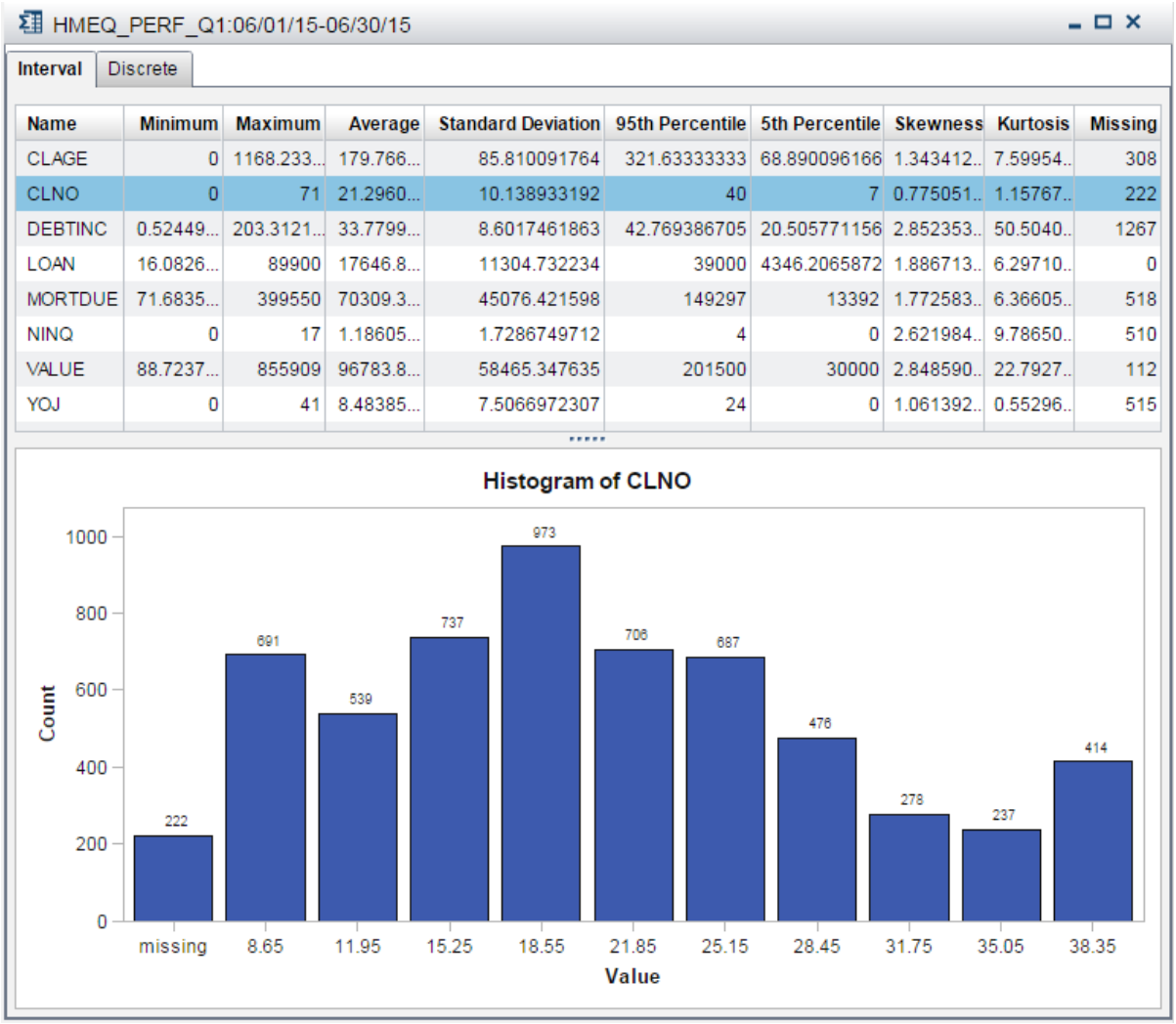
To create a new table summary:

1. Select **Data** ⇨ **Tables**.
2. Double-click on the table for which you want to add a summary.
3. Select the **Summary** page.
4. Click +.
5. In the New Summary window, select the **Collection period** and the specific date or time values for the collection period that is represented by the data in the table.

   Note: The **Collection period** is not used to filter the data.
6. (Optional) Specify a summary description.
7. Click **Run**. SAS Decision Manager runs a process to summarize the data and adds the new summary to the **Summary** page.

Double-click on the summary to open it.

The following display shows the **Summary** page for the HMEQ_PERF_Q1 table. The collection period represented by the data in the table is June 2015.



Name	Minimum	Maximum	Average	Standard Deviation	95th Percentile	5th Percentile	Skewness	Kurtosis	Missing
CLAGE	0	1168.233...	179.766...	85.810091764	321.63333333	68.890096166	1.343412..	7.59954..	308
CLNO	0	71	21.2960...	10.138933192	40	7	0.775051..	1.15767..	222
DEBTINC	0.52449...	203.3121...	33.7799...	8.6017461863	42.769386705	20.505771156	2.852353..	50.5040..	1267
LOAN	16.0826...	89900	17646.8...	11304.732234	39000	4346.2065872	1.886713..	6.29710..	0
MORTDUE	71.6835...	399550	70309.3...	45076.421598	149297	13392	1.772583..	6.36605..	518
NINQ	0	17	1.18605...	1.7286749712	4	0	2.621984..	9.78650..	510
VALUE	88.7237...	855909	96783.8...	58465.347635	201500	30000	2.848590..	22.7927..	112
YOJ	0	41	8.48385...	7.5066972307	24	0	1.061392..	0.55296..	515

Delete a Table Summary

To delete a table summary:

1. Select **Data** ⇨ **Tables**.
2. Double-click on the table whose summary you want to delete.
3. Select the **Summary** page.
4. Select the summary that you want to delete.
5. Click 🗑.

Add Attachments to a Table

To add an attachment such as a document file or an image file:

1. Select the **Attachments** page.
2. Click +, and select the attachment file.

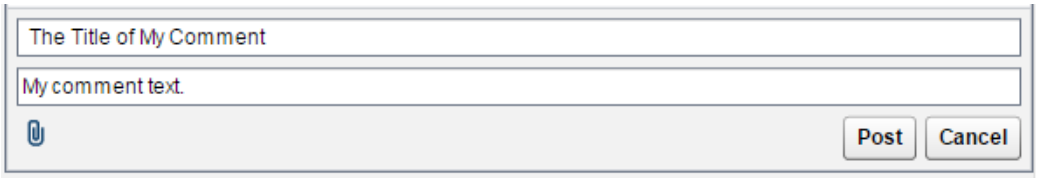
   Note: You cannot attach executable files to rule sets, rule flows, or data tables. Depending on your site's settings, you might be able to attach executable files to models, projects, and portfolios. For more information, see "Configure Support for Executable Attachments" in *SAS Decision Manager: Administrator's Guide*.
3. Click .

Note: You can delete an attachment by selecting the attachment and clicking .

Add Comments to a Table

You can add new comments or reply to existing comments. To add a new comment:

1. Select the **Comments** page.
2. Enter a topic title and enter the comment. The topic title is required, and the field for comments does not appear until you enter the topic title.



3. (Optional) Click to add an attachment such as an image or a document.

   Note: You cannot attach executable files to a comment.
4. Click **Post**.

To reply to an existing comment, enter your reply in the field immediately below the topic title for the existing comment, and click **Post**.

Click to see comments that have been posted by others.

To search for text in the comments, enter text in the search field at the top of the **Comments** page.

Remove a Table

Removing a table from the list of data sources does not delete the table from file system. To remove a table from the list of data sources:

1. Select **Data** ⇨ **Tables**.

2. Select the table that you want to remove from the list.
3. Click .

Part 3

Working with Business Rules

Chapter 4

Managing Business Rule Folders

About Business Rules Folders

Before you create any vocabularies, lookup tables, rule sets, or rule flows, you need to create business rules folders. Content that is related to business rules is stored in business rules folders.

All folders appear for each category in SAS Decision Manager. When you open a folder, only the items in the selected category appear.

Create New Top-Level Folders

Note: If folder administration is enabled for your site, the ability to create top-level folders is limited to folder administrators. See the description of the **brm.folder.config.enabled** configuration property in "Business Rules Manager Web Advanced Properties" in *SAS Decision Manager: Administrator's Guide* for more information.

If you are a folder administrator, you must specify the permissions for the folder, and you can specify locations for test information. To create a new top-level folder:

1. Select any category under **Business Rules**.
2. Click , and select **New Top-level Folder**.
3. In the New Folder window, enter the name of the new folder. Folder names are limited to 100 characters.
4. (Optional) Enter a description for the new folder. Descriptions are limited to 256 characters.

5. If you are a folder administrator, follow these steps:
   a. Select the group that you want to have access to the new folder.
   b. (Optional) Select the folder that contains test data for rule flows in the new folder.
   c. (Optional) Select the library where you want the tests saved for rule flows in the new folder.
6. Click **OK**.

Create New Folders

To create a new folder within another folder:

1. Select the parent folder in which you want to create a new subfolder.
2. Click , and select **New Folder**.
3. In the New Folder window, enter the name of the new folder. Folder names are limited to 100 characters.
4. (Optional) Enter a description for the new folder. Descriptions are limited to 256 characters.
5. Click **OK**.

Move Folders

Note: If folder configuration is enabled, you cannot move a top-level folder into another folder.

You cannot move a folder up to the level of a top-level folder. A folder cannot be moved into one of its own folders. To move folders:

1. Select the folders that you want to move, and then right-click and select **Move**. The Choose a Location window appears.
2. Select a new location for the folders, and click **Move**.

Delete Folders

A folder must be empty before you can delete it. To delete folders, select the folders that you want to delete, and then click .

Chapter 5

Managing Vocabularies

Introduction to Vocabularies, Entities, and Terms

Vocabularies, entities, and terms are the basic building blocks of a SAS Decision Manager database. Vocabularies contain entities, and entities contain terms.

Object	Description
Vocabulary	Vocabularies contain one or more business entities. Vocabularies categorize and structure the entities and terms needed to create a rules database.
Entity	An *entity* is an object in a business domain. For example, an entity could be Customer, Transaction, or Account. Entities contain terms. They group terms into logical units. Entities are not mapped to tables or to table columns when rules flows are published.

Object	Description
Term	A term is an attribute of an entity. For example, a customer entity might have terms such as name, address, and income. A transaction entity might contain terms for date, time, transaction amount, and account number. Terms are the objects with which you build business rules. A business rule can have *condition terms* and *action terms*. Suppose your rule is **`if balance>1000 then account="premium"`**. The term **`balance`** is a condition term, and **`account`** is an action term. Terms are mapped to table columns by the applications that use published rule flows that are within metadata.

Tips for Creating Entities and Terms

- Before you define vocabulary entities and terms, review the structure of the tables that input values will come from. Vocabularies should be structured similarly to these tables to ensure that terms are mapped correctly to input columns. Coordinate your work with the groups that will use the vocabulary. Coordination helps ensure that the vocabulary structure meets their requirements.
- Boolean data can be represented with terms that are defined either as Boolean data types or as Character data types. In some cases, Boolean values might be better represented by using terms defined as Character. For example, if your data already uses **`yes`** and **`no`** for Boolean data, then you probably want to use a Character term to process these values rather than try to translate those values to **`true`** and **`false`**.

Create a Vocabulary

To create a new vocabulary:

1. Select **Business Rules** ⇨ **Vocabularies**.
2. Right-click on the folder where you want to create the new vocabulary, and select **New Vocabulary**. Alternatively, you can select the folder, click , and then select **New Vocabulary**. The New Vocabulary window appears.
3. Enter the name of the new vocabulary. Vocabulary names can contain up to 32 characters and must be unique within a folder. Vocabulary names are case insensitive. For example, SAS Decision Manager considers **`name`** to be equal to **`NAME`**.
4. (Optional) Enter a description for the new vocabulary. Descriptions are limited to 256 characters.
5. Click **OK**.

Create an Entity

To create a new entity from the Vocabulary category view:

1. Right-click on the vocabulary where you want to create the new entity, and select **New Entity**. Alternatively, you can select the vocabulary, click , and then select **New Entity**. The New Entity window appears.

   Note: If you open a vocabulary, you can click + to create a new entity.

2. Enter the name of the new entity. Entity names can contain up to 32 characters and must be unique within a vocabulary. Entity names are case insensitive. For example, SAS Decision Manager considers **`name`** to be equal to **`NAME`**.

3. (Optional) Enter a description for the new entity. Descriptions are limited to 256 characters.

4. Click **OK**.

Note: When you create a new decision in Decision Builder, Decision Builder creates an entity named DecisionBuilderCreatedTerms. This entity contains terms that do not exist in the vocabulary specified for the decision.

Create a Term

To create a new term:

1. Right-click on the entity where you want to create the new term, and select **New Term**. Alternatively, you can select the entity, click , and then select **New Term**. The New Term window appears.

2. Enter the name of the new term. Term names can contain up to 32 characters and must be unique within a vocabulary. Term names are case insensitive. For example, SAS Decision Manager considers **`name`** to be equal to **`NAME`**.

   Note: Do not use any of these operators or keywords as term names: AND, OR, IN, NOT, LIKE, TRUE, or FALSE. Do not use **`_N_`** or any DS2 reserved word as a term name. See "Reserved Words in the DS2 Language" in *SAS DS2 Programmer's Guide*.

3. (Optional) Enter a description for the new term. Descriptions are limited to 256 characters.

4. Select the data type for the new term.

5. Select the domain type for the new term.

6. (Optional) Specify the domain values for the new term. Domain values are the set of expected values for a term. See "Specify Domain Values" on page 65 for more information.

7. (Optional) Select **Exclude from input** if you do not want the term to be mapped to a column in an input data set. (The application expects all terms to be mapped to columns in an input data set.)

8. (Optional) Select **Exclude from output** to exclude a term from the output data sets that are generated by rule flows.

   TIP To create a temporary term for use only while a rule flow is executing, select both **Exclude from input** and **Exclude from output**.

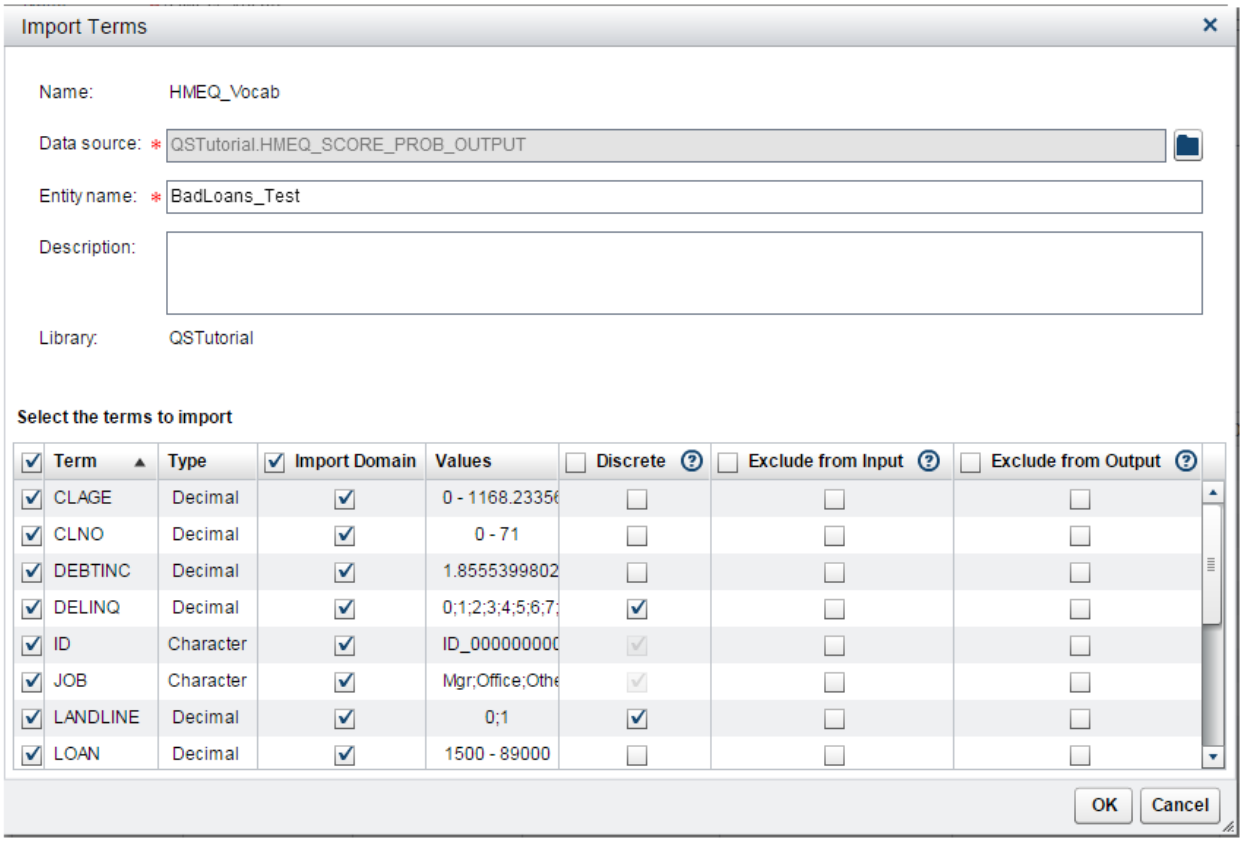
9. Click **OK**.

Import Terms from a Data Source

To import terms from a data source:

1. Create a new vocabulary or open an existing vocabulary. For information, see "Create a Vocabulary" on page 62.
2. Click . The Import Terms window appears.
3. Select the data source from which you want to import terms. SAS Decision Manager displays the terms, their type, and domain information.
4. Enter the entity name where you want to add the terms.
5. (Optional) Enter a description for the entity.
6. Select the terms and domain values that you want to import.

   TIP To select all of the items in a column, select the box beside the column heading.



7. Select the **Discrete** box for terms that have a discrete set of values. For more information, see "Specify Domain Values" on page 65.
8. All of the imported terms are included in both the input and the output unless you exclude them. Select any terms that you want to exclude. See Step 7 and Step 8 of "Create a Term" on page 63 for more information.
9. Click **OK**.

Specify Domain Values

Domain values are the set of expected values for a term. Domain values are not used to validate rules. They are used to enable faster and easier rule authoring. They are displayed in the Equation Editor, which enables you to add a value to an expression by double-clicking on the value.

Domain values can include term or variable names. For continuous values, you can use the greater than (>), less than (<), and equal (=) signs to set limits for ranges. You cannot include a semi-colon (;) within a domain value. You do not need to enclose Character values in quotation marks unless the value itself contains an apostrophe (').

Separate individual domain values with a semi-colon (;).

Note: To enter continuous Date and Datetime values, enclose the values in single quotation marks, followed by a **d** or **dt** as shown in the following table.

The following table shows examples of domain values.

Table 5.1 *Examples of Domain Values*

Data Type	Domain Type	Examples
Character	Discrete	`high risk;low risk` `"d'oscail";"d'fhill"`
Integer	Discrete	`0;800;3500`
	Continuous	`>100;<=myterm`
Decimal	Discrete	`3.14;12.98`
	Continuous	`>1.25;<=N1`
Date	Discrete	`01jul2012;31jul2012`
	Continuous	`>='01jan2013'd;<='31dec2013'd`
Datetime	Discrete	`01jul2012:10:52:00;31jul2012:23:00:00`
	Continuous	`<='01jul2012:00:00:00'dt;>'31jul2012:23:00:00'dt`

Data Type	Domain Type	Examples
Boolean	By default, Boolean values are set to **`True`** and **`False`** and cannot be changed.	

Edit Existing Vocabularies, Entities, or Terms

To edit existing items:

1. Select the items that you want to edit, and click **Open** in the toolbar.
2. Edit the item properties as needed.
3. Click .

Delete Vocabularies, Entities, or Terms

Note: You cannot delete a vocabulary, entity, or term if it is used in a rule set or in a decision.

Note: You cannot delete entities named DecisionBuilderCreatedTerms because the terms in that entity are used in a decision.

You can delete multiple vocabularies or terms at the same time, but all of the items that you are deleting must be of the same type. You can delete only one entity at a time.

Select the items that you want to delete, and click .

Move Vocabularies, Entities, Terms

Note: You cannot move a vocabulary, entity, or term if it is used in a rule set or in a decision.

You can move multiple vocabularies or terms at the same time, but all of the items that you are moving must be of the same type. You can move only one entity at a time.

To move items:

1. Select the items that you want to move, and then right-click and select **Move**. The Choose a Location window appears.
2. Select a new location for the folders, and click **Move**.

When you move multiple terms and terms with the same name already exist in that destination folder, SAS Decision Manager also appends an underscore, a number, and **`Move`** to the term names.

Duplicate Vocabularies, Entities, or Terms

You might want to duplicate a vocabulary when, for example, your company starts a new initiative similar to a previous one, and you need a similar vocabulary to work with. Duplicating terms is useful when you have the same actions that must be performed by different entities.

To duplicate a single vocabulary, entity, or term, right-click the item that you want to duplicate and select **Duplicate**. The window that appears contains the same fields that are displayed when you define a new vocabulary, entity, or item. For more information, see:

- "Create a Vocabulary" on page 62
- "Create an Entity" on page 63
- "Create a Term" on page 63

You can duplicate multiple vocabularies or terms at the same time. You cannot duplicate multiple entities at the same time. To duplicate multiple vocabularies or terms:

1. Select the items that you want to duplicate. All of the items must be of the same type.
2. Right-click and select **Duplicate**. The Choose a Location window appears.
3. Select the location for the duplicate items, and click **Duplicate**.

   When you duplicate multiple vocabularies or multiple terms and save the duplicates in a different folder than the original, SAS Decision Manager appends an underscore, a number, and `Move` to the names.

Search for Rule Sets by Term

To find all of the rule sets that use a specific term:

1. Select **Business Rules** ⇨ **Vocabularies**.
2. Select the term that you want to search for and click . The Search for Rule Sets window appears.

   TIP You do not have to select a term before you click . You can click , and manually enter the term that you want to search for.

3. Select the usage that you want SAS Decision Manager to search for.

   Anywhere
   finds terms that are used as condition terms, action terms, or in expressions

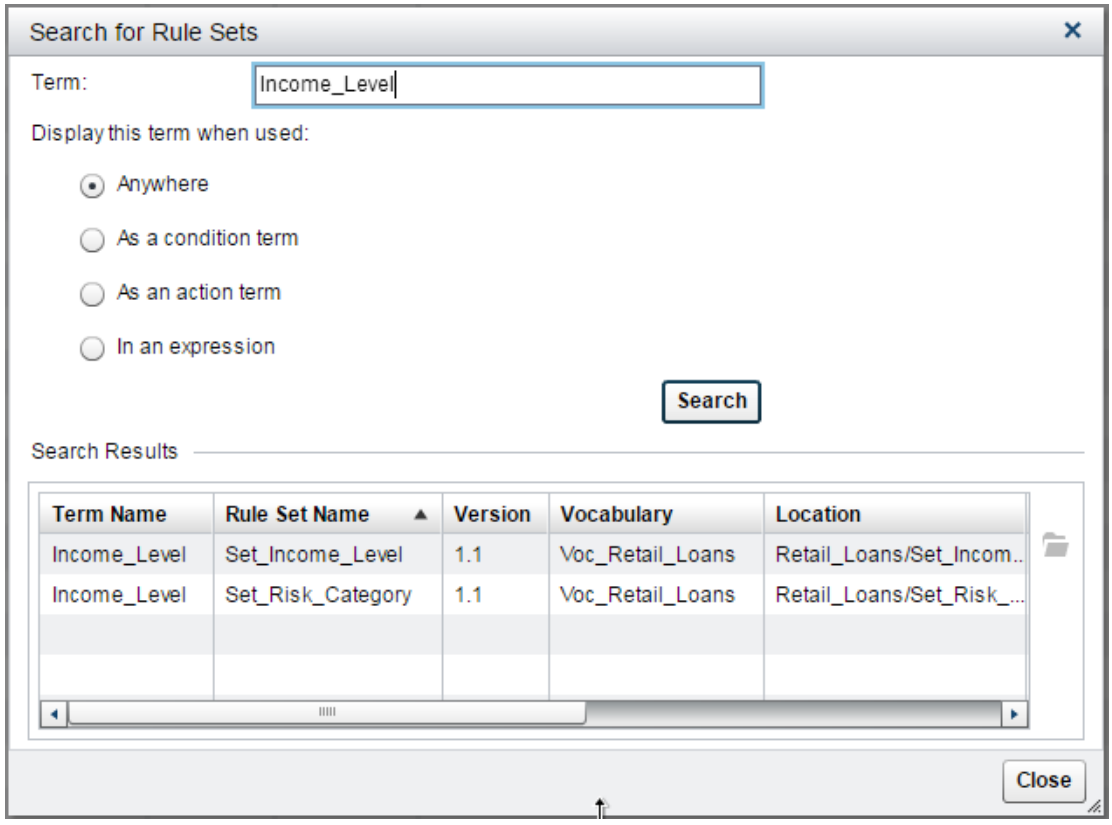
   As a condition term
   finds terms that are used only as condition terms (terms that have been added to the column or row headings of the decision table)

   As an action term
   finds terms that are used only as action terms (terms that have been added to the column or row headings of the decision table)

In an expression
finds terms only when they are used in rule expressions

4. Click **Search**. If the search returns results, SAS Decision Manager displays all of the rule sets and versions in which it found the term.

To open one of the rule sets, select the rule set in the search results list, and click .



Note: If a term is referenced implicitly in an expression, then selecting **In an expression** and searching for that term does not find it. For example, if you enter the expression `+10` into the rule set editor for the term `myterm`, then the resulting expression for **myterm** is `=myterm+10`. Because the expression that you entered into the rule set editor did not explicitly reference `myterm`, the search was not successful.

Chapter 6

Using Lookup Tables and Functions

About Lookup Tables and Functions

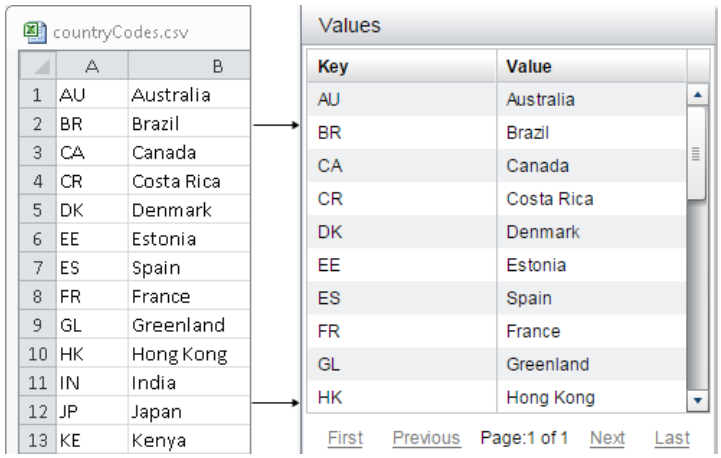
SAS Decision Manager provides the ability to import lookup tables and reference them from rules. Lookup tables are tables of key-value pairs. For example, you can use a lookup table to retrieve a part name based on the part code number or to retrieve the full name for a country based on its abbreviation.

You can import lookup data from comma-separated-values (CSV) files such as those created by Microsoft Excel into lookup tables in SAS Decision Manager. You can re-import updated CSV files as needed to refresh the lookup tables.

Note: SAS Decision Manager does not support CSV files that contain signature lines.

Note: You can configure the character that is used as a separator in CSV files that are imported through the SAS Decision Manager interface. See Step 4 of "Review Application Properties in SAS Management Console" in *SAS Decision Manager: Administrator's Guide* for more information.

Figure 6.1 *CSV File Imported Into SAS Decision Manager*



In a lookup table, each *lookup key* is associated with a *lookup value*. Lookup keys must be unique within each lookup table. Character strings in lookup tables are limited to 512 characters.

SAS Decision Manager provides two functions, LOOKUP and LOOKUPVALUE, that enable you to determine whether a lookup key exists in a lookup table and to retrieve a lookup value from a lookup table.

Create a New Lookup Table

You create a new lookup table by importing a CSV file.

To create a new lookup table:

1. Select the **Lookups** category.
2. Right-click on the folder where you want to create the new lookup table, and select **New Lookup Table**. The New Lookup Table window appears.
3. Enter a name for the new lookup table. Names are limited to 32 characters and can contain only alphanumeric characters and underscores. Lookup table names must be unique within the SAS Decision Manager database. Lookup table names are case insensitive. For example, SAS Decision Manager considers **`NAME`** to be equal to **`name`**.
4. (Optional) Enter a description for the new lookup table. Descriptions are limited to 256 characters.
5. Click , and select the CSV file that contains the lookup data.
6. Click **OK**.

Refresh a Lookup Table

To refresh a lookup table:

1. Right-click on the lookup table that you want to refresh, and select **Refresh Lookup Table**. The Refresh Lookup Table window appears.
2. Click , and select the CSV file that contains the lookup data.
3. Click **OK**.

Note: You can also refresh a lookup table by using the %BRM_IMPORT_LOOKUP macro. See "%BRM_IMPORT_LOOKUP" in *SAS Decision Manager: Administrator's Guide* for more information.

Delete Lookup Tables

Note: You cannot delete a lookup table if it is referenced in a rule.

To delete lookup tables, select the tables that you want to delete, and click .

Duplicate Lookup Tables

To duplicate a single lookup table:

1. Right-click on the lookup table, and select **Duplicate**. The Duplicate window appears.
2. Enter a name for the duplicate lookup table. Names are limited to 32 characters and can contain only alphanumeric characters and underscores. Lookup table names must be unique within the SAS Decision Manager database. Lookup table names are case insensitive. For example, SAS Decision Manager considers **`NAME`** to be equal to **`name`**.
3. (Optional) Enter a description for the duplicate table. Descriptions are limited to 256 characters.
4. Click , and select the folder where you want to save the duplicate lookup table.
5. Click **OK**.

   If you do not enter a new name for the duplicate table, SAS Decision Manager appends an underscore and a number to the duplicate table name.

To duplicate multiple lookup tables:

1. Select the tables that you want to duplicate.
2. Right-click and select **Duplicate**. The Choose a Location window appears.
3. Select the location where you want the save the duplicate tables, and click **Duplicate**.

SAS Decision Manager appends an underscore, a number, and **Move** to the duplicate table names.

Move Lookup Tables

You cannot move a lookup table if it is open. To move lookup tables:

1. Select the tables that you want to move, and then right-click and select **Move**. The Choose a Location window appears.
2. Select a new location for the tables, and click **Move**.

Dictionary

LOOKUP Function

Determines whether a lookup key exists in a lookup table. Specify the LOOKUP function as the expression for the term whose value contains the lookup key that you want to search for.

Restrictions: You can specify the LOOKUP function in condition expressions only.

If an expression contains the LOOKUP function, then the expression cannot contain anything else.

Returned data type: Boolean

Syntax

LOOKUP ('*lookup_table_name*')

Required Argument

lookup_table_name
specifies the name of the lookup table that you want to search.

Example

Suppose you have a Country_Codes lookup table that uses two-letter abbreviations for countries as the lookup key and country names as the lookup values.

Values

Key	Value
AU	Australia
BR	Brazil
CA	Canada
CR	Costa Rica

To verify that the value of the term Ctry_Key exists as a lookup key in the table Country_Codes, enter the LOOKUP function as the expression for the Ctry_Key term. For example, if the Ctry_Key column in the current input record contains the value **CA**, then the following expression evaluates to True.

Condition Term

Ctry_Key
Lookup('Country_Codes')

LOOKUPVALUE Function

Retrieves a lookup value from a lookup table.

Restrictions: You can specify the LOOKUPVALUE function in action expressions only.

If an expression contains the LOOKUPVALUE function, then the expression cannot contain anything else.

Returned data type: Lookup tables are stored as character data. However, you can assign the results of the LOOKUPVALUE function to a Character, an Integer, a Decimal, a Date, a Datetime, or a Boolean action term. The LOOKUPVALUE function converts the results to match the type of the action term.

Syntax

LOOKUPVALUE ('*lookup_table_name*', *term_or_string*)

Required Arguments

lookup_table_name
specifies the name of the lookup table that you want to search.

term_or_string
specifies the lookup key for the value that you want to retrieve. Enclose character strings in quotation marks.

Example

Suppose you have a Country_Codes lookup table that uses two-letter abbreviations for countries as the lookup key and country names as the lookup values. The Country_Codes lookup table contains the lookup key **CA**, and the lookup value that corresponds to that key is **Canada**.

Values

Key	Value
AU	Australia
BR	Brazil
CA	Canada
CR	Costa Rica

If the Ctry_Key column in the current input record contains the value **CA**, you can retrieve the lookup value that is associated with that key from the table Country_Codes with the following expression:

```
LOOKUPVALUE('Country_Codes',Ctry_Key)
```

If this expression is entered for the Country_Name action term, the expression assigns the value **Canada** to the term Country_Name.

Action Term

Country_Name
LookupValue('Country_Codes', Ctry_Key)

Chapter 7

Managing Rules and Rule Sets

About Rules, Rule Sets, and Expressions

A rule specifies conditions to be evaluated and actions to be taken if those conditions are satisfied. Rules are grouped together into rule sets. Rule sets are logical collections of rules that are grouped together because of interactions or dependencies between the rules or because they are processed together when they are published.

Most rules correspond to this form:

```
if condition_expressions then action_expressions
```

For example, suppose you have the following rule:

```
if customer_debt > customer_assets then app_status = "Decline"
```

In this case, **customer_debt** is a condition term, and **customer_debt > customer_assets** is a condition expression. The term **app_status** is an action term, and **app_status = "Decline"** is an action expression. To enter this rule in the rule set editor, you first need to add the terms **customer_debt** and **app_status** to the rule set editor, and then enter the expressions under the terms to which the expressions apply.

The following figure shows the rule set editor with this rule added to it:

		Condition Term	Action Term
#		customer_debt	app_status
1	If	>customer_assets	='Decline'

Condition expressions are not required. Rules with only action expressions are always executed.

A single rule can have multiple terms, conditions, and actions. Multiple condition expressions within the same rule are joined together with the AND operand. For example, suppose you define the following rule in SAS Decision Manager:

		Condition Term		Action Term	
#		Credit_Score	Homeowner	Approved	Interest_Rate
1	If	>700	=True	=True	=4.5

SAS Decision Manager generates the following rule condition:

```
(Credit_Score > 700) AND (Homeowner = True)
```

It generates the following assignments:

```
Approved=True
Interest_Rate=4.5
```

Action expressions are always assignment statements.

How Rules Are Evaluated and When Rule-Fired Records Are Generated

By default, the condition expressions for all rules in a rule set are evaluated sequentially regardless of the results of previous rules. However, you can use the ELSE and OR operators to control whether the condition expression for a rule is evaluated. See "Controlling Which Conditions Are Evaluated" on page 82 for more information.

If a rule's condition expression evaluates to True, SAS Decision Manager executes the rule's action expression.

If a rule's condition evaluates to True, the rule is said to have *fired*.

By default, every time a rule fires, it generates a rule-fired record. However, you can control when rule-fired records are generated by using the **Record rule-fired data** check box. See Step 7 of "Add a New Rule" on page 78.

Note: A rule that does not have a condition expression does not generate a rule-fired record.

Create a New Rule Set

To create a new rule set:

1. Select **Business Rules** ⇨ **Rule Sets**.
2. Right-click on the folder where you want to create the new rule set, and select **New Rule Set**. The New Rule Set window appears.
3. Enter a name for the new rule set. Rule set names are limited to 100 characters and must be unique within a folder.
4. (Optional) Enter a description for the new rule set. Descriptions are limited to 256 characters.
5. Select the vocabulary that is associated with the new rule set.

   Note: If a rule set has a locked version, you cannot change the vocabulary associated with the latest version of the rule set.
6. Click **Create**. SAS Decision Manager opens the new rule set and displays the **Properties** page.

Defining New Rules in the Rule Set

Add a New Rule

To add a new rule:

1. Open the rule set where you want to add the new rule.
2. Select the **Rules** page. The default view on the **Rules** page is the horizontal view. See "Selecting the Rule Set Editor View" on page 78 for information about the vertical and list views
3. Click +.
4. Add any additional terms that the new rule requires, and add expressions for the new rule in the table cells in the rule editor. See "Define Expressions for a Rule" on page 80 for more information.

   Note: You can add up to 200 condition terms and 200 action terms to a rule set. A greater number of terms might affect performance.
5. (Optional) Change the order of the new rule. The rule order, in addition to the IF, ELSE, or OR keyword (see "Controlling Which Conditions Are Evaluated" on page 82), controls how rules are evaluated within the rule set.

   TIP You can also change the order of the rules later by right-clicking on a rule and selecting either **Reorder** or **Swap**. See "Change the Order of Rules in a Rule Set" on page 92 for more information.
6. (Optional) Modify the name of the new rule on the **Rule Details** tab. Rule names are limited to 100 characters and must be unique within a rule set.
7. (Optional) Clear the **Record rule-fired data** check box on the **Rule Details** tab if you do not want a rule-fired record to be written each time this rule fires.
8. (Optional) Enter a description for the new rule on the **Rule Details** tab.
9. Click .

Selecting the Rule Set Editor View

There are three views available in the rule set editor: horizontal, vertical, and list. The default view is the horizontal view.

In the horizontal view, the terms used by the rules are displayed across the top of the rule set editor, and there is one row for each rule. You can return to the horizontal view from the other views by clicking .

		Condition Term			Action Term
#		HAS_LANDLINE	LOAN_SCORE	REASON	BadLoanFlag
1	If		>.5		True
2	If	=1	>.6 AND <=.7	'HomeImp'	True
3	If	=1	>.7 AND <=.8	'DebtCon'	False
+		Click here to add a new rule.			

To switch to the vertical view, click . In the vertical view, the terms used by the rules are displayed in the left column, and there is one column for each rule.

Condition Term	1	2	3	+
	If	If	If	
HAS_LANDLINE		=1	=1	*Click here to add a new rule.*
LOAN_SCORE	>.5	>.6 AND <=.7	>.7 AND <=.8	
REASON		'HomeImp'	'DebtCon'	

Action Term	1	2	3	
BadLoanFlag	True	True	False	

To switch to the list view, click . The list view is a simplified list of rules with operators, condition terms, action terms, and term values in one vertical list. By default, the rules are collapsed. Click to display the logic for all of the rules.

	Rule Name	Operator	Term	Value
▼	Default Rule Name 1			
		IF	LOAN_SCORE	>.5
		THEN	BadLoanFlag	True
▼	Default Rule Name 2			
		IF	HAS_LANDLINE	=1
		AND	LOAN_SCORE	>.6 AND <=.7
		AND	REASON	'HomeImp'
		THEN	BadLoanFlag	True
▼	Default Rule Name 3			
		IF	HAS_LANDLINE	=1
		AND	LOAN_SCORE	>.7 AND <=.8
		AND	REASON	'DebtCon'
		THEN	BadLoanFlag	False

Empty expressions are hidden in the list view. If you change back to the horizontal or vertical view, the empty expressions are re-displayed.

If you switch to the list view from the horizontal or vertical views, OR operators are changed to ELSE operators. You cannot select the OR operator in the list view. If you change back to the horizontal or vertical view, the OR operators remain set to ELSE operators, and every ELSE rule is assigned the appropriate action expressions.

For example, Figure 7.1 on page 83 shows a rule set with OR operators in the horizontal view. The following figure shows the first IF block of the same rule set in the list view.

	Rule Name	Operator	Term	Value
▼	Default Rule Name 1			
		IF	Order_Quantity	>=50 AND <100
		THEN	Offer_Percent	=10
▼	Default Rule Name 2	ELSE		
		IF	Order_History	>=15 AND <30
		THEN	Offer_Percent	=10
▼	Default Rule Name 3	ELSE		
		IF	Referrals	>=5
		THEN	Offer_Percent	=10
▼	Default Rule Name 4	ELSE		
		IF	Status	='Silver'
		THEN	Offer_Percent	=10

Define Expressions for a Rule

To define the expressions for a rule:

1. Add any additional terms to the rule set editor that the rule requires. For example, if your rule is `If balance < 100 then risk = "high"`, then add `balance` as the condition term and `risk` as the action term.

   To add terms to the rule set, select the terms in the Vocabularies pane, and then right-click and select either **Use as condition term** or **Use as action term**. You can also drag the terms onto the rule set editor.

   - To select a consecutive set of terms, click on the first term, hold down the Shift key, and click on the last term. To select nonconsecutive terms, hold down the Ctrl key, and click on each term that you want to select.
   - If the terms that you want to add to the table all belong to the same entity, you can add the entire entity to the table. Note that an entity can contain many terms, and you cannot undo this operation after you have added the terms. In the horizontal and vertical views, you must delete terms one at a time. In the list view, you can select the rows containing the terms, and then right-click and select **Delete**. The expression for the terms is removed, and rows without expressions are hidden in the list.

   In the horizontal and vertical views, terms are added to the column and row headings. In the list view, terms are added to the currently selected rule, so you must select the rule where you want to add a term before adding it.

   TIP By default, terms are displayed in the rule set editor in the same order in which they appear in the Vocabularies pane. You can reorder the terms by dragging the terms in the column or row headings.

   TIP You can add new entities and terms by clicking in the Vocabularies pane. You can also edit existing entities and terms by right-clicking on the entity or

term and selecting the appropriate option. See "Create an Entity" on page 63 and "Create a Term" on page 63 for more information.

2. (Optional) Select the operator for the rule. The default operator is IF. See "Controlling Which Conditions Are Evaluated" on page 82 for more information.

3. For each term that is used in the new rule, specify the expression that applies to that term in the row or column for the new rule. For example, if the rule is **`If balance < 100 then risk = "high"`**, the expression for **`balance`** is **`< 100`**, and the expression for **`risk`** is **`= "high"`**.

   Expressions can be up to 1024 characters long. They can contain numeric constants, character strings, vocabulary terms, operators, and SAS DS2 functions.

   Condition expressions can contain the LOOKUP function, and action expressions can contain the LOOKUPVALUE function. However, if the expression contains the LOOKUP or LOOKUPVALUE function, then the expression cannot contain anything else.

   You can enter expressions directly into the decision table, or you can use the Equation Editor to create and edit expressions. To open the Equation Editor, click in the table cell, and select  .

   As you enter expressions into each cell, SAS Decision Manager displays the rule conditions and actions, including the operators and term names that are added by SAS Decision Manager, on the **Rule Details** tab. (See "Terms and Operators Added by SAS Decision Manager" on page 89.) For example, suppose you enter the following rule in the editor:

		Condition Term		Action Term
#		Credit_Score	Down_Payment	Risk_Category
1	If	>=720 AND <750	>=20	='Medium'

   SAS Decision Manager displays the following expressions on the **Rule Details** tab.

```
Rule expression:

Condition: (Credit_Score >= 720 and Credit_Score < 750) AND (Down_Payment >= 20)

Action: Risk_Category = 'Medium'
```

   For more information about entering expressions, see the following topics:

   - "Tips for Entering Expressions" on page 83
   - "Using the Equation Editor" on page 83
   - "Punctuation for Data Values" on page 86
   - "Operators for Use in Expressions" on page 87
   - "Using the LIKE Operator" on page 88
   - "LOOKUP Function" on page 72
   - "LOOKUPVALUE Function" on page 73
   - "Using Functions in Expressions" on page 89
   - "Working with Missing Values" on page 89
   - "Terms and Operators Added by SAS Decision Manager" on page 89
   - "Leading Plus and Minus Operators" on page 90
   - "Examples of Expressions" on page 90

4. Click to save the rule set. SAS Decision Manager validates the syntax of the expressions. If it does not detect any problems, it saves the rule set. See "Validate the Expressions in a Rule Set" on page 92 for more information.

Controlling Which Conditions Are Evaluated

You add conditional processing within a rule set by using the IF, ELSE, and OR operators. By default, rules are assigned the keyword IF, which means that the rule's condition is evaluated regardless of the results of previous rules. You can change this outcome by changing the operator for a rule to ELSE or OR.

If you set a rule's operator to ELSE, then the rule's condition is evaluated only if the previous rule's condition evaluated to false. For example, given the rule set shown the following display, if Order_Quanitity is 12, the condition for rules 1 and 2 evaluates to false, and the condition for rule 3 evaluates to true. Therefore, the action for rule 3 is executed. The conditions for rules 4 and 5 are not evaluated.

		Condition Term	Action Term
#		Order_Quantity	Offer_Percent
1	If	<5	0
2	Else	>5 AND <=10	=5
3	Else	>10 AND <=20	=10
4	Else	>20 AND <=30	=15
5	Else	>30	=20

Use the OR operator to break up very long condition expressions into multiple condition expressions or to execute the same action expression for each of the several conditions. If you assign the OR operator to a rule, then you cannot enter an action expression for the rule. If any of the conditions evaluate to true, SAS Decision Manager executes the action of the last rule that was assigned the IF or ELSE operator. When you have several consecutive rules that are all assigned the OR operator, only the action for the first rule whose condition evaluates to true is executed. The conditions for the remaining consecutive OR rules are not evaluated.

For example, in the following rule set, rules 1 through 4 all use the action expression that is defined for rule 1. Rules 5 through 8 all use the action expression that is defined for rule 5.

Figure 7.1 *Rule Set That Defines Eight Rules in Two IF Blocks*

		Condition Term				Action Term
#		Order_Quantity	Order_History	Referrals	Status	Offer_Percent
1	If	>=50 AND <100				=10
2	Or		>=15 AND <30			
3	Or			>=5		
4	Or				='Silver'	
5	If	>=100				=20
6	Or		>=30			
7	Or			>=5		
8	Or				='Platinum'	

An IF block is a series of rules that begins with an IF operator and extends up to but does not include the next IF operator. In the rule set shown in Figure 7.1 on page 83, rules 1 to 4 are an IF block and rules 5 to 8 are a second IF block.

Tips for Entering Expressions

- A rule that does not have a condition expression always executes.
- A rule that does not have a condition expression must be the last rule in an IF block.
- If you assign the IF operator and a condition expression to a rule but that rule does not have an action expression, the condition expression is evaluated, but no action is taken. (See "Controlling Which Conditions Are Evaluated" on page 82 for information about the IF operator.)
- SAS Decision Manager issues a warning message if a term meets both of the following criteria:
  - The term is excluded from input but is not excluded from output.
  - The term is used only in a condition expression.

  If a term is excluded from input and included in output, you must assign it a value before using it in a condition expression.

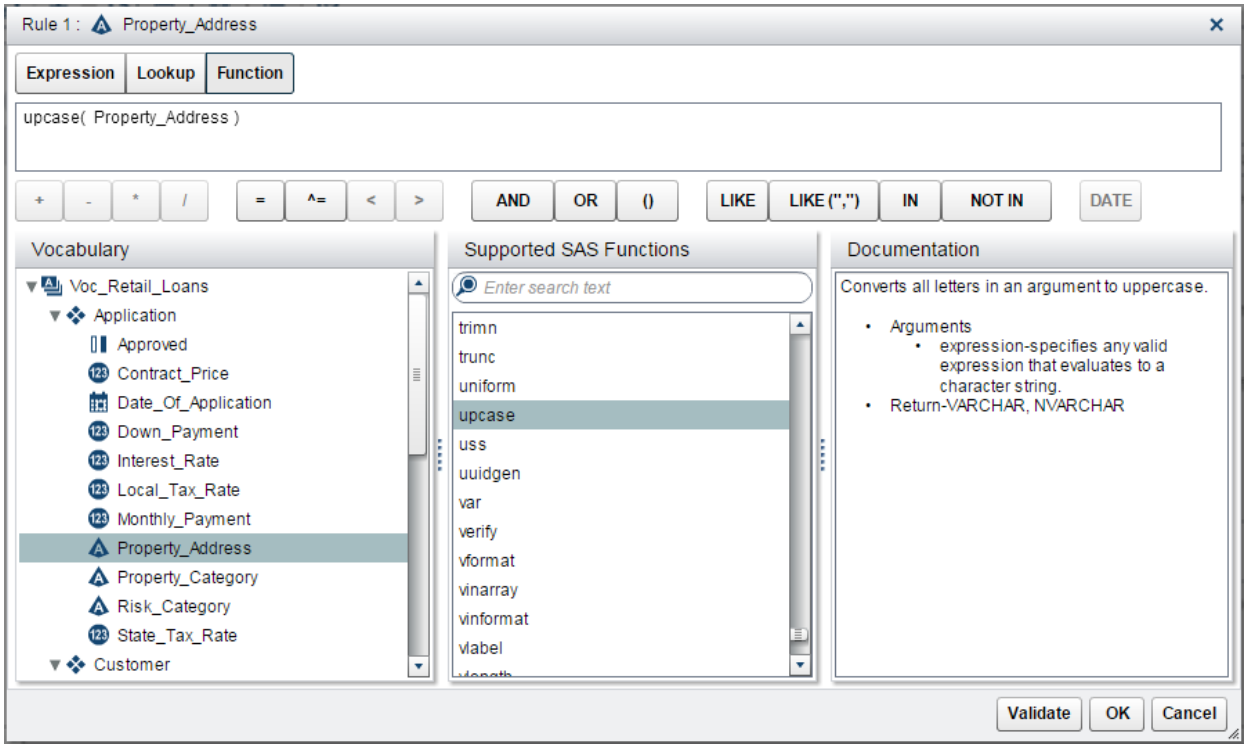
Using the Equation Editor

Open the Equation Editor

To open the Equation Editor, click in a cell in the rule set editor, and click  .

Note: The **Lookup** tab is available only for condition terms, and the **Lookup Value** tab is available only for action terms. Buttons for some of the operators might be disabled depending on the data type of the term, and because action expressions can be assignment expressions only.

Figure 7.2 *Expression Tab for a Condition Term*



Build an Expression in the Equation Editor

As you add elements to an expression, the Equation Editor builds the expression in the top field of the **Expression** tab. Click on the operators, vocabulary terms, and domain values as needed to add them to the expression. To add date constants to the expression, click **Date**. To add numeric constants or character strings to the expression, enter them directly into the top field. (Remember to use the correct punctuation. See "Punctuation for Data Values" on page 86.)

To build an expression that uses a DS2 function, click the **Function** tab. Click on a function name to display information about the syntax for that function. Double-click on a function name to add the function to your expression.

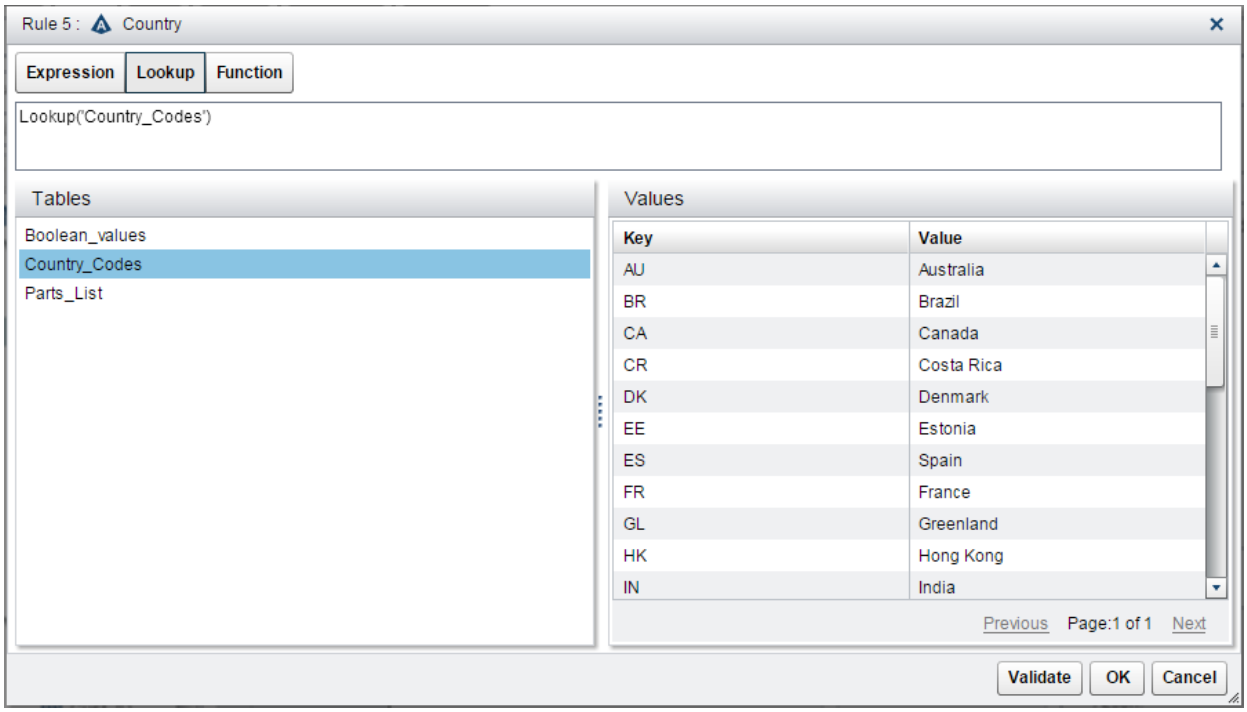
To build an expression that uses the LOOKUP or LOOKUPVALUE functions, click the **Lookup** or **LookupValue** tabs. You can enter the LOOKUP function in condition expressions only, and you can enter the LOOKUPVALUE function in action expressions only. See "Specify the LOOKUP Function in the Equation Editor" on page 85 and "Specify the LOOKUPVALUE Function in the Equation Editor" on page 85 for more information.

When you are finished building the expression, click **OK**. The Equation Editor validates the syntax of the expression. If the validation is successful, the editor adds the expression to the cell in the table from which you opened the editor. You can click **Validate** at any time to check the syntax of the expression that you are building.

Specify the LOOKUP Function in the Equation Editor

To use the Equation Editor to enter the LOOKUP function, click the **Lookup** tab. Double-click on the lookup table name that you want to specify in the function call and click **OK**.

Figure 7.3 *Lookup Tab in the Equation Editor*



For more information, see "LOOKUP Function" on page 72.

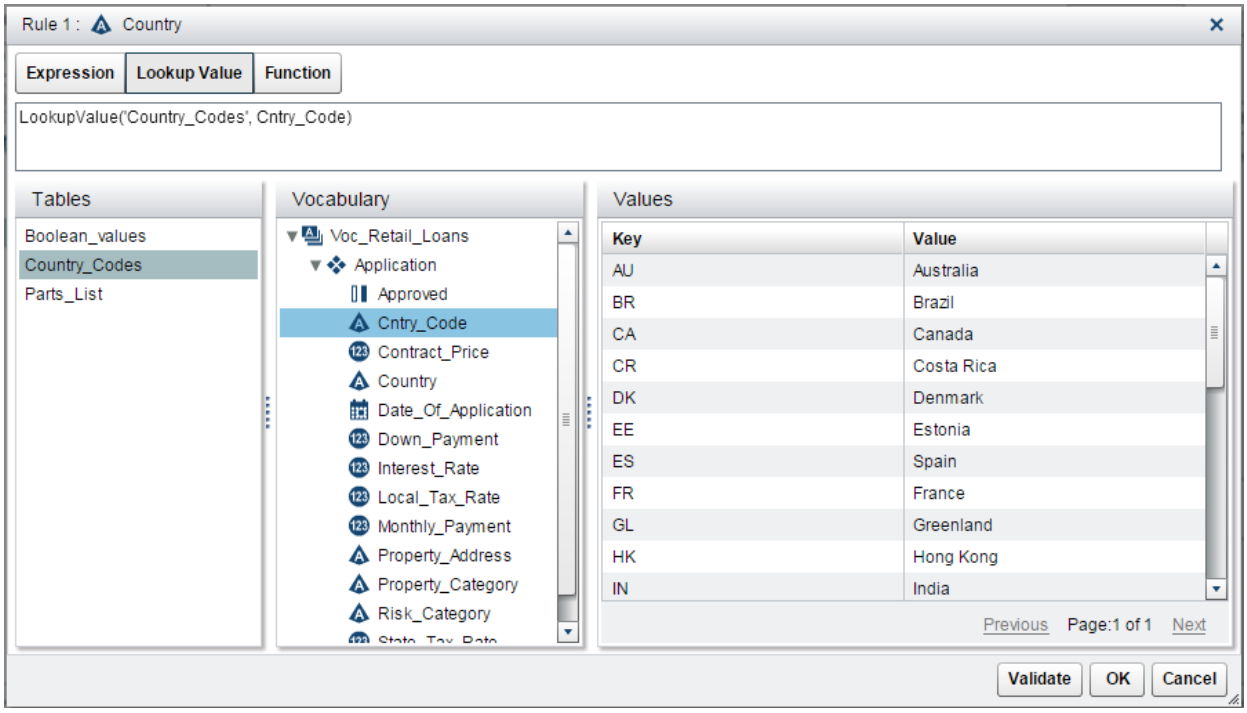
Specify the LOOKUPVALUE Function in the Equation Editor

To use the Equation Editor to enter the LOOKUPVALUE function:

1. Click the **LookupValue** tab.
2. Double-click on the lookup table name.
3. Specify the term name or the character string that contains the lookup key value. To specify a term, double-click on the term in the **Vocabulary** column. To specify a character string as the lookup key value, enter the character string in the field at the top of the **LookupValue** tab. Enclose the string in quotation marks.
4. (Optional) Click **Validate** to check the syntax of the expression.
5. Click **OK**.

For more information, see "LOOKUPVALUE Function" on page 73.

Figure 7.4 *LookupValue Tab in the Equation Editor*



Punctuation for Data Values

Values for some data types might need to be enclosed in quotation marks, as shown in the following table. Date and Datetime values must be followed with **d** and **dt**, respectively.

Table 7.1 *Punctuation Needed for Data Values*

Data Type	Punctuation Needed	Example

Character	Enclose character strings in either single or double quotation marks.	`='Gold Account'`
		`="Ineligible"`
		`="d'oscail"`

Date	Enter Date values by using the format DDMMMYYYY. Enclose each value in quotation marks followed by **d**.	`='01AUG2015'd` `>='31AUG2015'd`
Datetime	Enter Datetime values by using the format DDMMMYYYY:HH:MM:SS. Use 24-hour clock notation. Enclose each value in quotation marks followed by **dt**.	`='01AUG2015:15:00:00'dt` `<='31AUG2015:15:00:00'dt`
Boolean	Boolean values are not enclosed in quotation marks. Enter only **True** or **False**.	`=True` `=False`

Operators for Use in Expressions

The following table lists the operators that you can use in an expression. Do not enter a space between the elements of the operators <=, >=, or ^=. Some mnemonic equivalents for these operators cannot be used in SAS Decision Manager expressions. See "Operators in Expressions " in *SAS DS2 Programmer's Guide* for more information about the operators shown in the table.

Table 7.2 *Operators*

Operator	Definition	Example
*	Multiply	`0.085 * sales`
/	Divide	`amount / 5`
+	Add	`num + 3`
–	Subtract	`sale - discount`
=	Equal to	`= maxTriesAllowed`
+*value*	Leading plus*	+60
^=	Not equal to	`insufficientFunds ^= True`
>	Greater than	`daysLate > 5`
<	Less than	`num < 8`
>=	Greater than or equal to	`balance >= 1000`
<=	Less than or equal to	`balance <= 250`
IN (*value-list*)	Equal to an item in *value-list*	`in ('high','medium','low')`
NOT IN (*value-list*)	Not equal to an item in *value-list*	`not in (10,20,30)`
LIKE '*pattern-matching-expression*' LIKE ('*pattern-matching-expression*','*pattern-matching-expression*')	If the term's value matches *pattern-matching-expression*, the result is true.	`like 'HS%PP'` `like ('_976%','_223%')`
expression AND *expression*	If both expressions are true, the result is true.	`dateExpired >= '01AUG2015'd AND dateExpired <= '31AUG2015'd`
expression OR *expression*	If either expression is true, the result is true.	`dateEnrolled >= '01JAN2015' OR member = True`

* The application supports the leading plus (+) operator in action expressions only.

Using the LIKE Operator

Note: The LIKE operator does not work if the code type is set to DS1. See "Generating DATA Step Code for a Rule Flow" on page 113 for more information.

The LIKE operator determines whether the value of a term matches a pattern-matching expression. The syntax of an expression that uses the LIKE operator is as follows:

LIKE '*pattern-matching-expression*'

LIKE ('*pattern-matching-expression*'<,'*pattern-matching-expression*'>)

If a term's value matches the pattern that is specified by *pattern-matching-expression*, the expression evaluates to true (1). Otherwise, the expression evaluates to false (0).

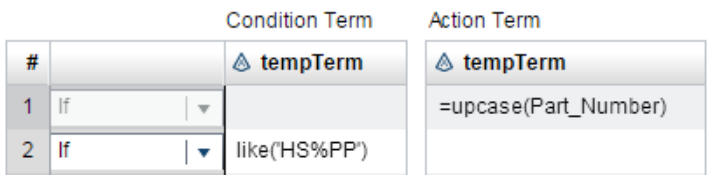
There are three classes of pattern-matching characters.

Table 7.3 *Pattern-Matching Characters*

Character	Description
underscore (_)	Matches any single character
percent sign (%)	Matches any sequence of zero or more characters *Note:* Be aware of the effect of trailing blanks. To match values, you might have to use the TRIM function to remove trailing blanks.
any other character	Matches that character

The LIKE expression is case sensitive. To search for mixed-case strings, use the UPCASE function to create an uppercase version of the term that you want to search. You can use a temporary term to store the results of the UPCASE function. (See Step 8 of "Create a Term" on page 63.) Use the LIKE operator to search the uppercase version of the term.

For example, you can search the term Part_Number for mixed-case strings that begin with HS and end with PP by using the two rules shown in the following display.



The following table shows examples of the matches that result if you search a term that could have these values: Smith, Smooth, Smothers, Smart, Smuggle.

Table 7.4 *Examples of LIKE Expressions*

LIKE Expression Example	Matching Results
`like 'Sm%'`	`Smith, Smooth, Smothers, Smart, Smuggle`

LIKE Expression Example	Matching Results
`like '%th'`	`Smith, Smooth`
`like 'S__gg%'`	`Smuggle`
`like 'S_o'`	(no matches)
`like 'S_o%'`	`Smooth, Smothers`
`like 'S%th'`	`Smith, Smooth`

Using Functions in Expressions

SAS Decision Manager supports the following functions in rule expressions:

- LOOKUP and LOOKUPVALUE functions. See "LOOKUP Function" on page 72 and "LOOKUPVALUE Function" on page 73 for more information.
- SAS DS2 functions. You can click on a function name in the Equation Editor to display information about the syntax for that function. For additional information about DS2 functions, see *SAS DS2 Language Reference*.

Working with Missing Values

You can use the MISSING function to check for missing values. This function returns a 0 (false) or 1 (true). Missing values have a value of **`false`** when you use them with logical operators such as AND or OR.

For more information, see "How DS2 Processes Nulls and SAS Missing Values" in *SAS DS2 Programmer's Guide*.

Terms and Operators Added by SAS Decision Manager

As you enter expressions into each cell, SAS Decision Manager displays the rule conditions and actions on the **Rule Details** tab. The operators and term names that are added by SAS Decision Manager are also displayed. Remember these rules when you are entering expressions:

- If you do not specify an operator at the beginning of an expression, SAS Decision Manager adds an equal sign to the beginning of the expression. For example, if you enter **`5+x`** for an expression, the expression resolves to **`=5+x`**.
- When an AND or OR operator is followed immediately by another operator in a condition expression, SAS Decision Manager inserts the condition term between the AND or OR operator and the operator that follows it. For example, if you enter **`>5 and <10`** for **`myterm`**, the expression resolves to **`myterm>5 and myterm<10`**. SAS Decision Manager inserts the term for top-level AND or OR operators in condition expressions only. It does not insert the term with nested AND or OR operators or in action expressions.

Leading Plus and Minus Operators

If you specify the leading plus operator in an action expression, SAS Decision Manager adds the term name to the expression. Leading minus operators are not supported.

The condition expression `+1` is invalid. If you enter `+1` as an action expression, the expression resolves to `x=x + 1`. The expression `=+1` is invalid as both a condition and as an action expression.

If you enter `-1` as either a condition or an action expression, the expression is interpreted as a negative number and not as a leading minus operator. The expression resolves to `x=-1`.

Examples of Expressions

The following table shows examples of expressions that you can specify.

Table 7.5 *Examples of Expressions*

Expression as Entered into the Decision Table for Term X	Resulting Expression	Valid as a Condition Expression	Valid as an Action Expression
`5`	`x=5`	Yes	Yes
`=5`	`x=5`	Yes	Yes
+10	`x=x+10`	No See Note 7.	Yes
–10	`x=-10`	Yes	Yes See Note 7.
`"mystring"`	`x="mystring"`	Yes	Yes
`=term1`	`x=term1`	Yes	Yes
`5 or >100`	`x=5 or x>100`	Yes	No See Note 1.
`^=5 and x<10`	`x^=5 and x<10`	Yes	No See Note 1.
`^=5 or >=(100/4)`	`x^=5 or x>=(100/4)`	Yes	No See Note 1.
`in (10,20,30)`	`x IN (10,20,30)`	Yes	No See Note 1.
`not in ('med','high')`	`x NOT IN ('MED','HIGH')`	Yes	No See Note 1.

Expression as Entered into the Decision Table for Term X	Resulting Expression	Valid as a Condition Expression	Valid as an Action Expression
`rate in ('med','high')`	`x = rate in ('med','high')`	Yes	Yes See Note 8.
`like ('M77__LL%','MA89_LL%')`	`x LIKE ('M77__LL%','MA89_LL%')`	Yes	No See Note 1.
`<'10JUN2012'd`	`x<'10JUN2015'd`	Yes	No See Note 1.
`>'10JUN2012:17:00:00'dt`	`x>'10JUN2015:17:00:00'dt`	Yes	No See Note 1.
`=ABS(-10)`	`x=ABS(-10)`	Yes	Yes
`=True`	`x=True`	Yes	Yes
`False`	`x=False`	Yes	Yes
`&myMacroVar`	`x=&myMacroVar`	Yes See Note 2.	Yes See Note 2.
`%EVAL(&myMacroVar)`	`x=%EVAL(&myMacroVar)`	Yes See Note 2.	Yes See Note 2.
`term1=5`	`x=term1=5`	Yes See Note 3.	Yes See Note 4.
`term1=3 or term2=5`	`x=term1=3 or term2=5`	Yes See Note 5.	Yes See Note 6.
`5 or (x>10 and <20)`	This expression is invalid as both a condition expression and as an action expression. SAS Decision Manager does not add column names after nested AND or OR operators.		
`>"mystring"`	This expression is invalid as both a condition expression and as an action expression. SAS Decision Manager checks whether literal types are compatible with the specified operators. Character strings are not compatible with numeric operators.		
`=4.6927e-101`	`x=4.6927e-101`	Yes See Note 9.	Yes

Notes:

1. Action expressions must be assignment expressions only.
2. SAS Decision Manager validates macro functions and variables based only on whether the syntax is correct. It does not check to determine whether a macro

function or variable will be accessible when the rule set is executed. Macro support in expressions is controlled by a configuration property in SAS Management Console. See "Support macros in rule expressions" in *SAS Decision Manager: Administrator's Guide* for more information.

3. This expression is valid. However, it should be avoided. As a condition statement, this expression checks to determine whether both **x** and **term1** are equal to 5. The recommended way to enter this expression is **=5 and term1=5**.
4. As an action expression, this expression becomes a Boolean assignment statement. This expression determines whether **term1** is equal to 5, and if so, assigns a value of 1 (true) to **x**. If not, it assigns a value of 0 (false) to **x**.
5. This expression is valid. However, it should be avoided. As a condition statement, this expression checks to determine whether both **x** and **term1** are equal to 3 or whether **term2** is equal to 5. The recommended way to enter this expression is **(=5 or term1=3) or term2=5**.
6. As an action expression, this expression becomes a Boolean assignment statement. This expression determines whether either **term1** is equal to 3 or **term2** is equal to 5, and if so, assigns a value of 1 (true) to **x**. If not, it assigns a value of 0 (false) to **x**.
7. Leading plus (+) operators are valid in action expressions only. Leading minus (–) operators are not supported. See "Leading Plus and Minus Operators" on page 90 for more information.
8. As an action expression, this expression becomes a Boolean assignment statement. The expression determines whether **rate** is equal to **high** or **low**, and if so, assigns a value of 1 (true) to **x**. If not, it assigns a value of 0 (false) to **x**.
9. This expression is valid. However, you should use caution when testing for equality by using scientific notation. Two numbers that appear to be the same might evaluate to different numbers because of the precision involved in scientific notation.

Validate the Expressions in a Rule Set

When you save a rule set, SAS Decision Manager checks whether the syntax in the expressions is valid, and if so, saves the rule set. However, you can click to check the syntax of rule expressions at any time.

SAS Decision Manager checks whether the results produced by the expressions are of the correct data type for the terms to which the expressions apply. Also, when domain values are defined for a term, SAS Decision Manager does not check whether the values that are assigned to the term are included in the list of domain values.

Change the Order of Rules in a Rule Set

There are two ways to change the order of the rules in a rule set. You can move a single rule to a new position, and SAS Decision Manager adjusts the position of the remaining rules in the rule set. Alternatively, you can swap the position of two rules, and SAS Decision Manager leaves the remaining rules in their original positions.

Move a Rule to a New Position in a Rule Set

Note: If you move a rule that uses the ELSE or OR operator to position 1 in the rule set, the operator is changed to IF.

To move a rule:

1. Right-click on the order number (in the horizontal and vertical views) or on the rule name (in the list view) of the rule that you want to move, and select **Reorder**. The Reorder the Rule window appears.
2. Select the new position number for the rule.
3. Click **OK**. SAS Decision Manager moves the rule to the new position and repositions the remaining rules up or down as needed.

Swap Two Rules

Note: If you move a rule that uses the ELSE or OR operator to position 1, the operator is changed to IF.

To swap the position of two rules:

1. Right-click on the order number (in the horizontal and vertical views) or on the rule name (in the list view) of one of the rules that you want to move, and select **Swap**. The Swap the Rule window appears.
2. Select the position number for the second rule that you want to move.
3. Click **OK**. SAS Decision Manager swaps the positions of the two rules and leaves all other rules in their original positions.

Copy Rules and Expressions

You can copy individual rules and expressions only within the same rule set.

Copy an Entire Rule

To copy and paste an entire rule:

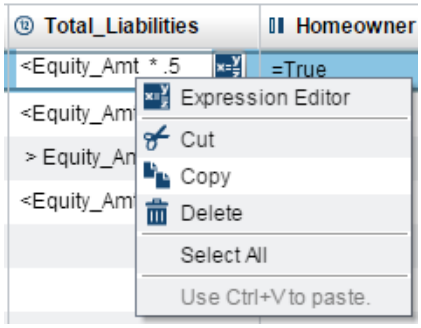
1. Right-click on the order number (in the horizontal and vertical views) or on the rule name (in the list view) of the rule that you want to copy and select **Copy**.
2. Right-click in the rule set, and select **Paste**. SAS Decision Manager adds the copied rule as the last rule in the rule set. You can then edit or reorder the new rule as needed.

Copy Text within a Rule

To copy and paste an expression or part of an expression:

1. Select the expression that contains the text that you want to copy.

2. Select the text that you want to copy. To select all of the text in a cell, right-click and select **Select All**.
3. Right-click on the text and select **Copy**.



4. Select the expression in which you want to paste the text, and press Ctrl+V.

Delete Terms, Rules, and Expressions

Delete a Term from a Rule Set

You cannot delete a term that is used in a rule flow or a decision that has been published.

In the horizontal and vertical views, right-click on the term in the column or row heading, and select **Delete Term**.

Note: In the list view, if you right-click on a row containing an expression and select **Delete**, the expression in that row is deleted from the rule The term remains in the rule set until you delete all of the expressions that use that term.

Delete a Rule

You can delete only one rule at a time. Right-click on the order number (in the horizontal and vertical views) or on the rule name (in the list view), and select **Delete Rule**.

You cannot delete the last rule in a rule set. You must delete the rule set instead.

Cut and Paste Text within a Rule

To cut and paste an expression or part of an expression:

1. Select the expression that contains the text that you want to cut.
2. Select the text that you want to cut. To select all of the text in a cell, right-click and select **Select All**.
3. Right-click on the text and select **Cut**.
4. Select the expression in which you want to paste the text, and press Ctrl+V.

Edit the Properties of a Rule Set

To edit the properties of a rule set, open the rule set and select the **Properties** page. You can edit the name and description. If the rule set is empty, you can change the vocabulary that is associated with the rule set. If any rules have been defined for the rule set, you cannot change the vocabulary.

The **Rule Set Logic** section of the properties page displays all of the rules in the rule set and includes the operators and term names that have been added by SAS Decision Manager.

Edit the Properties of a Rule

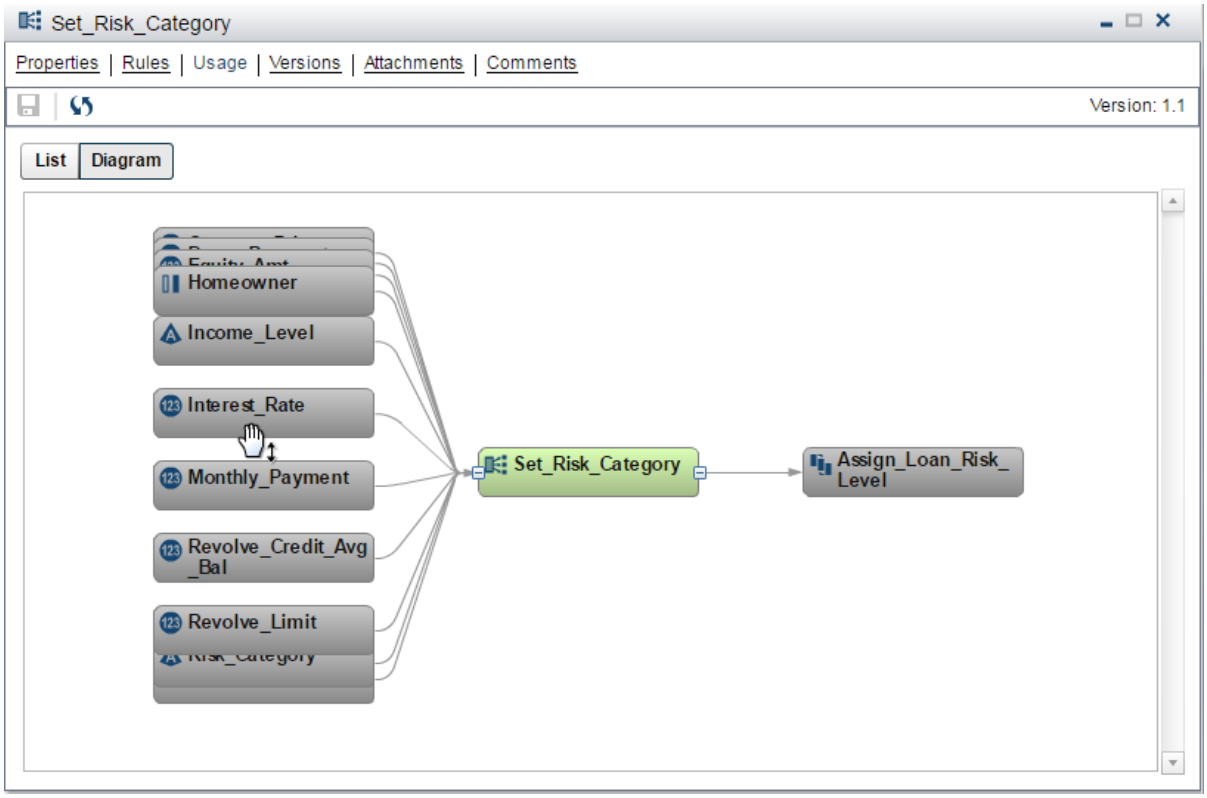
To edit the properties of a rule:

1. Open the rule set that contains the rule, and select the **Rules** page.
2. Select the rule.
3. Edit the properties as needed. You can edit the name, edit the description, or select whether rule-fired data is recorded for the rule.
4. Click .

Display Usage Information for a Rule Set

To display usage information for a rule set, select the **Usage** page. Click **List** to display the terms and lookup tables that are referenced in the rule set and rule flows that use the rule set. Click **Diagram** to display a diagram showing the rules, rule set, and any rule flows that use the rule set.

Note: The diagram displays information from only the current version of a rule flow. If a published version of a rule flow uses a rule set but the current version of the same rule flow does not, then the rule flow does not appear in the diagram.



Managing Rule Set Versions

About Rule Set Versions

The *latest* version of a rule set is the rule set that has the highest version number. It is also the last version that you saved. You can edit only the latest version of a rule set.

Only one version of a rule set can be unlocked at a time. If you create a new version of a rule set, SAS Decision Manager locks the existing latest version before it creates a new latest version.

To edit a rule set, it must be unlocked. You cannot unlock a rule set. To make changes to a rule set that has been locked, you must create a new version of the rule set and make changes to the new version.

To publish a rule flow, all of the rule set versions that are referenced in the rule flow must be locked. When you publish a rule flow, SAS Decision Manager automatically locks any unlocked rule set versions that are used in the rule flow.

Set the Displayed Version

On the **Versions** page, indicates the displayed version. The displayed version is the rule set whose information is displayed on all other pages, except for the **Properties** page. The **Properties** page displays information for the entire rule set, not for a specific version of the rule set.

To change the displayed version, select the version that you want to view, and click .

Create a New Version of a Rule Set

Note: When you create a new version of a rule set, SAS Decision Manager locks the latest version of the rule set if it is not already locked.

Note: You cannot save changes to a rule set that is locked. If you modify a rule set that is locked and click , SAS Decision Manager asks you if you want to save the changes as a new version.

To create a new version of a rule set:

1. Select the **Versions** page.
2. Click . The Create New Version window appears.
3. Select the version type: **Minor** or **Major**. Version numbers follow the format `Major.Minor`. If you select **Major**, the number to the left of the period is incremented. If you select **Minor**, the number to the right of the period is incremented.
4. (Optional) Enter a description.
5. Click **OK**.

Lock a Rule Set Version

Note: You cannot make changes to a rule set after it has been locked. You cannot unlock a rule set version.

To publish a rule flow, all of the rule set versions that are referenced in the rule flow must be locked. When you publish a rule flow, SAS Decision Manager automatically locks any unlocked rule set versions that are used in the rule flow.

To manually lock a version of a rule set:

1. Select the **Versions** page.
2. Select the version of the rule set that you want to lock.
3. Click .

Note: You can also lock a version by clicking **Lock** in the Edit Version window.

Edit a Version Description

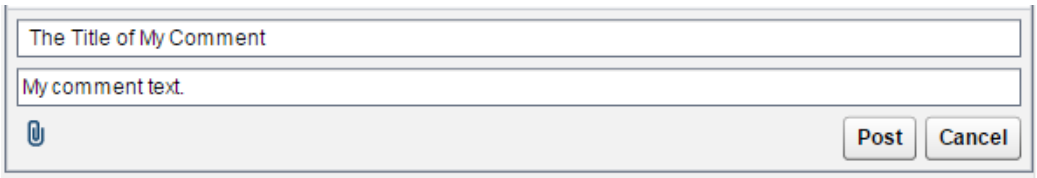
To edit a version description:

1. Select the **Versions** page.
2. Select the version of the rule set that you want to edit.
3. Click . The Edit Version window appears.
4. Edit the version description.
5. Click **OK** or **Lock** (if you also want to lock the version).

Add Comments to a Rule Set

You can add new comments or reply to existing comments. To add a new comment:

1. Select the **Comments** page.
2. Enter a topic title and enter the comment. The topic title is required, and the field for comments does not appear until you enter the topic title.



3. (Optional) Click to add an attachment such as an image or a document.

   Note: You cannot attach executable files to a comment.
4. Click **Post**.

To reply to an existing comment, enter your reply in the field immediately below the topic title for the existing comment, and click **Post**.

Click to see comments that have been posted by others.

To search for text in the comments, enter text in the search field at the top of the **Comments** page.

Add Attachments to a Rule Set

To add an attachment such as a document file or an image file:

1. Select the **Attachments** page.
2. Click +, and select the attachment file.

   Note: You cannot attach executable files to rule sets, rule flows, or data tables. Depending on your site's settings, you might be able to attach executable files to models, projects, and portfolios. For more information, see "Configure Support for Executable Attachments" in *SAS Decision Manager: Administrator's Guide*.
3. Click .

Note: You can delete an attachment by selecting the attachment and clicking .

Duplicate Rule Sets

To duplicate a single rule set:

1. Right-click on the rule set and select **Duplicate**. The Duplicate window appears.

2. Enter the name for the duplicate rule set.
3. (Optional) Enter a description for the rule set.
4. Select the folder where you want save the duplicate rule set.
5. Select the version of the rule set that you want to duplicate.
6. Click **OK**. If you do not enter a new name and the folder in which you save the duplicate rule set already has a rule set with the same name, SAS Decision Manager appends an underscore and a number to the name.

To duplicate multiple rule sets:

1. Select the rule sets that you want to duplicate.
2. Right-click and select **Duplicate**. The Choose a Location window appears.
3. Select the location where you want the save the duplicate tables and click **Duplicate**.

   SAS Decision Manager duplicates the current version of the selected rule sets.

   If you do not enter a new name and the folder in which you save the duplicate rule set already has a rule set with the same name, SAS Decision Manager appends an underscore and a number to the name.

Move Rule Sets

You cannot move a rule set if it is open. To move rule sets:

1. Select the rule sets that you want to move, and then right-click and select **Move**. The Choose a Location window appears.
2. Select a new location for the rule sets, and click **Move**.

Delete Rule Sets

Note: You cannot delete a rule set if it is used in a rule flow or if it is open.

To delete rule sets, select the rule sets that you want to delete, and click .

Validate and Save a Rule Set

To save changes to a rule set, click . SAS Decision Manager validates the syntax of the expressions and displays an error message if it finds any problems. If SAS Decision Manager does not detect any problems with any of the expressions, it saves the rule set. See "Validate the Expressions in a Rule Set" on page 92 for more information.

Chapter 8

Creating and Publishing Rule Flows

Introduction to Rule Flows

A business rule flow is a logical collection of multiple rule sets that define multiple conditions and actions. In general, the rule sets in a rule flow are executed in the order in which they are defined in the rule flow. However, with complex rule flows, certain sections of rule sets are usually executed more times than others. See "Simple Rule Flows, Complex Rule Flows, and BY Groups" on page 102 for more information.

After you publish a rule flow, other applications can deploy the published rule flows. The applications map terms used in the rule flow to table column in input data, evaluate the conditions in the rule flow, and execute the appropriate actions.

Simple Rule Flows, Complex Rule Flows, and BY Groups

There are two general types of rule flows: simple and complex. A simple rule flow has a single group of rule sets. All of the rule sets are run and output is generated for each input record.

A complex rule flow has at least three sections: Initial, Main, and Final. Rule sets in the Initial section are run only when the first input record is processed. Rule sets in the Main section are run for each input record. Rule sets in the Final section are run after the last input record has been processed by the rule sets in the Main section.

For complex rule flows, you can specify *BY-group terms*. If you specify BY-group terms, then SAS Decision Manager sorts the input data by those terms.

If you specify BY-group terms, SAS Decision Manager adds two new sections to the rule flow, Group Start and Group End. The rules sets in these sections are run with the first and last input record in each BY group.

Note: Adding rules to any of the sections in a complex rule flow is optional. You can leave unneeded sections empty.

See "When Are Output Records Generated?" on page 115 for more information.

Create a Rule Flow Using the Rule Flow Editor

To create a rule flow:

1. Select the **Business Rules** ⇨ **Rule Flows** category.
2. Right-click on the folder where you want to create the new rule flow, and select **New Rule Flow**. Alternatively, select the folder where you want to add the new rule flow, click , and select **New Rule Flow**. The New Rule Flow window appears.
3. Enter a name for the new rule flow. Rule flow names are limited to 32 characters and can contain any character except forward slash (/), backslash (\), left brace ({), right brace (}), colon (:), and question mark (?).
4. (Optional) Enter a description for the new rule flow. Descriptions are limited to 256 characters.
5. (Optional) Select **Create output only for records that fire rules** to limit the output of the rule flow. By default, all output records are written to the output data set. However, for some types of applications, only the output records for which at least one rule has fired are of interest. Limiting output is useful for applications that detect outliers, such as applications that detect fraud.
6. Click **Create**. SAS Decision Manager creates a new rule flow and opens the rule flow editor. By default, the **Properties** page is displayed.
7. Select the **Rule Sets** page.
8. (Optional) Select **Complex Rule Flow** from the menu in the toolbar. SAS Decision Manager adds Initial and Final sections to the rule flow table. The rules in these sections are run at the start and end of the rule flow.
9. Add rule sets to the rule flow. Right-click on a rule set in the **Rule Sets** pane, and select the appropriate **Add To *Section*** option. The options that are available depend on whether you are creating a simple or complex rule flow.

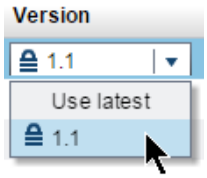
   Note: A rule flow can use only rule sets that are defined for the same vocabulary. After the first rule set is added to the rule flow, the vocabulary for the rule flow is established. Only the rule sets that use the same vocabulary are displayed in the Resources pane.

   Note: A rule set can be added to the same rule flow only once.
10. (Optional) If you selected **Complex Rule Flow**, you can specify BY-group terms. With BY-group processing, all of the input records that have the same values for the BY-group terms are processed before output is generated. One output record is written for each group.

    a. Click and select the BY-group terms. SAS Decision Manager adds Group Start and Group End sections to the table. The rules in these groups are run at the start and end of each BY group.

    b. (Optional) Add the rule sets that you want in the Group Start and Group End sections of the table.
11. (Optional) Select the version of each rule set. If the version is **Use latest**, then the most recently saved version of the rule set is always used when it is run. Specifying **Use latest** for the version is useful during rule flow development and testing. However, if the version of a rule set that is specified in the rule flow is unlocked

when the rule flow is published, SAS Decision Manager automatically locks the rule set version.



12. (Optional) Reorder the rule sets. To move a rule set, select the rule set, and click ⬆ or ⬇ to move it to a different row in the table. To move a rule set to a different section (Initial, Main, and so on), you must remove the rule set, and then add it to the other section. To remove a rule set, select the rule set and click ✕.
13. (Optional) Clear the check boxes in the **Run** column for any rules or sections that you do not want to be run the next time the rule flow is run. Selectively running certain rule sets is useful during rule flow development and testing.
14. Click 💾 to save the rule flow.

Create a Rule Flow by Using Discovery Techniques

About the Discovery Techniques

With the New Discovery wizard, you can use discovery techniques to define vocabularies, terms, rule sets, and rule flows. The discovery techniques that you can select from are:

Decision Tree

Decision Tree analysis produces a tree-like structure in which each branch of the tree represents a possible decision or event. The tree structure shows how one choice leads to the next. Each branch represents a mutually exclusive option. Decision trees are often used for data segmentation or prediction modeling. You can create decision trees to classify observations based on the values of nominal, binary, or ordinal targets or to predict outcomes for interval targets.

Note: With the Decision Tree technique, input columns with a SAS datetime format or a date format other than MONTH*w*. and WEEKDAY*w*. are excluded from the rule discovery process.

Scorecard

Scorecards provide a quantitative score of the odds that a customer will display a defined behavior such as respond positively to a campaign, make a purchase, default on a loan, and so on. The higher the score, the more likely the defined behavior will occur. The SAS Decision Manager Scorecard uses the Weight of Evidence technique to generate scores.

Note: With the Scorecard technique, input columns with a SAS datetime format or a date format other than MONTH*w*. and WEEKDAY*w*. are excluded from the rule discovery process.

Note: The Scorecard technique requires a SAS Enterprise Miner license.

Recency Frequency Monetary (RFM)
RFM is a technique that is used to identify existing customers who are most likely to respond to a new campaign or product offer. RFM analysis looks at when a customer last placed an order or bought something, how often the customer makes a purchase, and how much money they spend. Customers are assigned scores based on these factors.

Market Baskets
Market Basket analysis is used to predict items that are most likely to be purchased together. Market Basket analysis can be used to predict what items a customer is likely to buy.

Create a Rule Flow by Using the New Discovery Wizard

When you run the New Discovery wizard, it uses the discovery technique that you select to generate a rule flow and as many rule sets as are needed. If you do not select an existing vocabulary, the wizard also generates a vocabulary.

Note: The New Discovery wizard produces temporary data sets during the rule discovery process. Do not delete these temporary data sets before you attempt to import the results of the rule discovery process. If you delete these temporary data sets, you cannot import the generated rule sets.

Note: If folder configuration is enabled, you might not be able to import the results of the rule discovery process. See "Enable Business Rules Folder Administration" in *SAS Decision Manager: Administrator's Guide* for more information.

To create a rule flow using the New Discovery wizard:

1. Select the **Business Rules** ⇨ **Rule Flows** category.
2. Right-click on the folder where you want to create the new rule flow, and select **New Rule Flow**. Alternatively, select the folder where you want to add the new rule flow, click , and select **New Rule Flow**. The New Rule Flow window appears.
3. Enter a name for the new rule flow. Rule flow names are limited to 32 characters and can contain any character except forward slash (/), backslash (\), left brace ({), right brace (}), colon (:), and question mark (?).

   Note: The name that you enter is also used for the vocabulary name if you do not select an existing vocabulary. Vocabulary names must be unique within the SAS Decision Manager database. Rule flow names can contain spaces but vocabulary names cannot. If the name that you enter contains a space, it is converted to an underscore in the vocabulary name.
4. (Optional) Enter a description for the new rule flow. Descriptions are limited to 256 characters.
5. (Optional) Select **Create output only for records that fire rules** to limit the output of the rule flow. By default, all output records are written to the output data set. However, for some types of applications, only the output records for which at least one rule has fired are of interest. Limiting output is useful for applications that detect outliers, such as applications that detect fraud.
6. Select **Use discovery techniques to generate rules**.
7. Either select an existing vocabulary or select **Create a vocabulary**.

   Note: If you select an existing vocabulary, and the discovery process generates a vocabulary that has a term with the same name but a different data type, you cannot import the rules that are generated.

8. Click **Create**. SAS Decision Manager opens the New Discovery window.
9. Select the **Discovery technique**. The techniques that are available depend on the products that are licensed at your site. The Recency Frequency Monetary (RFM) technique is available with Base SAS. The Decision Tree and Scorecard techniques require a SAS/STAT license. The Market Baskets technique requires a SAS Enterprise Miner license.
10. Select the **Data source** that you want to use for the discovery analysis.

    Note: You cannot use the Market Baskets discovery technique with data sources that contain values for the **Item** term that do not conform to the SAS name rules for the VALIDVARNAME=V7 system option. See "VALIDVARNAME= System Option" in *SAS System Options: Reference* for more information.
11. Select the setup options for the discovery technique, and click **Next**. The setup options depend on the technique. See Table 8.1 on page 106.
12. Select the action terms that are required for the discovery technique. See Table 8.1.

    Note: If you specified an existing vocabulary in Step 7, and the action terms that you select are excluded from the output data, the rule flow will not run. See "Create a Term" on page 63 for more information.

    For the RFM and Market Baskets techniques, skip to Step 14.
13. For the Decision Tree and Scorecard discovery techniques, select the input variables that you want to be used as condition terms in the rule flow. Select the terms and click ➕ to move them to the **Conditions** list.
14. Click **Run** to run the analysis. SAS Decision Manager displays the rule sets that were generated by the analysis. You should check the SAS log before importing the data.
15. Click **Import** to import the data. If the data was imported successfully, SAS Decision Manager displays a confirmation message telling you what data was imported and which folder it was added to.
16. (Optional) Click **Rule_generation_log** and **Rule_import_log** to download the log files to your local machine. The log filename is `RuleFlowName_generation.log`, and the import log filename is `RuleFlowName_import.log`. If rules cannot be generated or the import process fails, the log files contain detailed error messages.
17. Click **Close** to close the New Discovery wizard. SAS Decision Manager opens the new rule flow in the rule flow editor and displays the **Rule Sets** page.

After using the New Discovery wizard to generate and import a new rule flow, all of the rule set versions in the rule flow will be unlocked, latest versions. When you publish the rule flow, SAS Decision Manager automatically locks any unlocked rule sets. See "Lock a Rule Set Version" on page 97 and Step 11 in "Create a Rule Flow Using the Rule Flow Editor" on page 103 for more information.

Table 8.1 *Setup Options and Terms for Discovery Techniques*

Discovery Technique	Setup Variables	Action Terms
Decision Tree	**Maximum number of rules**: Select the maximum number of rules that you want to be generated from the discovery analysis.	Select the terms whose values you want to predict, and click ➕ to move them to the **Actions** list.

Discovery Technique	Setup Variables	Action Terms
Scorecard	**Minimum points**: The scorecard points are scaled with this option as the minimum value. You can specify any nonnegative integer. **Maximum points**: The scorecard points are scaled with this option as the maximum value. You can specify any positive integer that is greater than the **Minimum points** value.	**Target variable**: specifies the variable that you are modeling. The variable must have exactly two discrete values such as `0` and `1` or `True` and `False`. **Target category**: specifies how the values of the target variable are mapped. The scorecard points are scaled to the likelihood of the two target variable values based on the sort order. Select **High** to indicate that the highest lexical value of the target variable is mapped to the **Maximum points** value. Select **Low** to indicate that the lowest lexical value of the target variable is mapped to the **Maximum points** value.
Recency Frequency Monetary	Select the binning method. **Independent**: Simple ranks are assigned to recency, frequency, and monetary values. The three ranks are assigned independently. **Nested**: A simple rank is assigned to recency values. Within each recency rank, customers are then assigned a frequency rank. Within each frequency rank, customer are assigned a monetary rank.	**Customer ID**: specifies a numeric or character term that uniquely identifies a customer. **Transaction date**: specifies the transaction date. **Transaction amount**: specifies the transaction amount.
Market Baskets	**Maximum number of rules**: Select the maximum number of rules that you want to be generated from the discovery analysis.	**ID**: specifies the customer ID. **Item**: specifies the item that was purchased. Each value for the item must follow the rules for valid names according to the VALIDVARNAME=V7 system option.

Open Rule Sets from the Rule Flow Editor

You can open a rule flow and some or all of its rule sets in the same layout. In the rule flow editor, either double-click on the rule sets that you want to open, or select the rule sets and click .

Add Attachments to a Rule Flow

To add an attachment such as a document file or an image file:

1. Select the **Attachments** page.
2. Click +, and select the attachment file.

   Note: You cannot attach executable files to rule sets, rule flows, or data tables. Depending on your site's settings, you might be able to attach executable files to

models, projects, and portfolios. For more information, see "Configure Support for Executable Attachments" in *SAS Decision Manager: Administrator's Guide*.

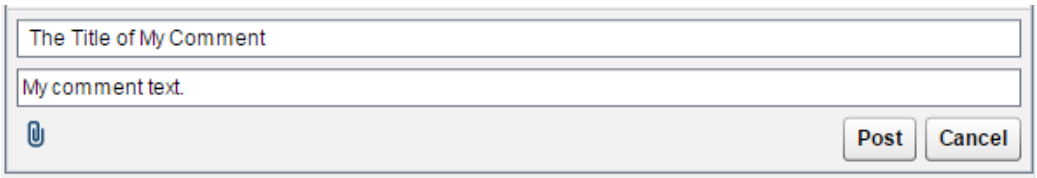
3. Click .

Note: You can delete an attachment by selecting the attachment and clicking .

Add Comments to a Rule Flow

You can add new comments or reply to existing comments. To add a new comment:

1. Select the **Comments** page.
2. Enter a topic title and enter the comment. The topic title is required, and the field for comments does not appear until you enter the topic title.



3. (Optional) Click to add an attachment such as an image or a document.

   Note: You cannot attach executable files to a comment.

4. Click **Post**.

To reply to an existing comment, enter your reply in the field immediately below the topic title for the existing comment, and click **Post**.

Click to see comments that have been posted by others.

To search for text in the comments, enter text in the search field at the top of the **Comments** page.

Change the Order of the Rule Sets

You can change the order of rule sets within the same section only (Initial, Main, Final, and so on). To reorder the rule sets in a rule flow, select the rule set that you want to move, and click or .

View the Terms Used in a Rule Flow

To view the input or output terms that are used in all of the rule sets in a rule flow, open the rule flow, and select the **Variables** page.

To view all of the terms that are used in single rule set, open the rule flow, select the rule set, and click .

Note: This icon is unavailable if you have made editing changes to the rule flow. You must save the changes to the rule flow before you click on this icon.

Managing Versions of a Rule Flow

About Rule Flow Versions

The *current* version of a rule flow is the rule flow that has the highest version number. It is also the last version that you saved. You can edit only the latest version of a rule flow.

Only one version of a rule flow can be unlocked at a time.

When you publish the current version of a rule flow, that version is locked and assigned a version number. A new current version is created. You cannot unlock a rule flow.

Set the Displayed Version

On the **Versions** page, indicates the displayed version. The displayed version is the rule flow whose information is displayed on all other pages, except the **Properties** page. The **Properties** page displays information for the entire rule flow, not for a specific version of the rule flow.

To change the displayed version, select the version that you want to view, and click .

Edit a Version Description

To edit a version description:

1. Select the **Versions** page.
2. Select the version of the rule flow that you want to edit.
3. Click . The Edit Version window appears.
4. Edit the version description.
5. Click **OK**.

Rename a Rule Flow

To rename a rule flow:

1. Close the rule flow if it is open. You cannot rename a rule flow if it is open.
2. Right-click on the rule flow and select **Rename**.
3. Change the name and click **OK**.

Duplicate Rule Flows

To duplicate a single rule flow:

1. Right-click on the rule flow, and select **Duplicate**. The Duplicate window appears.
2. Enter the name for the duplicate rule flow.
3. (Optional) Enter a description for the rule flow.
4. Select the folder where you want save the duplicate rule flow.
5. Select the version of the rule flow that you want to duplicate.
6. Click **OK**. If you do not enter a new name and the folder in which you save the duplicate rule flow already has a rule flow with the same name, SAS Decision Manager appends an underscore and a number to the name.

To duplicate multiple rule flows:

1. Select the rule flows that you want to duplicate.
2. Right-click and select **Duplicate**. The Choose a Location window appears.
3. Select the location where you want the save the duplicate tables and click **Duplicate**.

   SAS Decision Manager duplicates the current version of the selected rule flows.

   If you do not enter a new name and the folder in which you save the duplicate rule flow already has a rule flow with the same name, SAS Decision Manager appends an underscore and a number to the name.

Move Rule Flows

You cannot move a rule flow if it is open. To move rule flows:

1. Select the rule flows that you want to move, and then right-click and select **Move**. The Choose a Location window appears.
2. Select a new location for the rule flows, and click **Move**.

Remove Rule Sets from a Rule Flow

To remove rule sets from a rule flow, open the rule flow, select the rule sets, and click .

Delete Rule Flows

Note: You cannot delete a rule flow if it is used in a decision.

To delete rule flows, select the rule flows and click 🗑.

Testing a Rule Flow

You can test a rule flow before you publish it. If necessary, you can specify initialization or setup code that you want to run before the rule flow is run. SAS Decision Manager reports rule flow results and test data such as rule-fired data. SAS Decision Manager saves the test results from the last time a test was run.

Input Data for Rule Flow Tests

SAS Decision Manager expects the input data for the rule flow test to already exist and to be defined as a data table. See Chapter 3, "Managing Data Tables," on page 47 for information about defining data tables. Your user ID must have permission to access the data.

Create and Run a New Rule Flow Test

To test a rule flow:

1. Open the rule flow that you want to test.
2. Select the **Tests** page.
3. Click + to add a new test. The **Add a New Test** window appears.
4. Enter a name for the new test. Test names are limited to 30 characters.
5. (Optional) Enter a description for the test.
6. Select the data source that contains the input data for the test, and click **Next**.
7. Map the terms in the rule flow to columns in the input data set. If you click **Map terms**, the application automatically maps as many terms as possible. You can also map terms by manually selecting an input column for each rule flow input term.
8. Click **Next**.
9. (Optional) Enter any SAS code, such as initialization code or setup code, that you want to run before the rule flow is run. See "Specify Preprocessing Code" on page 112 for more information.
10. Click **Run** to run the test, or click **Save** to save it without running it.

If the test completes successfully, the status on the **Tests** tab changes to ✓. SAS Decision Manager displays the **Results** tab on which you can view the output of the rule flow, analyze the rule-fired data, and view the SAS code that was generated and run by SAS Decision Manager. See "View Rule Flow Test Results" on page 113 for more information.

Rule flow tests are associated with the rule flow version. After a test completes, the test version is displayed on the **Tests** page.

Run a Rule Flow Test

To run a rule flow test:

1. Open the rule flow that you want to test.

2. Select the **Tests** page.
3. Select the test that you want to run and click .

Copy a Rule Flow Test

To copy a test:

1. Open the rule flow.
2. Select the **Tests** page.
3. Select the test that you want to copy and click .

Edit a Rule Flow Test

To edit a rule flow test:

1. Open the rule flow.
2. Select the **Tests** page.
3. Select the test that you want to edit and click .

Delete a Rule Flow Test

To delete a rule flow test:

1. Open the rule flow.
2. Select the **Tests** page.
3. Select the test that you want to delete and click .

Specify Preprocessing Code

To specify code that you want to run before the rule flow is executed, enter the code during the **Preprocessing** step in the **Add a New Test** window.

You can use the &BRM_CODE_TYPE macro variable to specify whether SAS Decision Manager generates DS1 code for the rule flow test. See "Generating DATA Step Code for a Rule Flow" on page 113 for more information.

You can use the &DCM_USE_LATEST_VERSION macro variable to ensure that when a rule flow is run, the latest compatible version that is always used. See "Dynamically Running the Latest Rule Flow Version" on page 113 for more information.

You can use the &DCM_DS2_OPTIONS macro variable to specify DS2 options to be included in the code that is generated for a rule flow. For example, you can specify that missing values generate notes instead of error messages by defining the macro variable as follows:

```
%LET DCM_DS2_OPTIONS=STR%(MISSING_NOTE)
```

For more information, see "DS2_OPTIONS Statement" in *SAS DS2 Language Reference*.

View Rule Flow Test Results

When you test a rule flow, SAS Decision Manager displays the output of the rule flow together with other information on the **Results** tab. You can filter the rows that are displayed on the **Output Table** tab by clicking above the output table. On the **Rules Fired Analysis** tab, if you select a record in the **Output Records** table, SAS Decision Manager displays the rules that fired for that record in the table at the bottom of the page. See also "When Are Output Records Generated?" on page 115.

Note: The _recordCorrelationKey column in the output table is a unique key that is added to each output record. This key enables the output records to be correlated with the records in the rule-fired details table. See "Columns in the Rule-Fired Details Table" on page 336.

Generating DATA Step Code for a Rule Flow

You can generate DATA step (DS1) code or DS2 code for a rule flow. In many cases, you will get better performance by specifying DS1. However, consider specifying DS2 if your input data is in Teradata, Greenplum, or Hadoop, and you have installed the SAS Code Accelerator. In addition, the LIKE operator is not supported for DS1.

The **brm.runtime.codetype** configuration property in SAS Management Console determines whether SAS Decision Manager generates DS1 code or DS2 code during rule flow testing and when a published rule flow is run. By default, this property is set to DS2. You can change this property to specify DS1.See "Business Rules Manager Web Advanced Properties" in *SAS Decision Manager: Administrator's Guide* for more information.

To generate DS1 code for a specific rule flow regardless of the setting of the **brm.runtime.codetype** property, you can specify the &BRM_CODE_TYPE macro variable in preprocessing code. Define this variable in preprocessing code such as in the **Preprocessing Code** section of a rule flow test or in the **Precode** section of the **Precode and Postcode** tab in SAS Data Integration Studio. Define this variable before calling the %BRM_RULE_FLOW macro:

```
%let BRM_CODE_TYPE=DS1;
```

Using this macro variable helps you determine whether you want to change the **brm.runtime.codetype** setting.

Dynamically Running the Latest Rule Flow Version

You can use the &DCM_USE_LATEST_VERSION macro variable and either the &DCM_RULEFLOW_NAME or &DCM_DEPLOYED_RULEFLOW_NAME macro variable to ensure that when a rule flow is run, the latest version of the rule flow is always used. If you specify both &DCM_RULEFLOW_NAME and &DCM_DEPLOYED_RULEFLOW_NAME, then the name specified by &DCM_DEPLOYED_RULEFLOW_NAME is used.

For &DCM_DEPLOYED_RULEFLOW_NAME, specify the name of the published rule flow and the identification number of the rule flow. You can find the published name and identification number in the **Name** column of the rule flow **History** page. For example:

```
%let DCM_DEPLOYED_RULEFLOW_NAME=published_flow_name(ID_number);
```

Note: If you specify &DCM_RULEFLOW_NAME and SAS Decision Manager finds multiple rule flows that match the specified name, it writes an error message in the SAS log, and the rule flow is not executed. If you encounter this issue, specify the specific rule flow by using &DCM_DEPLOYED_RULEFLOW_NAME.

Define these macro variables in preprocessing code such as in the **Preprocessing Code** section of a rule flow test or in the **Precode** section on the **Precode and Postcode** tab in SAS Data Integration Studio. Define these variables before calling the %BRM_RULE_FLOW macro. For example:

```
%let DCM_USE_LATEST_VERSION=Y;
%let DCM_RULEFLOW_NAME=rule_flow_name;
```

Note: SAS Data Integration Studio uses the latest version of the rule flow that matches the variable mappings in the Business Rules transformation. SAS Decision Manager writes a note in the SAS log that states which version was selected.

Running Rule Flows in SAS Data Integration Studio

Reducing Overhead

If you have previously run a rule flow using the **Location of generated debug code file** option in SAS Data Integration Studio, you can use the &BRM_USE_EXISTING_CODE macro variable to reduce overhead when you run the same version of the same rule flow again. On the **Precode and Postcode** tab in SAS Data Integration Studio, set this macro variable to **Y**:

```
%LET BRM_USE_EXISTING_CODE=Y;
```

Setting this variable eliminates variable remapping and other checks, such as determining whether the rule flow can be run in-database.

Customizing the Database Connection

Your site might have multiple environments, such as testing and production, that are using the same metadata server. In these cases, if a rule flow uses a lookup table or specifies **Use latest** for a rule set version, you might need to customize the database connection to ensure that running jobs access the database in the correct environment. To customize the connection, you can specify the following macro variables on the **Precode and Postcode** tab in SAS Data Integration Studio:

&DCM_DBMS_OVERRIDE
: specifies PostgreSQL as the database. Specify **POSTGRES**.

&DCM_DATABASE_OVERRIDE
: specifies the database name.

&DCM_SERVER_OVERRIDE
: specifies the fully qualified host name of the server on which the database is installed.

&DCM_PORT_OVERRIDE
: specifies the port that is used by the database.

&DCM_AUTH_DOMAIN_OVERRIDE
: specifies the authorization domain for the database.

For example:

```
%let dcm_dbms_override=%str(POSTGRES);
%let dcm_database_override=%str(database_name);
%let dcm_server_override=%str(server);
%let dcm_port_override=%str(port);
%let dcm_auth_domain_override=%str(dcm_auth_domain);
```

When Are Output Records Generated?

If an input record does not fire any rules, then an output record might not be created, depending on the rule flow. If an input record fires a rule, then an output record is created. In this case, the point at which output records are generated depends on the structure of the rule flow.

simple rule flow
: One output record is generated for each input record.

complex rule flow without BY-group terms
: One output record is generated for each input record. If there are rules in the Final section, an output record is also generated after the rules in the Final section run.

complex rule flow with BY-group terms
: One output record is generated for each BY-group. This output record is generated after the rules in the Group End section run. An additional output record is generated after the rules in the Final section run.

Publish a Rule Flow

Publishing is the process of writing a business rule flow to the content server. After you publish a rule flow to the content server, other applications can use it.

When you publish the current version of a rule flow, that version of the rule flow is locked and cannot be unlocked. For more information, see "Managing Versions of a Rule Flow" on page 109.

Note: To publish a rule flow, all of the rule set versions that are referenced in the rule flow must be locked. When you publish a rule flow, SAS Decision Manager automatically locks any unlocked rule set versions that are used in the rule flow.

To publish a rule flow:

1. Open the rule flow.
2. If the rule flow contains changes that have not been saved, click . You cannot publish a rule flow if it contains changes that have not been saved.

3. Click . The Choose a Location window appears.

   Note: If a rule flow has already been published, SAS Decision Manager always publishes the rule flow to the same location. It does not prompt for a location the next time the rule flow is published.

4. Select the location where you want to publish the rule flow.

   Note: This window lists all of the objects that are defined in the SAS metadata folders. To limit the list to folders only, select the **Show folders only** check box.

   Note: In the Choose a Location window, to create a new subfolder, click .

5. Click **OK**.

Display Publish Information for Rule Flows

Publish information for a specific version of a rule flow is available on the **Versions** page. The information available includes the published rule flow name, the folder path to which the rule flow was published, the date on which the version was published, and the display name or user ID of the user that published the rule flow. To display publish information for a rule flow:

1. Open the rule flow.
2. Select the **Versions** page.
3. Click **Details** for the version that you are interested in.

Deploy a Rule Flow as a Stored Process

A stored process is a SAS program that is stored on a server and defined in metadata, and which can be executed as requested by client applications. When you deploy a rule flow as a stored process, the rule flow is made available as a stored process on the SAS Stored Process Server.

Note: To deploy a rule flow, all of the rule set versions that are referenced in the rule flow must be locked. When you deploy a rule flow, SAS Decision Manager automatically locks any unlocked rule set versions that are used in the rule flow.

Note: Only simple rule flows can be deployed as stored processes.

To deploy a rule flow as a stored process:

1. Open the rule flow.
2. If the rule flow contains changes that have not been saved, click . You cannot deploy a rule flow if it contains changes that have not been saved.
3. Click . The Choose a Location window appears.
4. Select the location where you want to deploy the rule flow.

   TIP This window lists all of the objects that are defined in the SAS metadata folders. To limit the list to only folders, select the **Show folders only** check box.

   Note: To create a new subfolder in the Choose a Location window, click .

5. Click **OK**.

For more information about stored processes, see *SAS Stored Processes: Developer's Guide*.

Viewing Lineage Information for a Rule Flow

About Lineage Information

The lineage viewer is provided by SAS Lineage. The relationship information that is displayed by SAS Lineage is taken from the Relationship database that is a part of the SAS Web Infrastructure Platform Data Server. SAS Lineage can display most types of SAS metadata. This data includes models, rule flows, and data objects, including columns, tables, external files, stored processes, and more.

SAS Lineage displays three types of diagrams:

- a network diagram that displays all relationships
- a dependency diagram that displays governance information
- a dependency diagram that displays parent and child relationships

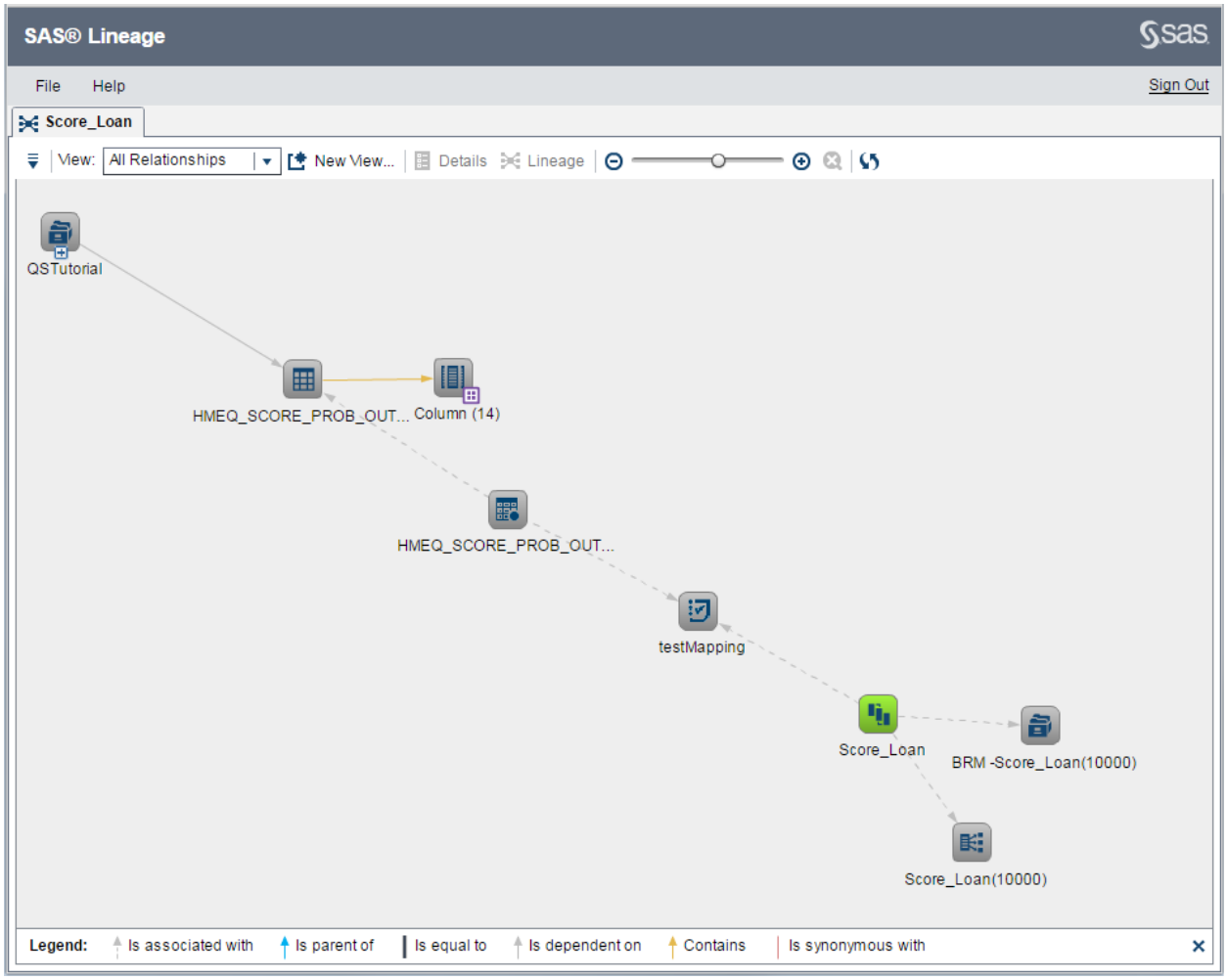
View Lineage Information for a Rule Flow

To view lineage information:

1. Open the rule flow and click . If you are not already signed in to SAS Lineage, you are prompted to sign in.
2. Enter your user ID and password, and click **SIGN IN**. SAS Lineage displays the All Relationships diagram for the rule flow.

For additional information about SAS Lineage, click **Help** or see *SAS Lineage: User's Guide*.

The following image shows the All Relationships diagram for a simple rule flow with one rule set.



Part 4

Working with Models

Chapter 9

Managing Folders for Model Projects and Portfolios

Overview of Managing Folders

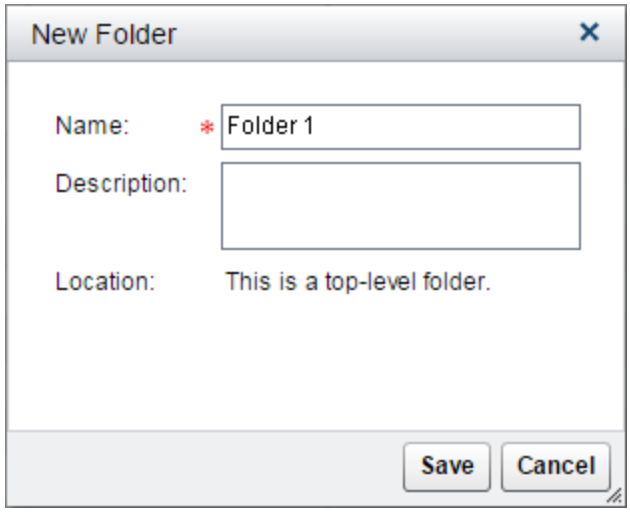
In the Projects and Portfolios category views, you can add, delete, and archive or restore folders. You must create a folder before you can create a project or portfolio. You can create subfolders within a top-level folder to organize your projects and portfolios. Designing a folder structure enables you to get summary information about the contents of the folder. This information includes the number of models, versions, and scoring tests, as well as reports for the models that contain model variables and target variables. Your folder structure could be similar to your business departmental hierarchy, or it could list individual project, portfolio, or model names.

To view the summary information, select a folder and then select **Actions** ⇨ **View Summary**.

Create a New Folder

Before you add new projects or portfolios to manage models, you must add folders to store them in.

1. Click and select **New Folder** or **New Top-Level Folder**. The New Folder window appears.



2. Enter a name for the folder.
3. (Optional) Enter a description for the folder.
4. Click **Save**.

Note: Alternatively, you can right-click an item and select the menu option for the action that you want to perform.

Rename a Folder

To rename a folder, right-click the folder, and select **Rename**. Enter the new name, and click **OK**. Folder names are case sensitive. SAS Decision Manager considers **`myfolder`** and **`MYFOLDER`** to be two unique folders.

Alternatively, click on the folder and select **Rename** from the **Actions** menu.

Delete a Folder

To delete a folder, right-click the folder, and select **Delete**. Click **OK** in the warning message.

Alternatively, click on the folder and then click .

Archive and Restore Folders

In the Projects and Portfolios category views a folder and its contents can be archived and restored to a different system.

Using the archive and restore facilities, a SAS Decision Manager administrator can back up a folder in one repository and restore it to another repository. The folder is archived as a compressed ZIP file.

Note: The attachments and comments at the portfolio level, project level, and model level are managed by separate services and are not stored within the portfolio, project, and model objects. Therefore, the attachments and comments are not included in the archived ZIP file.

Before you restore a folder, you should first create a folder to restore it to, since the restored projects reside at the same level that you specified. A best practice is to give the restored folder the same name as the archived ZIP file. The contents of the archived folder are restored to the new folder.

Note: All tables that are referenced within the projects and portfolios that are restored must be registered in the SAS Metadata Repository and made available to the **Data** ⇨ **Tables** category view. For more information, see Chapter 3, "Managing Data Tables," on page 47.

Folders cannot be restored in these situations:

- The name of the organizational folder to be restored is the same as a project name in the archived folder.
- The same archived ZIP file has already been restored in a folder on the same WebDAV server.

To archive a folder:

1. Select a folder.
2. Select **Actions** ⇨ **Archive**.
3. Select a folder where the contents are to be saved.
4. Enter a name.
5. Click **Save**.

To restore a folder:

TIP Create a folder first into which to place the restored project.

1. Select a folder.
2. Select **Actions** ⇨ **Restore**.
3. Navigate to the folder where the contents are saved.
4. Select the file.
5. Click **OK**.

Note: After restoring a folder, verify that the following requirements have been met:

- All scheduled jobs for scoring tests, performance, and retraining for the projects within a folder must be deleted and re-created on the system where the folder was restored.
- All user-defined templates for a model, report, and properties must exist on the system where the folders were restored. If they do not exist you must re-create the templates.

For more information, see the following topics:

- "Schedule a Scoring Test" on page 197
- "Schedule a Retrain Definition" on page 280
- "Schedule Performance Definitions" on page 262
- "Creating and Managing Templates" on page 138

Chapter 10

Working with Projects

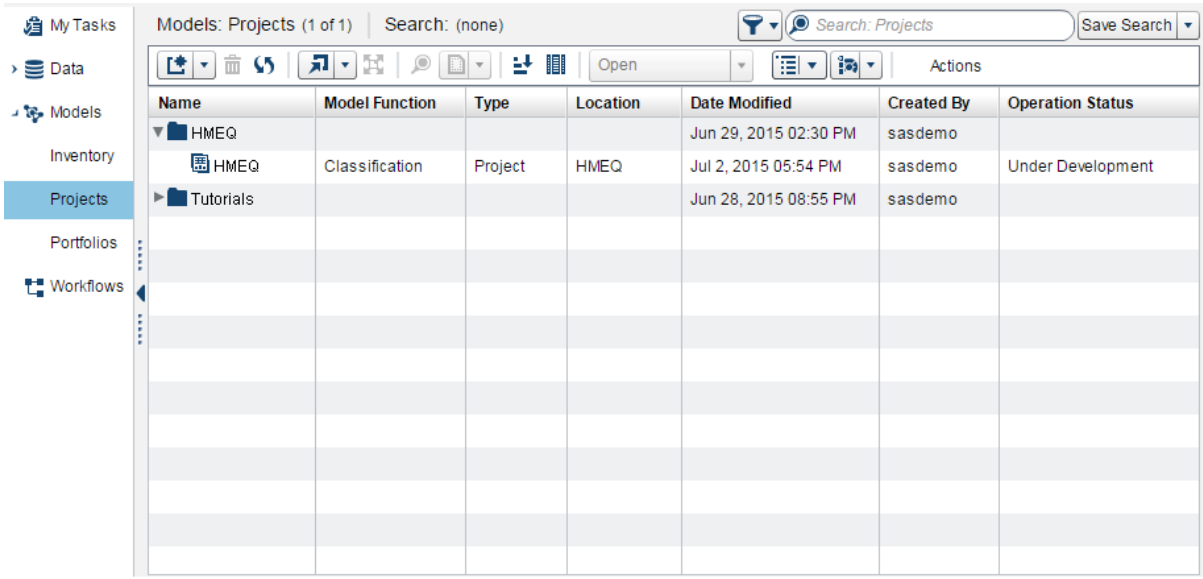
Overview of Projects

A model project consists of the models, variables, reports, performance results, and other resources that you use to determine a champion model. For example, a banking project might include models, data, and reports that are used to determine the champion model for a home equity scoring application. The home equity scoring application predicts whether a bank customer is an acceptable risk for granting a home equity loan.

You create projects within folders. The models within a project are associated with a version. A *version* is used to organize project content and model information for a specific time period.

Note: If you see a folder named **FactoryMiner** it is being used in the Portfolios category to store portfolios that contain project segments and models from SAS Factory Miner. Do not remove this folder.

Figure 10.1 *Model Projects Category*



Planning a Project

Before you begin a project, you must plan your project resources. Here is a list of questions to consider and conditions to meet for a modeling project:

- After you know which users are assigned to a project, an administrator must ensure that the user is assigned to the appropriate user group and role. For more information, see "Configuring Users, Groups, and Roles" in *SAS Decision Manager: Administrator's Guide*.
- How do you want to structure your project? A project is stored in a folder that can contain multiple levels so that you can customize the structure. For example, your project folder could be similar to your business departmental hierarchy or it could list

individual project names. For more information, see "Overview of Managing Folders" on page 121.

- What models do you want to use in the project? If the models were created using SAS Enterprise Miner, SAS/STAT, or the SAS/ETS procedures COUNTREG and SEVERITY, all model components are available when you import the model. If your model is a SAS code model that is not contained in a miningresult.spk file or a model that was created by third-party software such as R, you must ensure that you have imported all of the model component files. For more information, see "Overview of Importing Models" on page 145.
- How do you want to define your project input and output variables? When you create a project, you can import the variables using input and output prototype tables, copy the variables from an existing champion model, or define individual variables. If you use prototype tables to define the project input and output variables, the tables must be registered in the SAS Metadata Repository. For more information, see "Defining Project Input and Output Variables" on page 133.
- How do you want to track the progress of a version? The Workflows view enables you to track the progress of tasks from the version level. An authorized user can create a workflow and associate it with a version. For more information, see "Overview of Using Workflows" on page 307.
- You might have project documents that you would like to access. You can attach documents at the project or model level on the **Attachments** page. For more information, see "Add Attachments to a Project" on page 137.
- You might have comments that you would like others to see or respond to. You can add comments at the project or model level on the **Comments** page. For more information, see "Add Comments to a Project" on page 137.
- Several reports are available to help you assess candidate models. You can review the types of reports that are available and plan for which reports you want to use. Your plans might also include a custom report that you can run. For more information, see "Overview of Model Comparison, Validation, and Summary Reports" on page 202.
- After your champion model is in a production environment, you can monitor the performance of the model using your organization's performance data. For more information, see "Overview of Performance Monitoring" on page 243.
- When you define performance monitoring reports, you can set up performance index alert and warning conditions to notify users when conditions exceed the indexes. For more information, see "Performance Index Warnings and Alerts" on page 251.

Prerequisites for Creating Projects

Projects can be created only by administrators and advanced users. Ensure that users who create projects are assigned to the group **Model Manager Administrator Users** or **Model Manager Advanced Users** in SAS Management Console.

All modeling projects require that you know the model function type before you create a project. The following model function types are available:

- Classification
- Prediction
- Segmentation

- Analytical

To determine the model function type for your project, see Table 10.1 on page 130.

If you use prototype tables to define the project input and output variables, you must create the project input and output tables and register them in the SAS Metadata Repository using the Data category view or SAS Management Console. If you use SAS Management Console, you must then add the registered tables in a library by using the Data category view to make the tables available to the application.

For more information, see the following documentation:

- "Defining Project Input and Output Variables" on page 133
- Chapter 3, "Managing Data Tables," on page 47

Create a Project

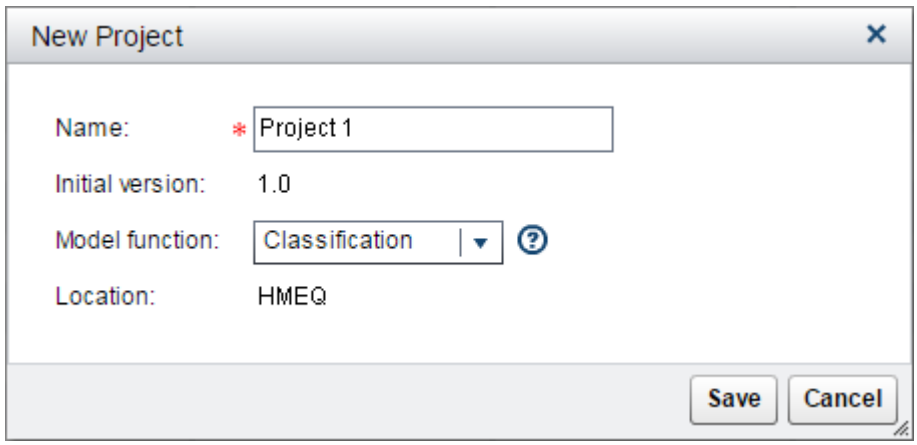
To create a project:

1. Select a folder or create a new folder in which to store the new project.

   Windows Specifics
   Short names are recommended for folders, projects, and models. Short names are essential when you perform specific actions for the model such as scoring, running reports, or monitoring performance. If the value of the **URL** model property exceeds 262 bytes, you receive an error message that the action could not be completed.

2. Click and select **New Project**. The New Project window appears.

   Note: Alternatively, you can right-click a folder and select **New Project**.



3. Enter a name for the project.

   The initial version is displayed and reflects the level for sequential versions.

4. Select a model function (Classification, Prediction, Segmentation, or Analytical) to indicate the type of models that can be imported into the project. The location of the project is displayed.

5. Click **Save**.

To delete a project, select a project and then click .

Alternatively, you can right-click an item and select the menu option (**Publish**, **Rename**, or **Delete**) for the action that you want to perform.

Project Properties

About Project Properties

Project properties contain the project metadata. Project metadata includes information such as the name of the project, the type of project, the project owner, the project identifier, the name and path of the repository, and of the tables and variables that are used by project processes.

Project properties are organized into the following types:

- General Properties
- Specific Properties
- System Properties
- User-Defined Properties

General Properties

General Properties are system-defined properties that you cannot modify, with the following exceptions: folder description, operation status, and lock status.

Property Name	Description
Model function	Specifies the type of output that your predictive model project generates. The **Model function** property that you specify affects the model templates that are provided when you are ready to import models into a project. After it has been declared, the **Model function** property for a project cannot be changed. Ensure that the types of models that you are going to use in the project fit within the selected model function type. For more information about the types of model functions, see Table 10.1 on page 130.

Property Name	Description
Operation status	Specifies the current state of the project: Under Development indicates that the project has started but a champion model is not yet in production. Active indicates that a champion model for this project is in production. Inactive indicates that the champion model is temporarily suspended from production. Retired indicates that the champion model for this project is no longer in production. To set the status, select an option from the **Operation status** drop-down list.
Lock project variables	Specifies that the project metadata is locked and the project definition cannot be modified. For more information, see "Lock or Unlock Project Variables" on page 138.

Table 10.1 *Types of Model Functions*

Model Function	Description	Example
Analytical	Function for any model that is not Prediction, Classification, or Segmentation.	None
Classification	Function for models that have target variables that contain binary, categorical, or ordinal values.	**DEFAULT_RISK** = {Low, Med, High}
Prediction	Function for models that have interval targets with continuous values.	The score output of a prediction model could estimate the weight of a person. The output of a model would be P_Weight.
Segmentation	Function for segmentation or clustering models.	Clustering models

Specific Properties

Specific Properties contain information about tables that are used by the project as well as various input and output variables and values that are used in scoring the models in test and production environments. This data can be added or modified after you add

variables. For more information, see "Defining Project Input and Output Variables" on page 133.

Property Name	Description
Default test table	Specifies a default SAS data set that can be used to create the New Dynamic Lift and Interval Target Variable reports.
Default scoring input table	Specifies a default SAS data set that is used as the input data table for all scoring tests within the project. If you specify a value for the **Default scoring input table** property, the value is used as the default input table in the Add a New Scoring Test window.
Default scoring output table	Specifies a default SAS data set that defines the variables to keep in the scoring results table and the scoring test output table. If you specify a value of the **Default scoring output table** property, the value is used as the default output table in the Add a New Scoring Test window.
Default performance table	Specifies the default performance table for all model performance monitoring tests within a project.
Default train table	Specifies the default train table that is used for retraining models and for the Training Summary Data Set report. The **Default train table** is also used to validate scoring functions or scoring model files when a user publishes the associated project champion model or challenger models to a database. This property is optional.
Champion version	Specifies the version that contains the champion model in a production environment.
Model function	Specifies the type of output that your predictive model project generates. The **Model function** property that you specify affects the model templates that are provided when you are ready to import models into a project. After it has been declared, the **Model function** property for a project cannot be changed. Ensure that the types of models that you use in the project fit within the selected model function type.
Training target variable	Specifies the name of the target variable that was used to train the model.
Target event value	Specifies the target variable value that defines the desired target variable event.

Property Name	Description
Class target values	For class, nominal, ordinal, or interval targets, the set of possible outcome classes, separated by commas. For example, binary class target values might be `1, 0` or `Yes`, `No`. Nominal class target values might be `Low, Medium, High`. These values are for information only.
Class target level	Specifies the class target level of binary, nominal, ordinal, or interval.
Output event probability variable	The output event probability variable name, when the **Model function** property is set to **Classification** or **Analytical**.
Output prediction variable	The output prediction variable name, when the **Model function** property is set to **Prediction** or **Analytical**.
Output segmentation variable	The output segmentation variable name, when the **Model function** property is set to **Segmentation** or **Analytical**.

System Properties

System Properties are system-defined properties (UUID, Location, and URL) that you cannot modify.

Windows Specifics

Short names are recommended for folders, projects, and models. Short names are essential when you perform specific actions for the model such as scoring, running reports, or monitoring performance. If the value of the **URL** model property exceeds 262 bytes, you receive an error message that the action could not be completed.

User-Defined Properties

You can add you own project properties under **User-Defined Properties**. The property-value pair is metadata for the project.

To create a user-defined property:

1. On the **Properties** page, select **User-Defined**.
2. Click +. The New User-Defined Property window appears.
3. Enter a name and value for the property. Do not include spaces or double-byte character sets.
4. Click **OK**.

To delete a user-defined property:

1. On the **Properties** page, select **User-Defined**.
2. Select a property.

3. Click 🗑. A confirmation window appears.
4. Click **OK** to delete the property.

Defining Project Input and Output Variables

About Defining Project Input and Output Variables

Project input and output variables are the variables that are used by the champion model and challenger models. Project input and output variables must be defined before a champion model can be published to a production environment. You can define the project input and output variables when you create a project or during the champion model selection process.

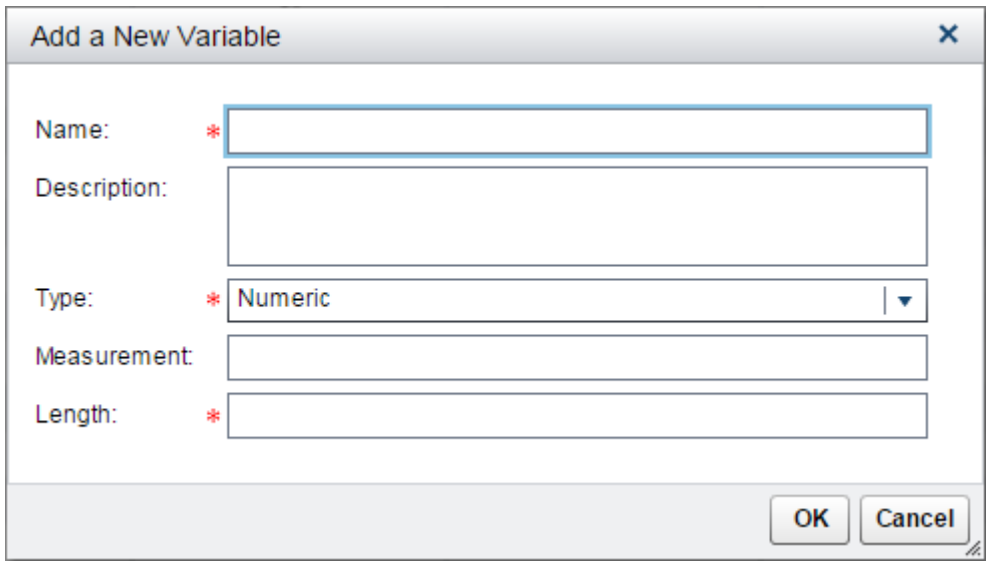
You define the project input and output variables by creating input and output prototype tables and then importing the variables using these tables, or by copying the input and output variables from another project. If you declare a champion model and the project variables have not been defined, you are prompted to add model input variables to the project and to map model output variables to project output variables.

From the **Variables** page of a project, you can add, delete, edit, copy, and import project variables.

Add a New Variable

To add a new variable:

1. Click +. The Add a New Variable window appears.



2. Enter a name.
3. (Optional) Enter a description.
4. Select a type:
   - **Numeric**

- **Character**

5. (Optional) Enter a measurement.
6. Enter a length.
7. Click **OK**.
8. Click to make the changes effective for other pages.

Delete a Variable

To delete a variable:

1. Select a variable.
2. Click . A confirmation window appears.
3. Click **OK** to delete the variable.

Edit a Variable

To edit a variable:

1. Select a variable.
2. Click .
3. Edit the necessary fields and click **OK**.
4. Click to make the changes effective for other pages.

Copy Variables

To copy variables from a project:

1. Click .
2. Select a project.
3. Click **OK**.
4. Click to make the changes effective for other pages.

Import Variables

To import variables from a table:

1. Click .
2. Select a data source.
3. Click **OK**.
4. Click to make the changes effective for other pages.

Managing Project Versions

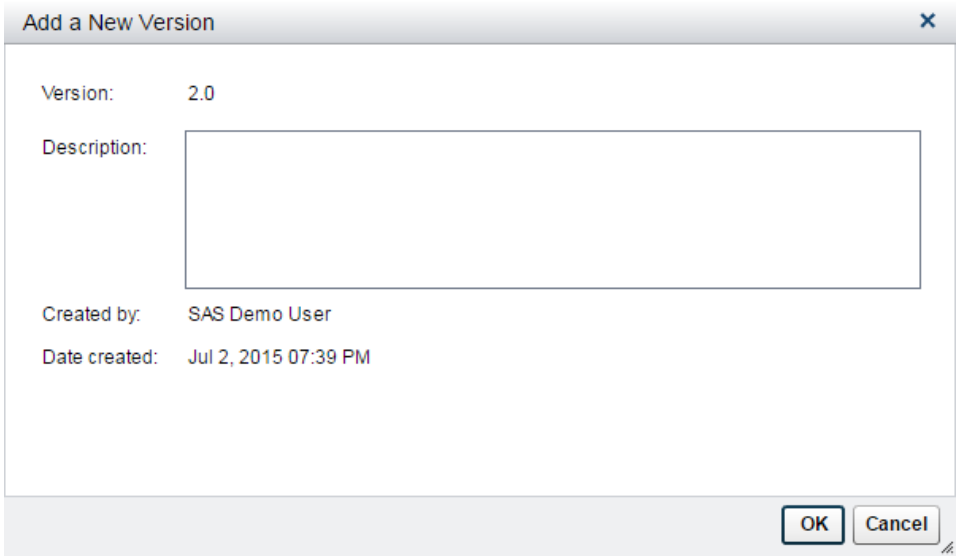
Overview of Project Versions

After a project is created, you can view information about the project on the **Versions** page. An initial version is created automatically, and it functions as a time-phased container for your projects. The version is a sequential number that increments by plus one each time you add a new version. You can also specify a description for the version when adding a new one, such as a time interval for a project cycle. Your version might represent a calendar year, a retail season, or a fiscal quarter. A project can contain multiple versions. A version contains all of the candidate model resources that you need to determine a champion model as well as all champion model resources. For example, you might develop models for a scoring program that determines whether a customer is eligible for a home equity loan. The version contains all of the models, scoring tests, and reports that are used to determine the champion model. Expand **Details** to view more information about the project version.

Create a New Version of a Project

To create a new version:

1. Select the **Versions** page.
2. Click ➕. The Add a New Version window appears.



3. The next sequential number appears as the new version number for the project.
4. (Optional) Enter a description for the version.
5. Click **OK**.

You can also delete a version if it is not the current version, or if it is locked. Select a version and click 🗑

Set the Displayed Version

To set the displayed version:

1. Select the **Versions** page.
2. Select a version and click , or double-click a version.
3. The icon indicates the version that is being displayed.

Lock and Unlock a Project Version

You can enable or disable modifications of some version models properties and files. Locking a version restricts the activities that you can do with the project. You normally lock a version after you declare a champion model in preparation for deploying the champion model to a production environment.

To lock or unlock a version:

1. Select the **Versions** page.
2. Select a version and click to lock or unlock the version. The label **Locked** after the version name indicates the version that is being locked.

Attach a Portable Formats File

The portable formats file contains the user-defined formats that are associated with the train table that was used to create a model. In order for the validation to be successful, the table that is selected at publish time to validate the model must be associated with the same user-defined formats. You must transform the user-defined formats SAS catalog into a portable formats file and attach it to the version. This action enables the user-defined formats to be published to the database with the model.

To attach a portable formats file:

1. Select the **Versions** page.
2. Select a version and click .
3. Navigate to the appropriate folder and select the portable formats file to attach to the selected version.
4. Click **OK**.

View Life Cycle Status

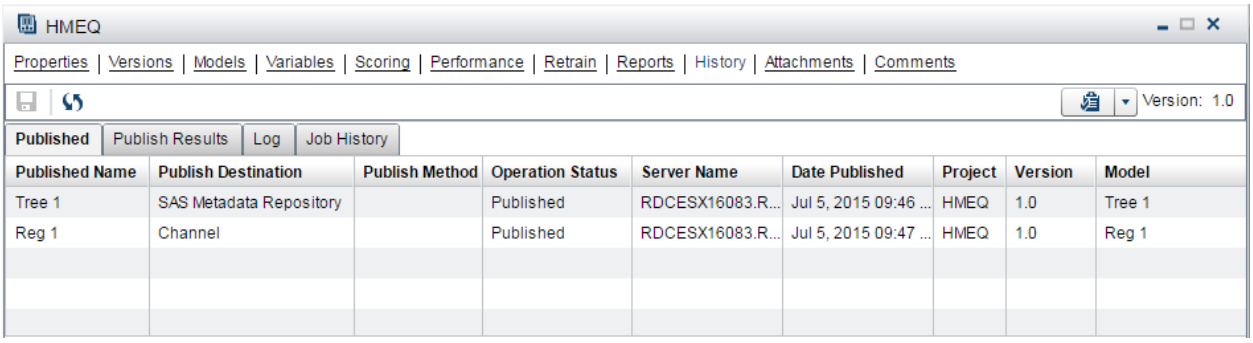
Note: Only life cycle content for migrated versions can be viewed.

To view the life cycle status:

1. On the **Versions** page of a project, select a version and expand **Details**.
2. Click **View Life Cycle Status**.
3. View the information and click **OK**.

View Project History

On the **History** page, you can view the history log for changes to the project, the history of models that were published at the project and model level, and the history of scoring, performance, and retrain jobs that were executed.



Published Name	Publish Destination	Publish Method	Operation Status	Server Name	Date Published	Project	Version	Model
Tree 1	SAS Metadata Repository		Published	RDCESX16083.R...	Jul 5, 2015 09:46 ...	HMEQ	1.0	Tree 1
Reg 1	Channel		Published	RDCESX16083.R...	Jul 5, 2015 09:47 ...	HMEQ	1.0	Reg 1

Add Attachments to a Project

On the **Attachments** page, you can view and add attachments such as images or documents. All new attachments are associated with the project. Values in the version and location columns appear only for attachments that were migrated from a previous release of SAS Decision Manager. The version and location columns also appear for performance and training summary data sets that are associated with the selected version. The value for location is the directory path where the attachment is stored in the model repository. Attachments for versions within a project that were migrated now appear at the project level.

To add an attachment:

1. Click ✚.
2. Select a file to attach and click **Open**.

   Note: Depending on your site's settings, you might be able to attach executable files to models, projects, and portfolios. For more information, see "Configure Support for Executable Attachments" in *SAS Decision Manager: Administrator's Guide*.

Note: Click ✕ to remove an attachment.

Add Comments to a Project

On the **Comments** page, you can add new topics or respond to an existing topic. You can also search the comments.

To add a comment:

1. Enter a topic name and a comment.
2. (Optional) Click to attach a file to the new topic. Repeat this step to attach multiple files.

   Note: You cannot attach executable files to a topic.

   Note: You can also click **Remove** to remove an attachment.
3. Click **Post**.

Lock or Unlock Project Variables

You cannot modify project variables that are locked for a project. Also, you cannot set a new champion or challenger model for the project.

To lock or unlock a project:

1. In the Projects category view, select a project.
2. Select **Actions** ⇨ **Lock Project Variables**. Note that the **Lock project variables** check box is selected on the Properties page of the project.
3. To unlock the project, select **Actions** ⇨ **Lock Project Variables**. Note that the **Lock project variables** check box is deselected on the Properties page of the project.

   Note: You can also select or deselect the **Lock project variables** check box on the **Properties** page of a project.

Creating and Managing Templates

About Creating and Managing Templates

There are three different types of templates that you can create or edit. The template types are Model, Report, and Properties. Models are associated with a specific model template. A model template contains properties and component files that define a type of model. Report templates can be used to create user-defined reports. A report template contains report requirements such as report name and the number of required models to run the report. The properties template contains user-defined properties and values for the model and project object types. In the Projects category view, you can create a new template or manage existing templates.

Note: Only users who are in the Model Manager Administrator Users or Model Manager Advanced Users groups and who have Write permission to the WebDAV folder where the user-defined model and report templates are stored can save a new template or save changes to an existing template. For more information, see "Verify WebDAV Folder Permissions for User-Defined Templates " in *SAS Decision Manager: Administrator's Guide*.

For more information about the different types of templates, see the following topics:

- "Model Templates" on page 339

- "Report Templates"
- "User-Defined Properties Template"

Manage Templates

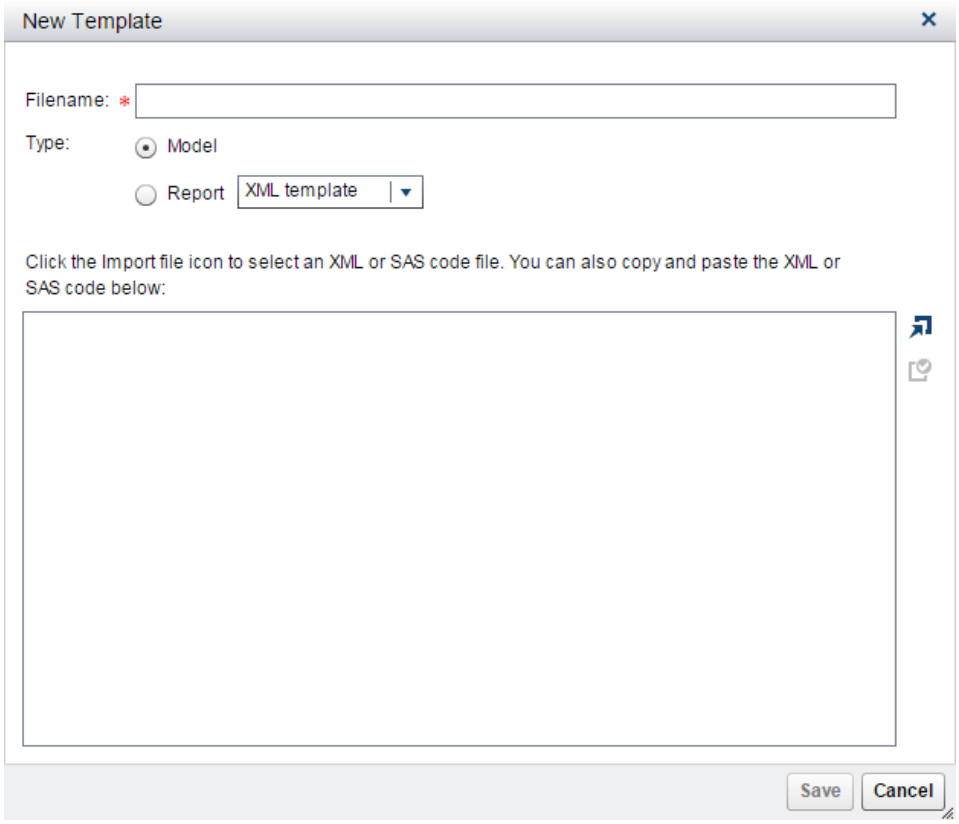
To manage templates:

1. Click and select **Manage Templates**.
2. Select an XML template or SAS code file to edit or delete. The Reserved column must be marked as No in order for the template to be editable. Life cycle templates cannot be edited but can be viewed as Read-only.
   - To edit a file, click . Make the appropriate changes and click **Save**.
   - To delete a file, click . Click **Yes**.
3. Click **Close**.

Create a New Template

To add a new template:

1. Click and select **New Template**.



2. Enter a filename.
3. Select a type:
   - **Model**

- **Report** (XML template or SAS code)

4. Click to select an XML or SAS code file. You can also copy and paste the XML or SAS code in the text box.

   Note: Ensure that the selected template type matches the XML content type before importing the file.

5. Click to validate the XML.

6. Click **Save**.

User-Defined Properties Template

When you add a user-defined property using the UserDefinedProperties.xml file, you specify the name of the property, the initial value of the property, and the type of object in the model repository to which it applies. The user-defined property is created for the specified object type when the object is added to the model repository. For example, if the XML file specifies a user-defined property *Due date* for an object type of project, the project object has a property of **Due date** each time a project is created in the Projects category. User-supplied properties are not added to existing objects in the model repository.

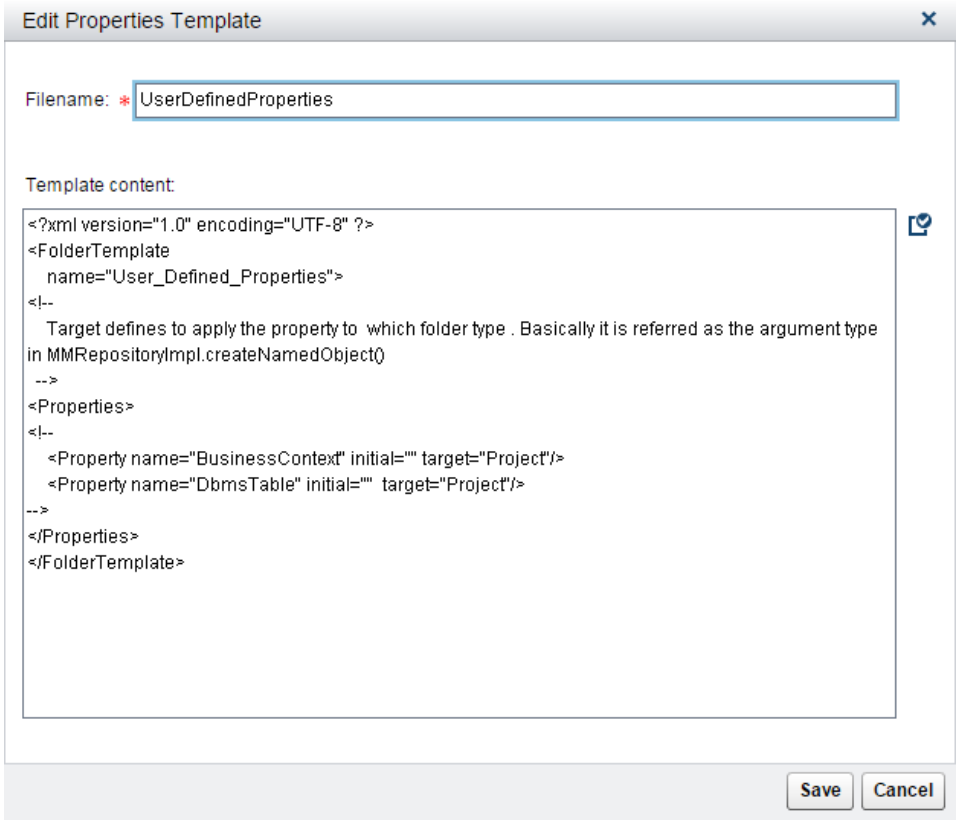
You can specify these object types in the UserDefinedProperties.xml file:

- AnalyticalModel
- ClassificationModel
- ClusteringModel
- PredictionModel
- Project

To add user-defined properties for an object type:

1. Click and select **Manage Templates**.

2. Select the `UserDefinedProperties.xml` template and click .



3. Add properties using an XML property element for each property. These arguments are required:

   name="*property-name*"
   specifies the name of the property.

   initial="*initial-value*"
   specifies a value for the property when it is added as a property for the specified object type in the model repository. If you do not want to specify an initial value, use two double quotation marks, **`initial=""`**

   target="*object-type*"
   specifies the object type in the model repository for which the user-supplied value applies.

   Example: **`<Property name="Due date" initial=" " target="Project"/>`**

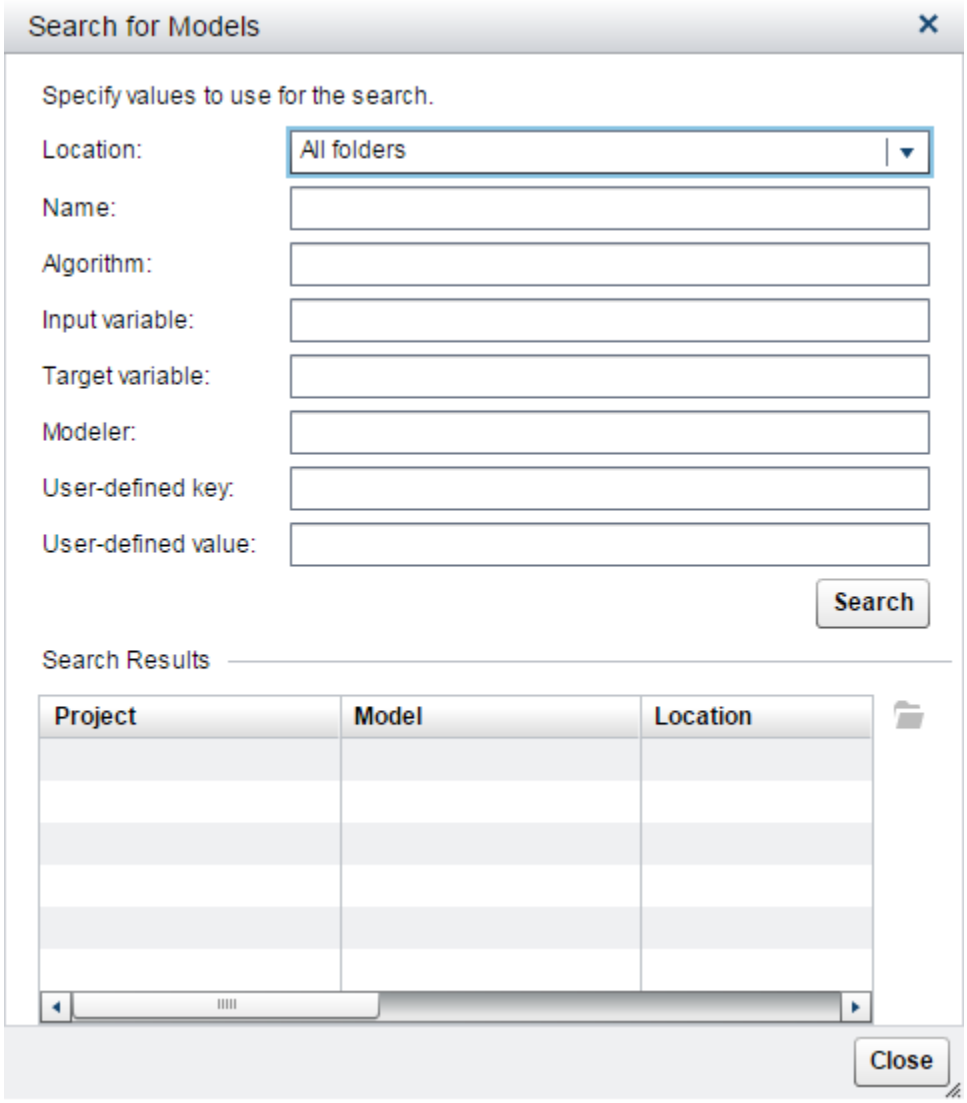
4. Click **Save**.
5. Click **Close**.

Searching for Models

You can search for models based on certain criteria in the Projects and Portfolios category views. The results appear below the search criteria. You can also search all models in the model repository and filter the results in the Inventory category by using the category search options. For more information, see "Search and Filter Inventory of Models" on page 169.

To search for models:

1. Click The default is to search **All folders**



2. Select a location:
   - **All folders** searches all folders in the category view.
   - **Current** searches only the selected object (folder, project, or portfolio) and its contents.
3. Enter a name for the model.
4. Enter an algorithm.
5. Enter an input variable. The field is case sensitive.
6. Enter a target variable. The field is case sensitive.
7. Enter a modeler.
8. Enter a user-defined key or value. The user-defined key field is case sensitive.
9. Click **Search**.

10. Select a model from **Search Results** and click or double-click to open the model. You can view or edit the model. Click **OK**.
11. Click **Close**.

The search results display the following information:

Column	Description
Project	Specifies the name of the project.
Model	Specifies the name of the model.
Location	Specifies the location of the model.
Algorithm	Specifies the name of the algorithm, such as regression or logistic, that is used by the model.
Type	Specifies one of the model function types: • Analytical • Classification • Prediction • Cluster

Chapter 11

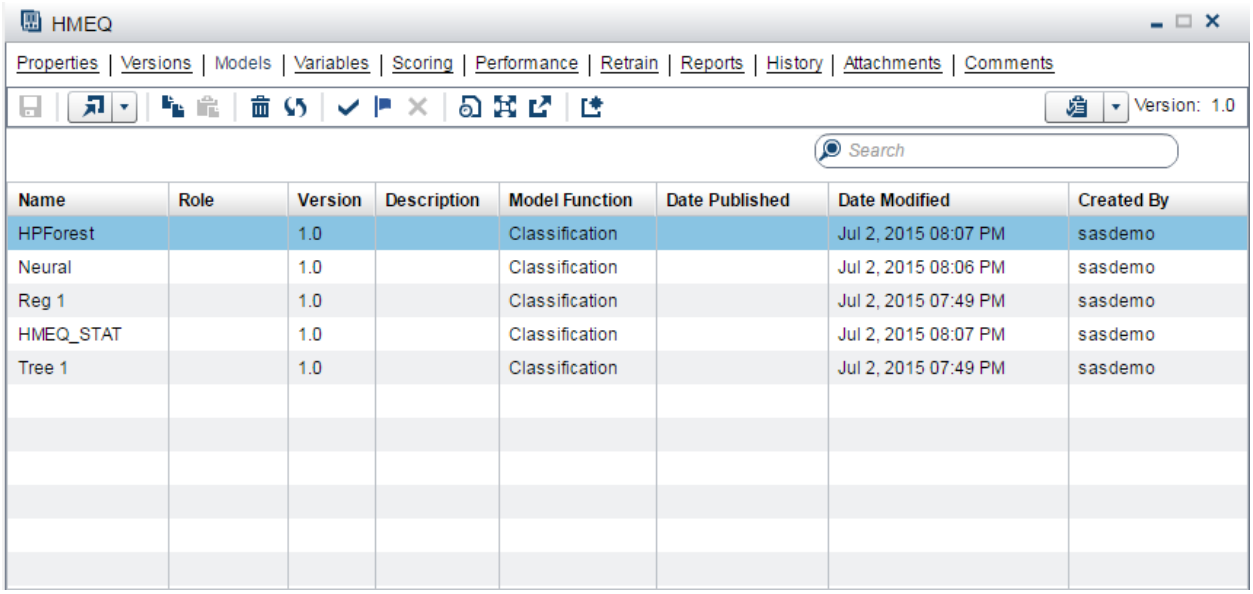
Importing Models

Overview of Importing Models

After you create a project, you import models into a project version on the **Models** page. A project can contain multiple versions. You can see only the models for the selected version on the **Models** page. After model evaluation, you set one of the candidate models as the champion model.

You can also import generic models into a folder from the SAS Workspace Server. For more information, see "Importing Models into a Folder" on page 153.

Figure 11.1 *Models Page*



Name	Role	Version	Description	Model Function	Date Published	Date Modified	Created By
HPForest		1.0		Classification		Jul 2, 2015 08:07 PM	sasdemo
Neural		1.0		Classification		Jul 2, 2015 08:06 PM	sasdemo
Reg 1		1.0		Classification		Jul 2, 2015 07:49 PM	sasdemo
HMEQ_STAT		1.0		Classification		Jul 2, 2015 08:07 PM	sasdemo
Tree 1		1.0		Classification		Jul 2, 2015 07:49 PM	sasdemo

There are many methods of importing your SAS models into your project version:

- Import a model from the SAS Metadata Repository
- Import a model from a SAS Model Package File
- Import a model from a SAS analytic store file
- Import a model from local files
- Import a model from a PMML file
- Add model files to an existing model

SAS macros are also provided so that you can use SAS code to import or register SAS models into your project. For more information, see "Overview of Access Macros" in *SAS Model Manager: Macro Reference* and "Using Macros to Register Models Not Created by SAS Enterprise Miner" in *SAS Model Manager: Macro Reference*.

Keep the following details in mind:

- Scorecard models can be imported using the SAS Code Models local files method and the SAS Model Package File import method.
- HPFOREST procedure models can be imported using the SAS Metadata Repository import and the SAS Model Package File import. You cannot import PROC HPFOREST models using local files.
- High-Performance analytics models that are not created with SAS Enterprise Miner can be registered to the SAS Metadata Repository using the %AA_Model_Register. These models can then be imported to SAS Decision Manager by importing the models from the SAS Metadata Repository from a SAS model package file.
- Before you can import COUNTREG procedure and SEVERITY procedure models, you must create the model score code using the %MM_Countreg_Create_Scorecode macro and the %MM_Severity_Create_Scorecode macro. After the score code is generated, you can use the %MM_Model_Register macro or the local files method to import these models. For more information about the types of model component tables, see "Generating Score Code for COUNTREG Procedure Models" in *SAS Model Manager: Macro Reference*.

- SAS Decision Manager can publish to a database or Hadoop the models that are associated with the **DATA step** score code type. Models that have a score code type of **Analytic store** can be published only to Hadoop and Teradata. Models that have a score code type of **SAS Program**, **PMML**, or **DS2** cannot be published to a database.
- Model component table variable names must start with a letter or underscore and can contain letters, an underscore (_), a hyphen (-), and a period (.). Variable names that have special characters can be used only when the SAS administrator has set the VALIDVARNAME system option to ANY. For more information, see "Valid Variable Name Options" in *SAS Decision Manager: Administrator's Guide*.

Windows Specifics

Short names are recommended for folders, projects, and models. Short names are essential when you perform specific actions for the model such as scoring, running reports, or monitoring performance. If the value of the **URL** model property exceeds 262 bytes, you receive an error message that the action could not be completed.

CAUTION:

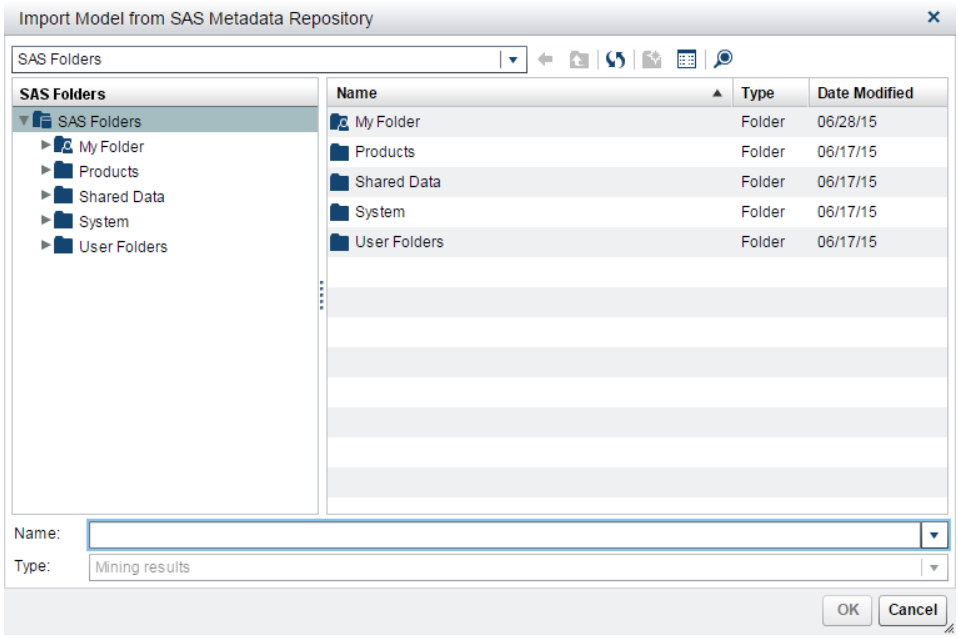
Unexpected results might occur if you import a model that was previously exported using SAS Decision Manager. A best practice is to import models that were not previously exported by SAS Decision Manager.

Import a Model from the SAS Metadata Repository

If your SAS Enterprise Miner 5.1 (or later) model files or your models that were created by the %AA_Model_Register macro are registered in your SAS Metadata Repository, you can import them into SAS Decision Manager from the repository.

To import a model from the SAS Metadata Repository:

1. Click and select **from the SAS Metadata Repository**.



2. Navigate to the location of the file and select the model file to import.
3. Enter a name for the model and click **OK**.

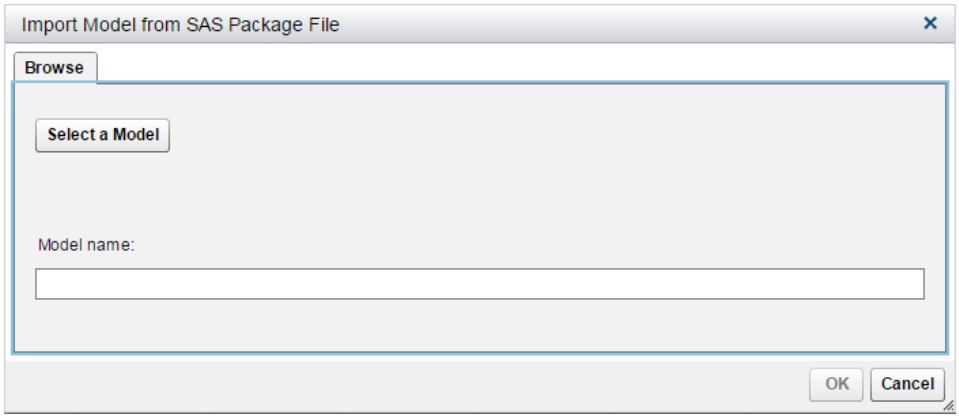
Import a Model from a SAS Package File

Import a SAS Package File

A SAS model package (SPK) file is a SAS Enterprise Miner SPK file or an SPK file that was created by using the %AA_Model_Register macro. SPK files contain complete model information. They enable you to import a complete model that is not registered in a SAS Metadata Repository.

To import a model from a SAS model package file:

1. Click and select **from a SAS package file**.



2. On the **Browse** tab, click **Select a Model** and navigate to the location of the file. Select the file to import and click **Open**.
3. Enter a name for the model.
4. Click **OK**.

Create SAS Package Files in SAS Enterprise Miner

To create SAS Package Files in SAS Enterprise Miner:

1. Open the SAS Enterprise Miner diagram that contains the model, and then run the model.
2. After the model run is complete, right-click the node in the SAS Enterprise Miner Diagram Workspace, and select **Create Model Package**. The new SPK filename appears under the Model Packages folder in your SAS Enterprise Miner Project Navigator.
3. Right-click the filename and select **Save As** to copy the SPK file from the SAS Enterprise Miner server to your computer.
4. Specify a destination folder on your computer, such as, **`C:\MMData`**, and save the file to your workstation folder.

Create SAS Package Files Using the %AA_Model_Register Macro

These models can be created by SAS procedures and are supported by SAS Decision Manager:

- SAS/STAT item store models
- High-performance models
- SAS/ETS COUNTREG procedure models
- SAS/ETS SEVERITY procedure models

You can use the %AA_Model_Register macro to create an SPK file to contain these models. For more information, see "Overview of Access Macros" in *SAS Model Manager: Macro Reference*.

Import a PMML Model

Predictive Modeling Markup Language (PMML) is an XML-based standard for representing data mining results. PMML is designed to enable the sharing and deployment of data mining results between vendor applications and across data management systems. You can import PMML models that are produced by using other applications. PMML 4.2 is supported. Models that are created using PMML 4.2 support DATA step score code.

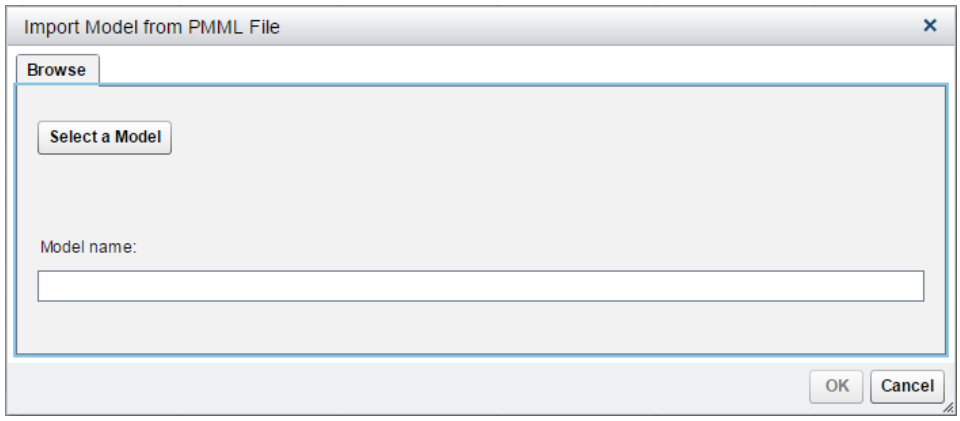
Note: If your PMML model contains variable names with blanks or special characters other than underscores, you must set the system option VALIDVARNAME=ANY to perform tasks such as scoring tests, performance monitoring, and reporting that use the score.sas file. For information about setting the VALIDVARNAME="ANY", see "Valid Variable Name Options" in *SAS Decision Manager: Administrator's Guide*.

For more information, see "PROC PSCORE and PMML Support" on page 365. If you have a license for SAS Enterprise Miner, see the topic "SAS Enterprise Miner PMML Support" in the product Help or in *SAS Enterprise Miner: Reference Help* available at http://support.sas.com/documentation/onlinedoc/miner/.

Note: PMML variables must be valid SAS variable names and cannot contain more than 20 characters. In addition, SAS Decision Manager does not support the importing of a PMML file that contains multiple models.

To import a PMML model:

1. Click  and select **from a PMML file**.



2. On the **Browse** tab, click **Select a Model** and navigate to the location of the file. Select the file to import and click **Open**.
3. Enter a name for the model.
4. Click **OK**.

Import Models from Local Files

You can import R models, and you can also import models that you created using SAS code, but that were not created in or exported from SAS Enterprise Miner. An example of a model might be a SAS LOGISTIC procedure model, a SEVERITY model, or an R logistic model. You can also add files later that were not available when the model was originally imported.

When you import models using the local file method, keep the following in mind:

- The table names that you specify as model components must start with a letter or underscore.
- Table names can contain a period.
- Table names cannot be more than 32 characters long.
- Spaces or special characters (for example, ~`!@#$%^&*()+={}[]|\\:;'<>?/") are not valid in a table name.

For more information, see Model Template Component Files on page 339.

Note: HPFOREST models cannot be imported using local files.

To use the Local Files method, you must prepare model component files. Model component files provide the metadata that is used to process a model in SAS Decision Manager. The model component files that you prepare are dependent upon the project's model function. You can find the model function in the project property **Model function**. The model functions for SAS code models are analytical, classification, prediction, or segmentation. The model functions for R models are analytical, classification, or prediction. For a list of component files by model function, see "Model Templates" on page 339. If you do not have all of the component files when you import the model, you can create them and add them later. For more information, see "Add Model Files to an Existing Model" on page 162.

SAS code models, at a minimum, require a score code component file (score.sas) and other component files to define the model input and output variables in SAS tables.

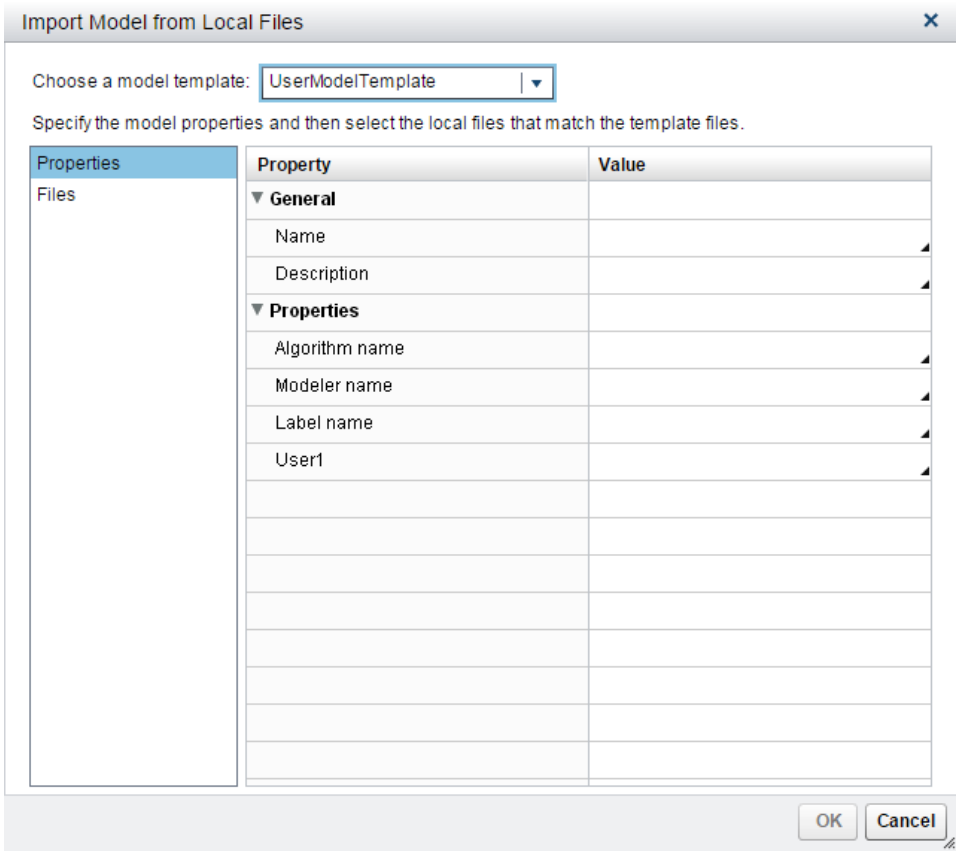
Prediction and classification models also require a component file to define target variables.

R models, at a minimum, require SAS and R score code component files, a file for the output parameter estimate, and the other component files to define the model input and output variables using either SAS data sets or XML files. Prediction and classification models also require a component file to define target variables. For more information, see "Overview of Using R Models with SAS Decision Manager" on page 367.

The score code component file (score.sas) is DATA step score code and is used as input by the SAS Scoring Accelerator when publishing a model to a database. In the scoring function publish method, some SAS language elements and syntax are not supported when you create or modify your score code. Only the SAS language elements and syntax that are required to run critical data transformations and model scoring functions are available. If you use a statement or function that is not supported, your model is not published to the database. For more information, see "Considerations When Creating or Modifying DATA Step Score Code" in *SAS In-Database Products: User's Guide*.

To import models from local files:

1. Click and select **from local files**.



2. Select a model template from the drop-down list.

   Note: If you specify values for the properties and then select a different template, the values are cleared.

3. Click **Properties** and specify the model properties.

4. Click **Files** and select the local files from the SAS Workspace Server that match the template files. You cannot delete a file once you have added it. To replace the file, select another file or cancel the import and start over.

5. Click **OK**.

Import a Model from a SAS Analytic Store File

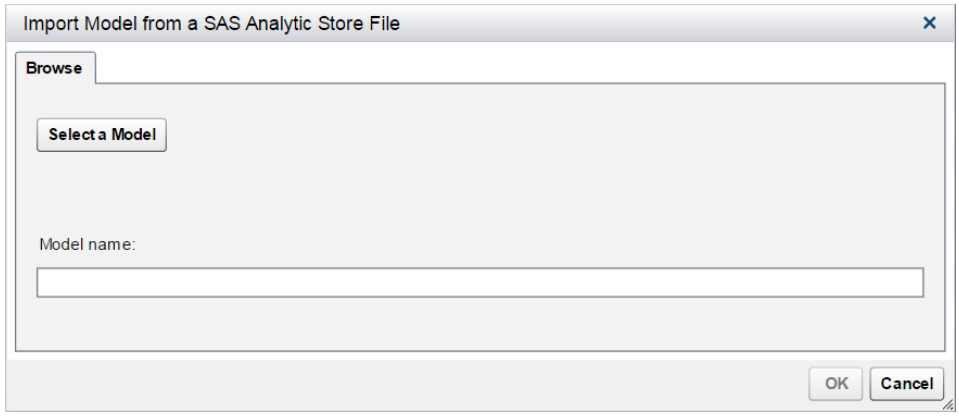
The HPFOREST and HPSVM procedures are used by SAS Factory Miner to create the SAS analytic store (SASAST) file that contains the model scoring files and model input and output variables. You can register SAS analytic store models from SAS Factory Miner, or you can register them by importing a model from a SAS analytical store file into SAS Decision Manager. SAS analytic store models that are created using SAS Enterprise Miner and that are in SAS package file (SPK) format are not supported in this release.

SAS Decision Manager supports publishing SAS analytic store models only to Hadoop and Teradata. The scoring files that are required for analytic store scoring by the SAS Scoring Accelerator are generated by the HPFOREST or HPSVM procedures. The HPFOREST and HPSVM procedures can also be used in Base SAS to create a SAS analytic store model. For more information, see "Introduction to Analytic Store Scoring" in *SAS In-Database Products: User's Guide* and *SAS Factory Miner: User's Guide*.

Note: When you are importing a model from a SAS analytic store file, the model function defaults to the value of **`Analytical`** and the file does not contain a target variable. When you are registering models from SAS Factory Miner to the model repository, the model function value of classification or prediction is accepted.

To import a model from a SAS analytic store file:

1. Click and select **Import** ⇨ **from a SAS analytic store file**.



2. On the **Browse** tab, click **Select a Model** and navigate to the location of the file. Select the file to import and click **Open**.
3. Enter a name for the model.
4. Click **OK**.

Export Models from a Project

To export an individual model from the **Models** page of a project to the SAS Workspace Server:

1. Select a model and click .
2. Select a location on the SAS Workspace Server.

   Note: You cannot export a model to a location where a model with the same name already exists.
3. Click **OK**.
4. Click **Close**.

Importing Models into a Folder

About Importing Models into a Folder

You can import an individual model or multiple models at one time from the SAS Workspace Server into a folder in the model repository. The model repository folders are located on the SAS Content Server. To access the SAS Workspace Server, map a network drive to a drive on the machine where the SAS Decision Manager Middle-Tier Server is running. All of the model files in the selected location are imported. The folder name is used as the model name. If you have subfolders, each subfolder is considered to be a separate model. Each folder's contents are considered to be model files. A folder should not contain both model files and subfolders at the same level. If the top-level folder contains both, the subfolders are ignored and only one model is added with the files that are within the selected folder.

Note: The name of the model folders that are imported from the SAS Workspace Server can contain only alphanumeric characters, spaces, an underscore (_), a hyphen (-), and a period (.).

After you import models into a folder, you can also perform the following tasks:

- export one or more models
- update one or more models
- duplicate or move models

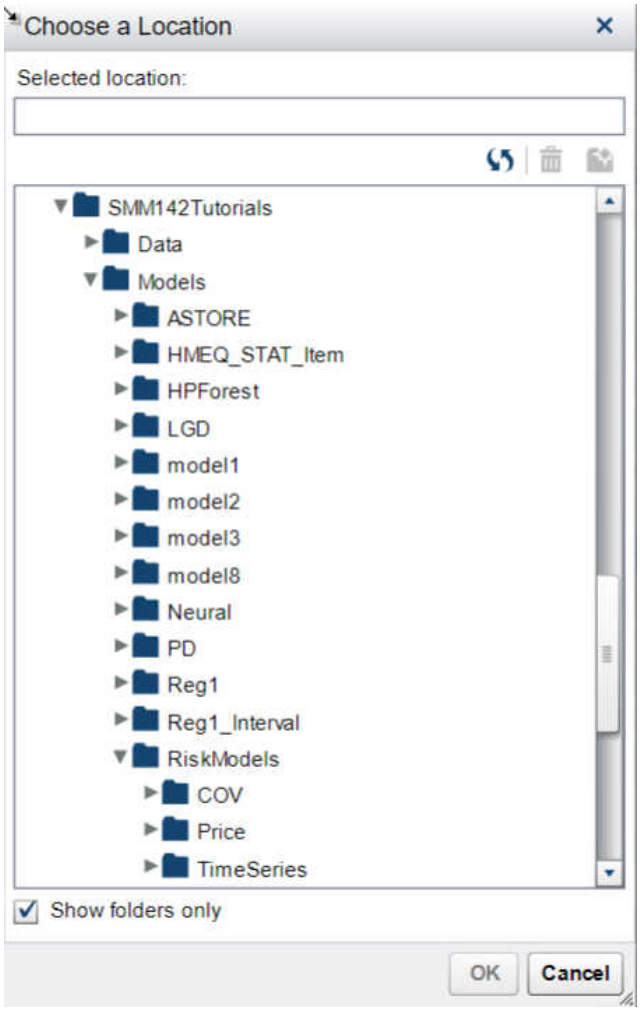
Note: These models cannot be moved or copied from a folder to a project within the user interface. When you export models from the folder level, models within a project are not included. Only models that reside directly within the selected folder are exported.

Import Models into a Folder

To import models:

1. Select a folder, click , and select **Import models**.

Note: Alternatively, right-click a folder and select **Import Models**.



2. Select the location of the model or models on the SAS Workspace Server.
3. Click **OK**.

Export Models from a Folder

You can export an individual model or multiple models at one time from a folder in the model repository to the SAS Workspace Server. The model repository folders are located on the SAS Content Server.

Note: When you export models from the folder level, models within a project are not included. Only models that reside directly within the selected folder are exported.

To export an individual model:

1. Right-click a model and select **Export**.
2. Select a location on the SAS Workspace Server.

   Note: You cannot export a model to a location where a folder with the same name already exists.
3. Click **OK**.
4. Click **Close**.

To export multiple models:

1. Right-click a folder and select **Export Models**.
2. Select a location on the SAS Workspace Server.

   Note: You cannot export a model to a location where a folder with the same name already exists.
3. Click **OK**.
4. Click **Close**.

Update Models in Folders

You can update an individual model or multiple models at one time in the model repository. Only models that have previously been exported to the SAS Workspace Server can be updated. The reason for this is that a properties file is exported with the model files. The properties file contains the UUIDs of the models. Both new and modified model files are included in the updates to the model. A new model version is created each time you update the model.

To update an individual model:

1. Right-click a model and select **Update**.
2. Select the location of the model on the SAS Workspace Server.
3. Click **OK**.
4. Click **Close**.

To update multiple models:

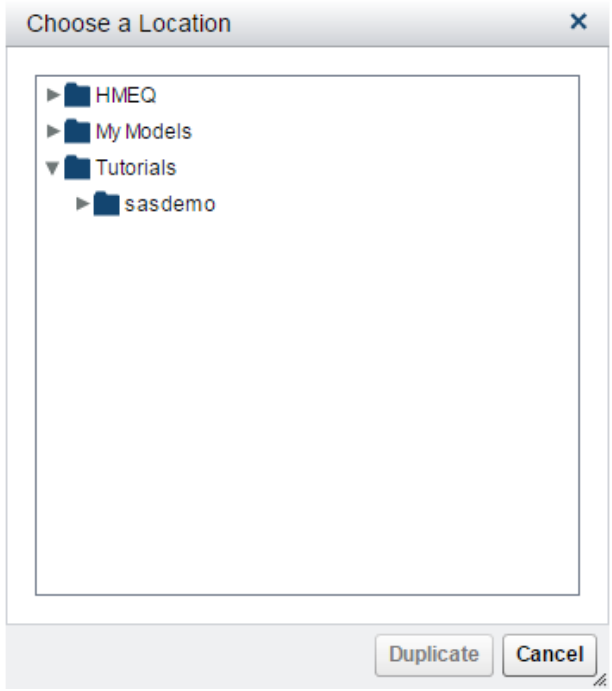
1. Click and select **Update models**.
2. Select the location of the model or models on the SAS Workspace Server.
3. Click **OK**.
4. Click **Close**.

Duplicate or Move Models

You can duplicate a model or move a model to another folder. When you duplicate a model that has multiple model versions, only the current model version is duplicated. When you move a model, all of the model versions are kept.

To duplicate a model:

1. Right-click a model and select **Duplicate**.
2. Select a location.



3. Click **Duplicate**.

To move a model:

1. Right-click a model and select **Move**.
2. Select a location.
3. Click **Move**.

Chapter 12

Managing Model Content and Versions

Overview of Managing Model Content and Versions

When you open a model, you can modify the model properties, add or view model versions, add attachments, and add comments. You can open a model from the Inventory category and from the **Models** page of a project.

Set Model Properties

After you import a model, you can specify additional property values for your imported model. On the **Model Properties** page, you can perform the following tasks:

- View the input and output variables, and create a scoring output table

- Map model output variables to project output variables
- Specify model-specific properties and user-defined properties
- View and edit score code
- View and add model files
- Create input and output variables from the score.sas file
- View the history to see a log that shows changes to the model, and to the published models

To set the model properties:

1. Select and open a model and view the **Model Properties** page. See the below table for what types of properties can be specified.

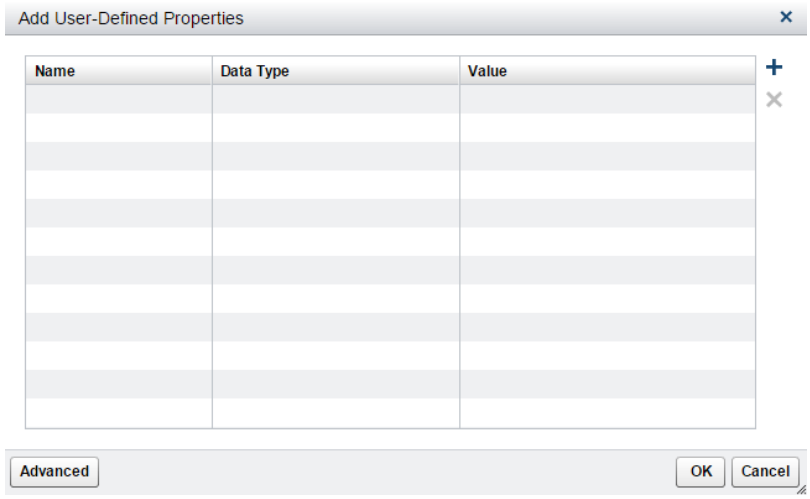
Model Properties	Description
General	On this page you can view the model name, who created it, and the dates it was created and modified. The only property that you can edit is the description. For more information, see "General Properties" on page 129.
Specific	On this page you can enter information for various items. Some values are automatically populated and cannot be modified. For editable properties, click **Browse**, enter, or select a value. For more information, see "Specific Properties" on page 349.
System	This page is a Read-only and is created after a model has been imported. The system properties for models do not require any configuration after the model is imported. For more information, see "System Properties" on page 132.
User-Defined	On this page you can view the user-defined properties for a model. You can also create user-defined properties. For more information, see "User-Defined Properties" on page 132.
Factory Miner	On this page you can view the SAS Factory Miner model properties. *Note:* This tab only appears for SAS Factory Miner models.

2. Click .

To create a scoring output table, see "Create Scoring Output Tables" on page 194

Add Model User-Defined Properties

1. On the **Model Properties** page, select **Properties** ⇨ **User-Defined**.
2. Click ➕. The Add User-Defined Properties window appears.



3. Click ➕ to insert a new row.

   Note: Alternatively, you can click **Advanced** to manually specify the name and value pairs for the new properties.

4. Specify a name and data type for the property. A value for the property is optional.

   Note: The name of a user-defined property can contain alphanumeric characters and the underscore (_). Spaces and other special characters are not allowed.

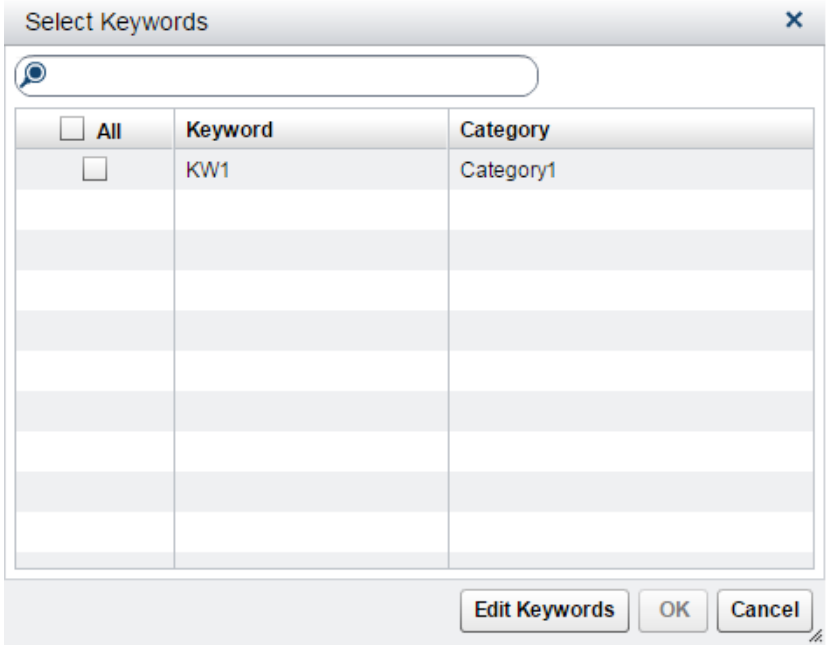
5. Click **OK**.
6. (Optional) Select a user-defined property and click to view the history of changes for that property. Click **Close**.

Add and Edit Model Keywords

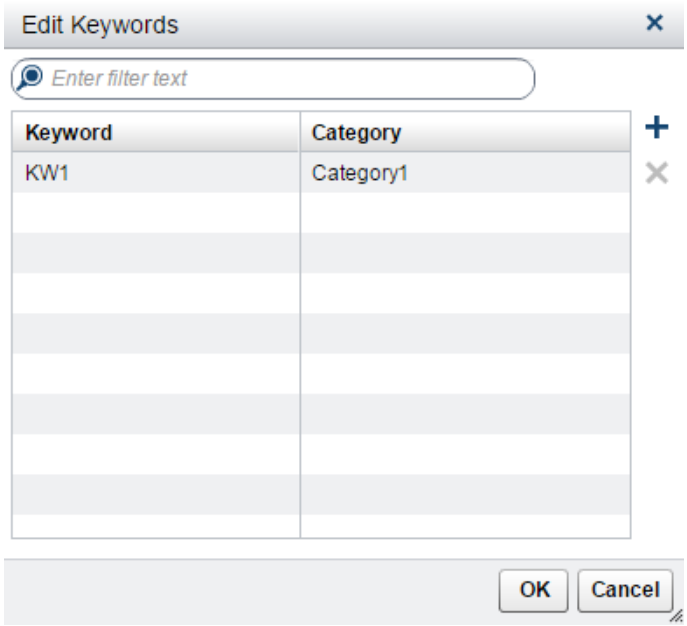
Select **Properties** ⇨ **General** to view the model keywords.

To add keywords:

1. Click ➕. The Select Keywords window appears.



2. (Optional) Click **Edit Keywords** to add or remove keywords from the list. The Edit Keywords window appears.



   a. Click + to add a keyword to the list. Enter a value for the keyword and specify a category.

   b. Select a keyword and click ✕ to remove it from the list. Click **Remove** in the confirmation message.

   c. Click **OK**.

3. Select one or more keywords from the list.

4. Click **OK**.

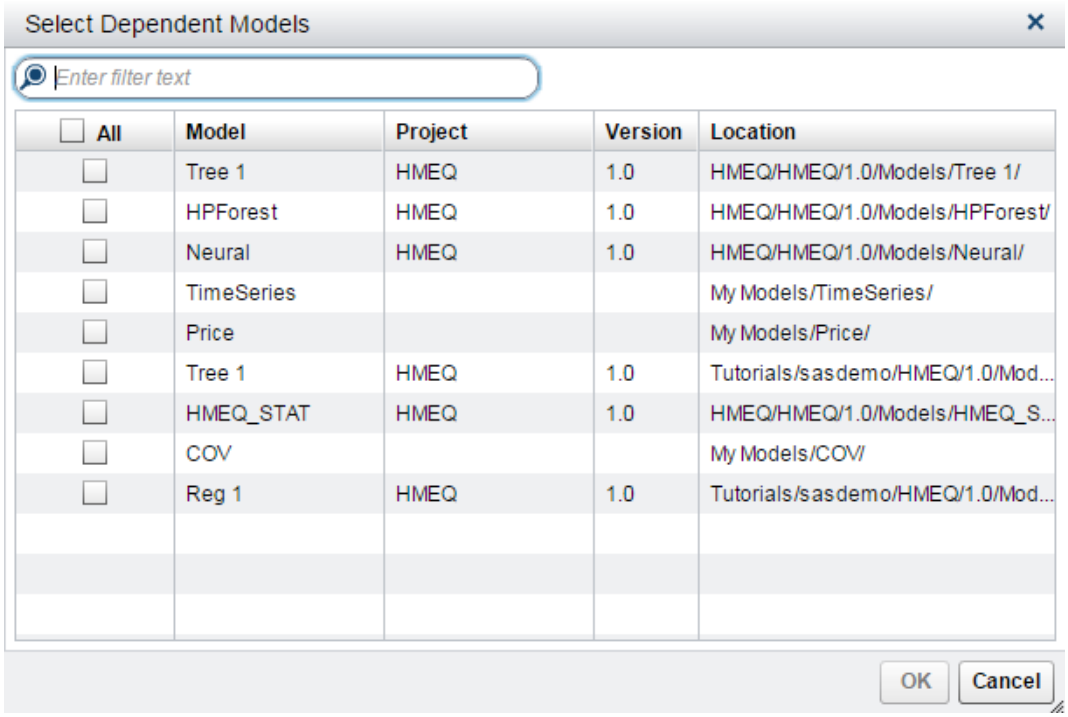
5. Click to save the model properties.

Add Model Dependencies and View Lineage

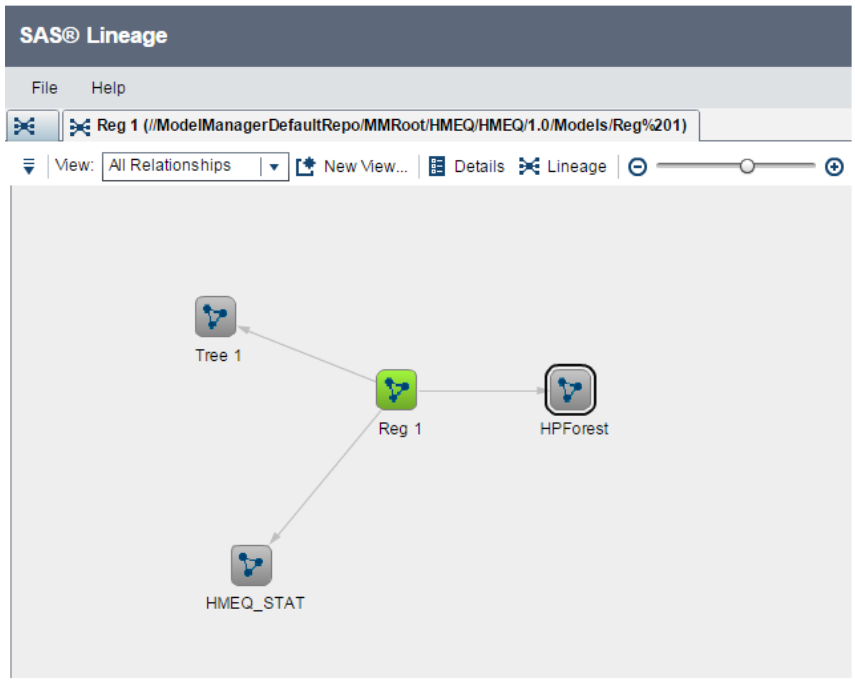
Select **Properties** ⇨ **General** to view the model dependencies.

To add model dependencies:

1. Click ➕. The Select Dependent Models window appears.



2. Select one or more models.
3. Click **OK**.
4. Click 💾 to save the model properties.
5. Click to view the lineage of the model. The SAS Lineage application appears.



Note: For information about SAS Lineage and using the lineage viewer, select **Help**.

To remove model dependencies:

1. Select one or more models from the list.
2. Click ✕.
3. Click 💾 to save the model properties.

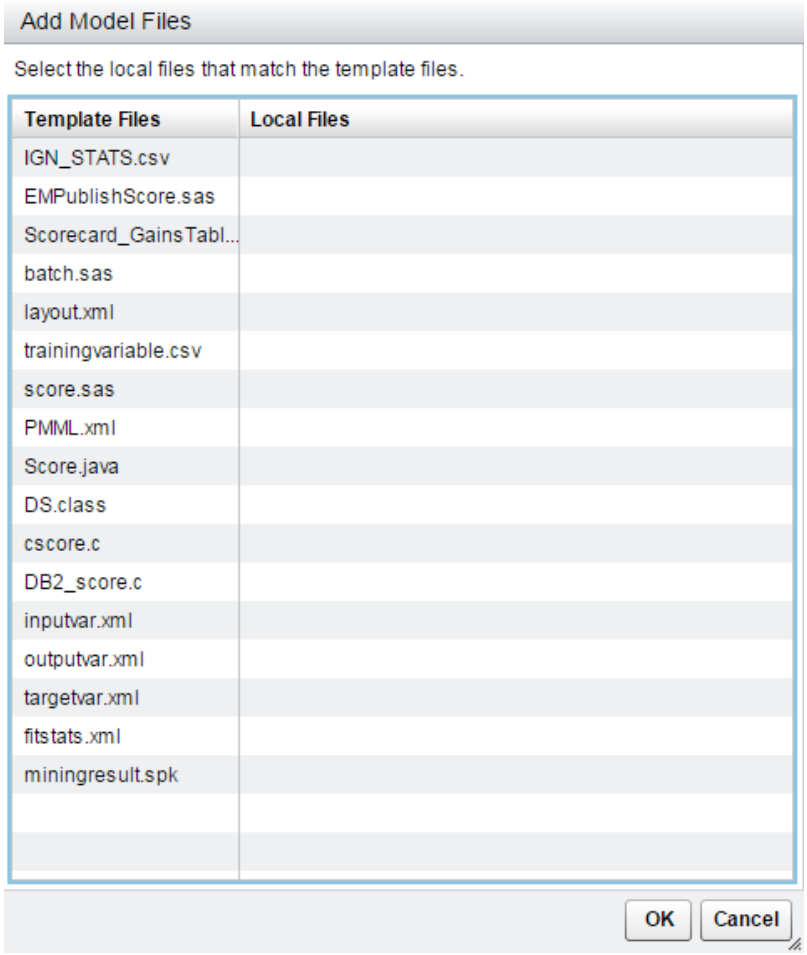
Add Model Files to an Existing Model

Suppose you want to import a model, but you lack some of the model component files that are needed to complete a model import. The model files utility enables you to add files later that were not available when the model was originally imported.

To add a local file to an existing model:

1. Select and open a model.
2. On the **Model Properties** page, select **Advanced** ⇨ **Model Files**.
3. Click ＋.
4. Select a row and click **Browse** to select the local files that match the template files.

   Note: Not all files must be specified. You can specify one or more.



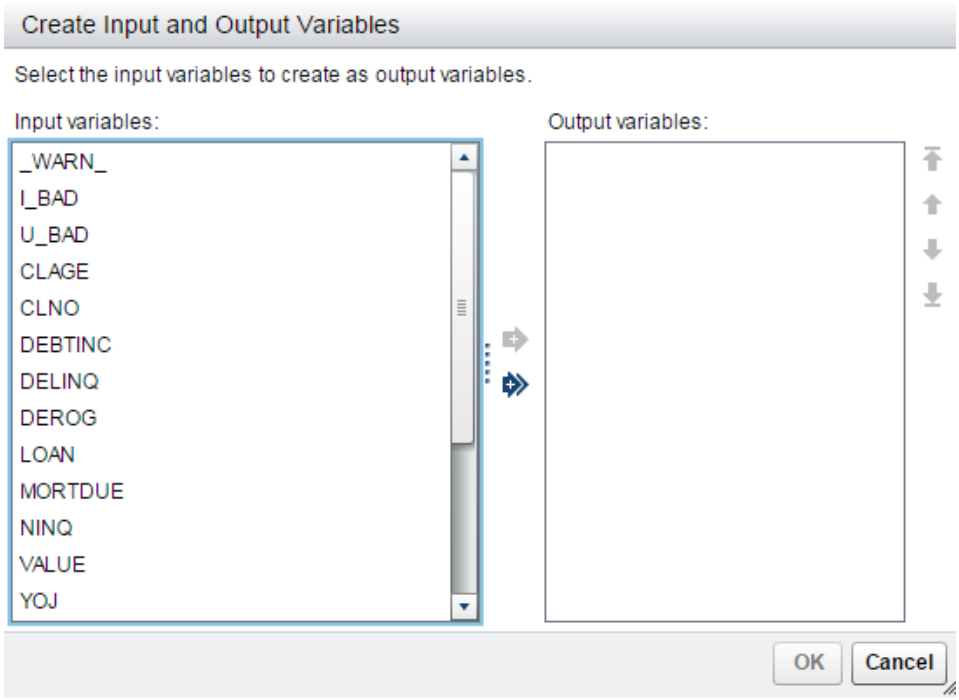
Note: If you are adding files to a model that is locating inside a folder, the Choose Files window appears for you to select a file from the **SAS Workspace**.

5. When the update is complete, click **OK**.
6. Click . If you do not see your updates immediately, you might need to close the model and reopen it.

Create Input and Output Variables from a SAS Code File

You can create model input and output variables from the score.sas file. Creating the variables enables you to generate missing metadata for model variables.

1. On the **Model Properties** page, select **Advanced** ⇨ **Model Files**.
2. Select a SAS code (.sas) file and click (for example, score.sas).
3. Click **Yes** in the confirmation messages to replace the existing input and output variables. The Create Input and Output Variables window appears.



Note: If a champion is already set, you might receive another message indicating that the champion and challenger models will be cleared. Also, if you add new variables that are not project variables, you are prompted to add the model variables to the project variables when setting the model as the champion.

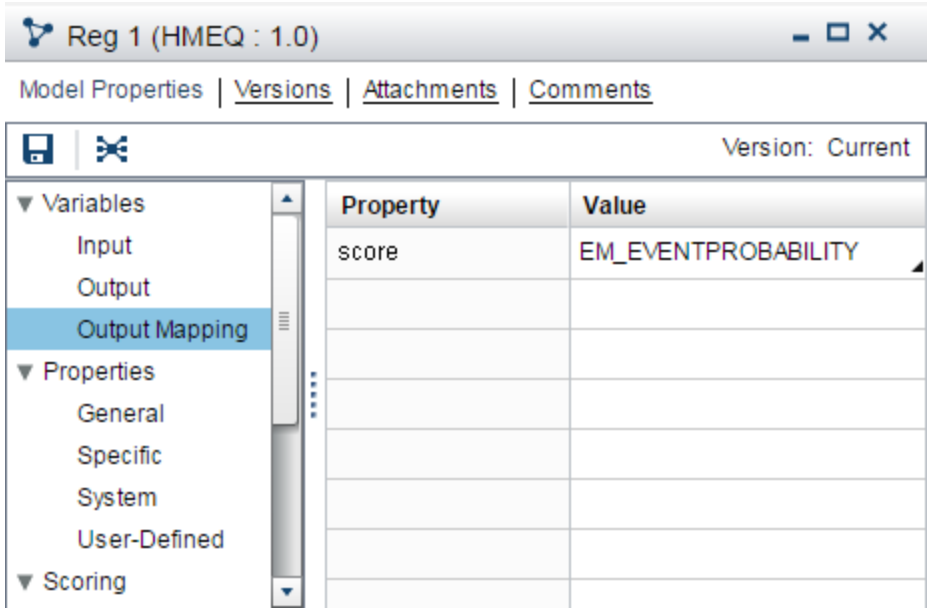
4. Select input variables that you want to add as output variables for the model.
5. Click **OK**. The inputvar.xml and outputvar.xml model files are generated.

Map Model Variables to Project Variables

After a model has been imported and the remaining model properties are set on the **Model Properties** page, you must map the model output variables to the project output variables. For more information about project input and output tables, see Defining Project Input and Output Variables on page 133.

To map model variables to project variables:

1. Select and open a model.
2. Select **Model Properties** ⇨ **Output Mapping**.
3. Click the box in the **Value** column beside the variable in the **Property** column to display project variables.



4. Select a model output variable.
5. Repeat steps 3 and 4 for each model variable that requires mapping.
6. Click .

User-Defined Model Templates

When you import a SAS code model or R model, you must define the component files to be used in the model and specify the properties for the model. SAS Decision Manager provides model templates that you can use as an example to create your own model template. You can define model component files and specify system and user properties for your model template. The model templates that are included cannot be modified. For a list of the component files that must be created for the different model types, see "Model Template Component Files" on page 340. For a list of properties, see "Specific Properties" on page 349.

Note: Only users who are in the Model Manager Administrator Users or Model Manager Advanced Users groups and who have Write permission to the WebDAV folder where the user-defined templates are stored can save a new template or save changes to an existing template. For more information, see "Verify WebDAV Folder Permissions for User-Defined Templates " in *SAS Decision Manager: Administrator's Guide*.

Several sample user template XML files are included with the installation package and are available to be used as a starting point for creating your own model template. For more information, see "Creating and Managing Templates" on page 138.

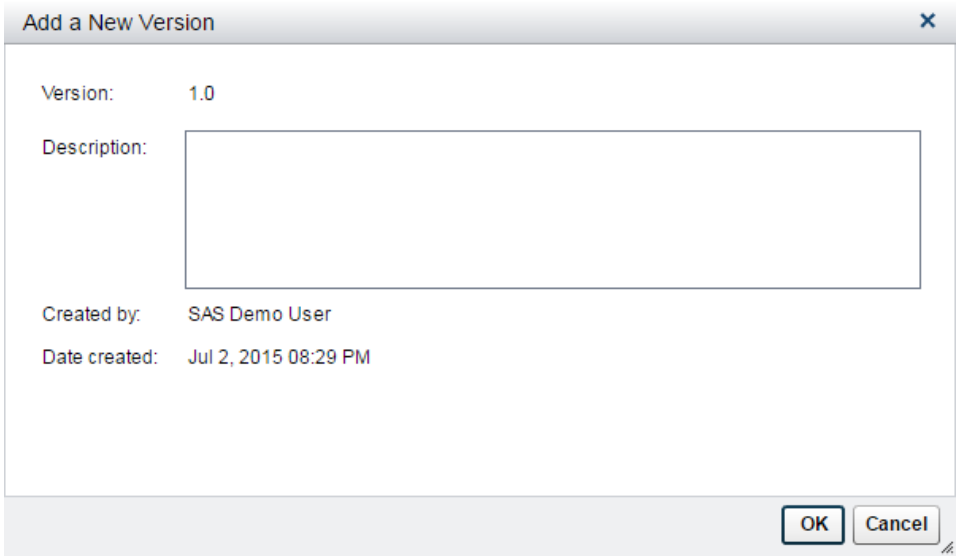
Managing Model Versions

About Model Versions

The current version of a model is the latest version in which the model properties and file contents are editable. If you add a new model version manually or perform an action that automatically creates a new model version (such as setting it as the champion model or publishing a champion model from the project level), a snapshot of the model's contents is taken and a version number is assigned. A new version is also created when you update a model that is located within a folder. However, the contents of the new model version that is created can no longer be edited. You can only view the contents of the new model version. Model versions cannot be deleted.

Add a Model Version

1. Select the **Versions** page.
2. Click +. The Add a New Version window appears. The next sequential number appears as the new version number.



3. (Optional) Enter a description for the version.
4. Click **OK**.

Set the Model Version

1. Select the **Versions** page.
2. Select a version and click . The icon indicates the version that is being displayed.

Add Attachments

You can view and add attachments such as images or documents. Attachments can be added at the object-level for portfolios, projects, and models.

To add an attachment:

1. Select the **Attachments** page.
2. Click +.
3. Select a file to attach and click **Open**.

   Note: Depending on your site's settings, you might be able to attach executable files to models, projects, and portfolios. For more information, see "Configure Support for Executable Attachments" in *SAS Decision Manager: Administrator's Guide*.

Note: Click ✕ to remove an attachment.

See Also

"Add Attachments to a Project" on page 137

Add Comments

You can add new topics or respond to an existing topic. You can also search the comments. Comments can be added at the object-level for portfolios, projects, and models.

To add a comment:

1. Select the **Comments** page
2. Enter a topic name and a comment.
3. (Optional) Click 📎 to attach a file to the new topic. Repeat this step to attach multiple files.

   Note: You cannot attach executable files to a topic.

   Note: You can also click **Remove** to remove an attachment.
4. Click **Post**.

See Also

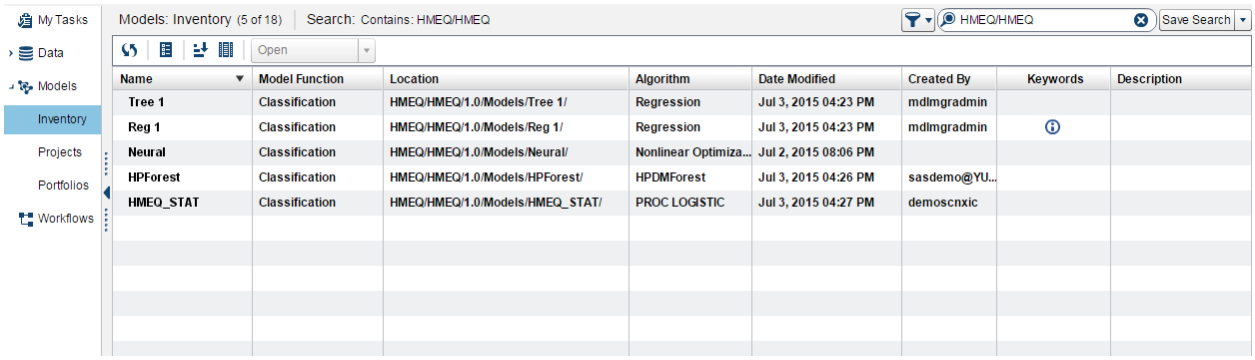
"Add Comments to a Project" on page 137

Chapter 13

Working with Model Inventory

Overview of Model Inventory

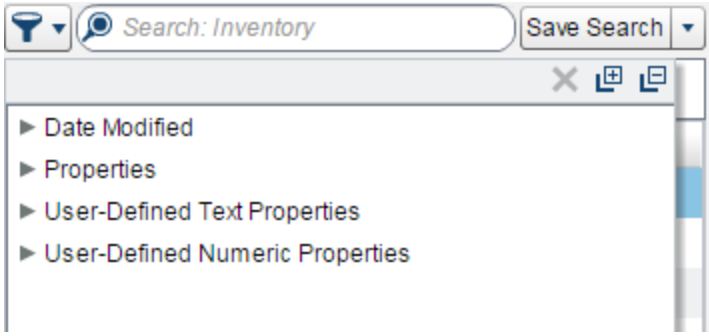
The Inventory category enables you to access all of the models in the model repository in one place, whether they are located in a folder, portfolio, or project. You can also filter the search results by date modified, keywords, model properties, and user-defined properties. You can add user-defined properties as columns in the list in order to display the values for each model.



Search and Filter Inventory of Models

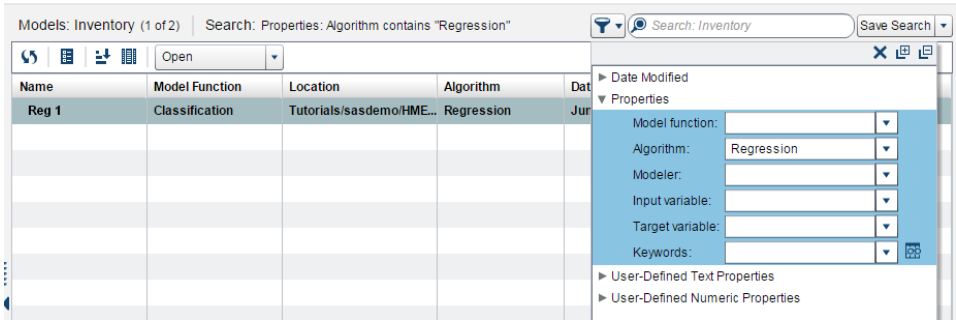
In addition to entering search terms in the search box, you can filter the search results by date modified, keywords, model properties, and user-defined properties.

1. Select **Models** ⇨ **Inventory**.
2. In the search box, enter the search terms that you want to include in the new search.
3. Click . The additional search options appear.



4. Expand one or more search options and specify values to filter the search results. The model inventory list is filtered automatically.

   For example, select **`Regression`** from the **Algorithm** property drop-down list.



5. (Optional) Save your search.

   Note: You cannot modify an existing search, but you can create a new one and replace the old one.

   a. Click **Save Search**. The Save Search window appears.

   b. In the Save Search window, enter the name of the new search or enter the same name as the saved search that you want to update.

   c. (Optional) Enter a description.

   d. Click **Save**.

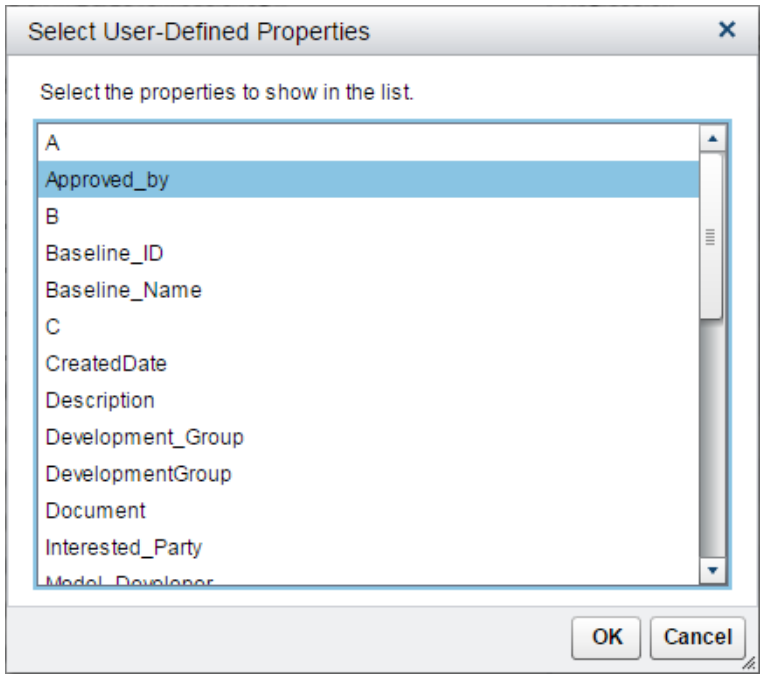
   e. In the confirmation window, click **Yes** to replace the existing search.

Add Model User-Defined Properties to Inventory List

You can add user-defined properties as columns to the list. The values of the properties are displayed for each model.

To add user-defined properties to the list:

1. Click .



2. Select the properties that you want to appear in the list.
3. Click **OK**. The columns that are added to the list can then be used to sort the list of models.

Chapter 14

Working with Portfolios

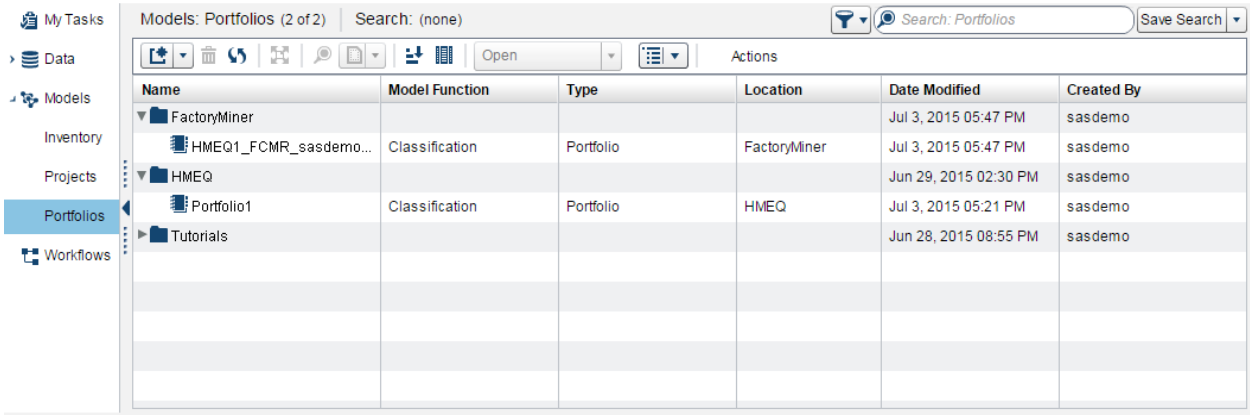
Overview of Portfolios

SAS Decision Manager enables you to create a portfolio in the model repository. You use a portfolio to manage multiple projects and models in one place. From a portfolio level, you can create multiple projects from a control table and then add new versions or new input variables to all projects within the portfolio. After you set the champion model for each project, you can monitor the performance of the champion models for all projects and publish the champion models to the SAS Metadata Repository.

Portfolios are also created when you use SAS Factory Miner to register projects to the model repository. The portfolios contain the project segments and models from the SAS Factory Miner project. For more information about how to register SAS Factory Miner project segments and models, see *SAS Factory Miner: User's Guide*

Note: Because of how portfolios and projects are created in the SAS Decision Manager model repository, SAS Factory Miner project names, model names, and segment variable values cannot contain special characters. That is, only alphanumeric

characters, the underscore, and the hyphen are allowed. Users are unable to register models when special characters are encountered.



Here are the tasks that can be performed for a portfolio:

- Create a portfolio
- Add a new version
- Add an input variable to all projects
- Publish project champion models
- Monitor performance of project champion models

Planning a Portfolio

Before you begin a portfolio, you must plan your portfolio resources. Here are questions to consider and conditions to meet for modeling projects within a portfolio:

- After you know which users are assigned to the projects within a portfolio, a SAS Decision Manager administrator must ensure that the user is assigned to the appropriate user group and role. For more information, see "Configuring Users, Groups, and Roles" in *SAS Decision Manager: Administrator's Guide*.
- How do you want to structure the projects within the portfolio? A portfolio is an object within a folder. The Portfolios category view enables multiple levels of folders so that you can customize how you structure the portfolios. For more information, see Chapter 9, "Managing Folders for Model Projects and Portfolios," on page 121.
- What models do you want to use in each project of the portfolio? If the models were created using SAS Enterprise Miner, SAS/STAT, or the SAS/ETS procedures COUNTREG and SEVERITY, all model components are available to SAS Decision Manager when you import the model. Only models that are contained in an SPK file can be imported. At least one SPK file must be prepared for each project and the SPK files should be placed in the same location. If your model is a SAS code model or a PMML model that is not contained in an SPK file, you must import it separately into the desired project within the portfolio, after the portfolio is created. You must also ensure that you have imported all of the model component files. For more information, see "Import Models from Local Files" on page 150 and "Import a PMML Model" on page 149.
- What model function do you want to use in each project of the portfolio?

SAS Decision Manager has several model function types:

- Classification
- Prediction
- Segmentation
- Analytical

After the model function is specified for the portfolio, the **Model function** property for a project cannot be changed. Ensure that the types of models that you are going to use in each project of the portfolio fit within the selected model function type. For more information, see Table 10.1 on page 130.

- How do you want to define your project input and output variables? When you create a portfolio, you can import the variables using input and output prototype tables. The project variables are set for each project within the portfolio. The prototype tables must be registered in the SAS Metadata Repository. Tables that were registered using the SAS Management Console must also be made available in the Data category view before you create the portfolio. For more information, see "Defining Project Input and Output Variables" on page 133.
- What method do you want to use to track the progress of a version? The Workflows and My Tasks category views enable you to track the progress of tasks from the version level for each individual project within a portfolio. An authorized user can create a workflow and associate it with a version. For more information, see "Overview of Using Workflows" on page 307.
- When you publish project champion models from a portfolio to the SAS Metadata Repository, you must specify a location in which to store the models. You might need to create a folder in the SAS Metadata Repository, if one does not already exist. For more information, see "Publishing Models from a Portfolio" on page 181.
- After your project champion models are in a production environment, you can monitor the performance of the project champion models within a portfolio in SAS Decision Manager using your organization's operational data. If you use SAS Decision Manager to monitor performance of projects within a portfolio, you must first prepare performance tables using the operational data and then register the tables in the SAS Metadata Repository using the Data category view. Tables that are registered to the SAS Metadata Repository using SAS Management Console must also be made available to the Data category view. For more information, see "Creating a Performance Table" on page 360.
- When you run performance monitoring reports, you can set up performance index alert and warning conditions to notify users if conditions exceed the indexes. For more information, see "Performance Index Warnings and Alerts" on page 251.

Prerequisites for Creating Portfolios

After you have planned the projects and models that you want to have in your portfolio, you must create a project control table that contains the segment identifiers, projects, and models. The project control table can then be used to create a hierarchy of your portfolio.

Portfolios can be created only by authorized users who have the capability to access the Portfolios category. Ensure that users who create portfolios are assigned to the group **Model Manager Administrator Users** or **Model Manager Advanced Users** in SAS Management Console.

The project control table must contain the project names (project_name variable) to create the projects within the portfolio. At least one segment identifier variable (for example, segid) is required, and that segment identifier variable must also be in the performance data set. When you want to monitor the performance of project champion models, you must also associate the model name (model variable) with each project (project_name) and segment identifier (segid, or another name for the segments) in the table.

You must know the model function type before you create a portfolio. SAS Decision Manager has several model function types:

- Classification
- Prediction
- Segmentation
- Analytical

To determine the model function type for your project, compare your model to the descriptions in Table 10.1 on page 130.

If you use prototype tables to define the project input and output variables, you must do one of the following two things before you can create a portfolio. Create the project input and output tables and register them in the SAS Metadata Repository using the **Data** category view. Tables that are registered to the SAS Metadata Repository using the SAS Management Console must then be made available to the **Data** category view of SAS Decision Manager. See the following documents for details:

- For instructions about creating project input and output tables, see "Creating Project Input and Output Tables" on page 357.
- For instructions about registering tables using the Data category view, see Chapter 3, "Managing Data Tables," on page 47.

Creating a Project Control Table

After you have planned the projects and models that you want to have in your portfolio, you must create a project control table that contains the segment identifiers, projects, and models. The project control table is then used to create the hierarchy of your portfolio when you create a new portfolio. The variable names that are required in the project control table are at least one segment identifier (for example, segid), project_name, and model. All variables other than project_name and model are treated as segment identifier variables. The segment identifier variables do not have a required naming convention.

Here is an example of the code to create a project control table.

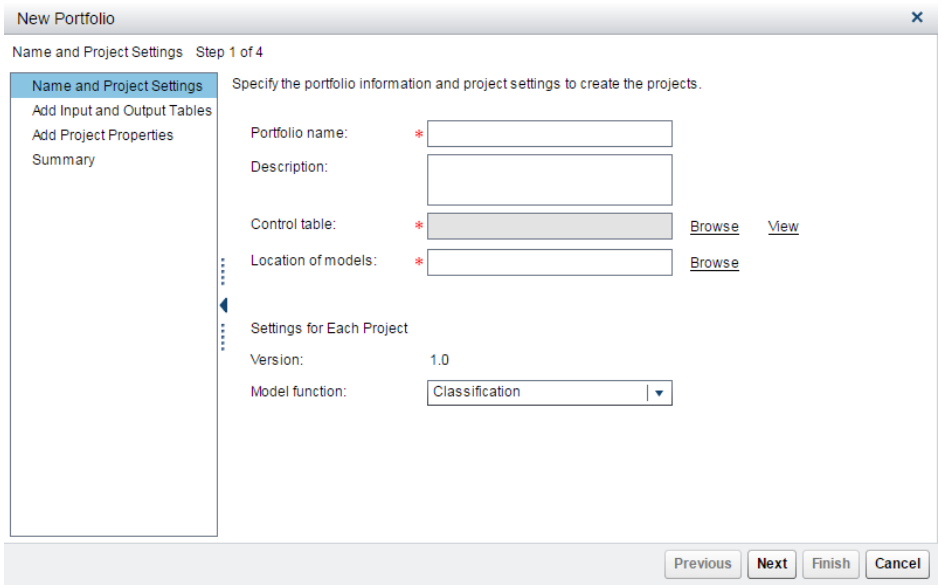
```
data control_Table;
 length segid project_name model $20;
 infile datalines dsd dlm=',' missover;
 input segid project_name model;
 datalines;
 seg01,US,reg1.spk
 seg02,Canada,tree1.spk
 seg03,Germany,hpf_class.spk
 ;
 run;
```

Create a New Portfolio

To create a new portfolio:

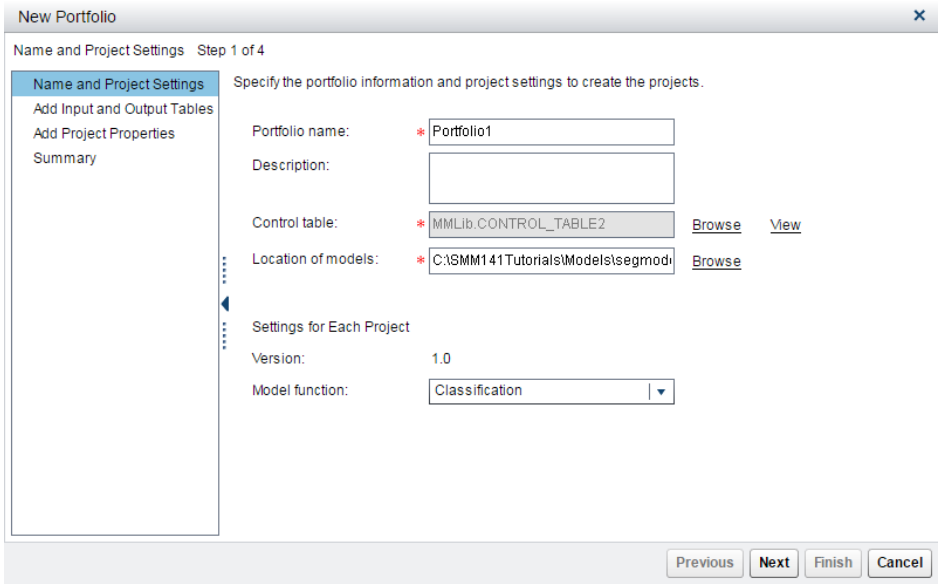
1. Verify that the project control table contains the required variables. For more information, see "Prerequisites for Creating Portfolios" on page 175.
2. Select a folder or create a new folder in which to store the new portfolio.
3. Click and select **New Portfolio**. The New Portfolio window appears.

   Note: Alternatively, you can right-click a folder and select **New Portfolio**.



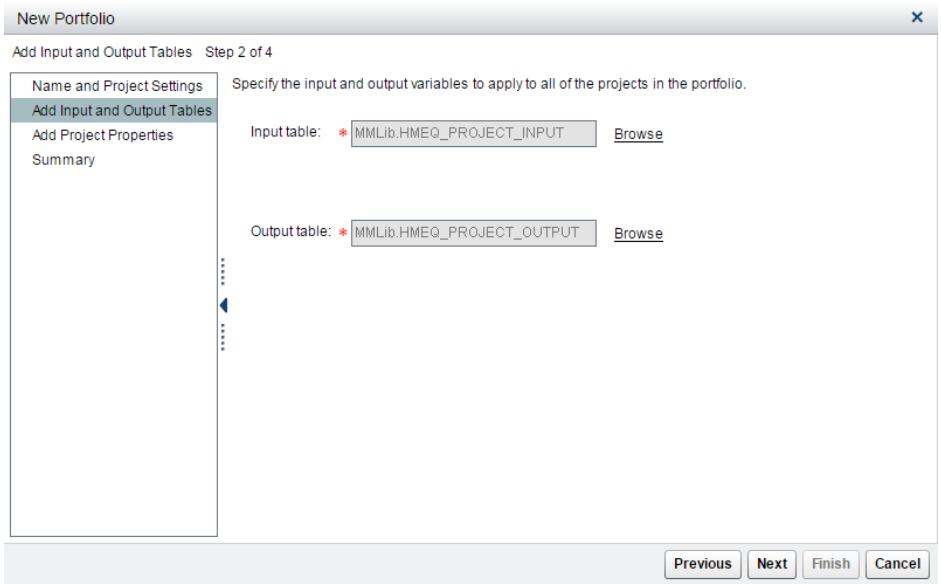
   Note: The value for the initial version is auto-populated and is the version name that is created within each project for the new portfolio.

4. Enter a name for the portfolio.
5. (Optional) Enter a description for the portfolio.
6. Click **Browse** to select the control table. Click **OK**.
7. Click **Browse** to select the location of the model SPK files that are specified in the control table. Click **OK**.
8. Select a model function to indicate the type of models that should be imported into each project within the portfolio.



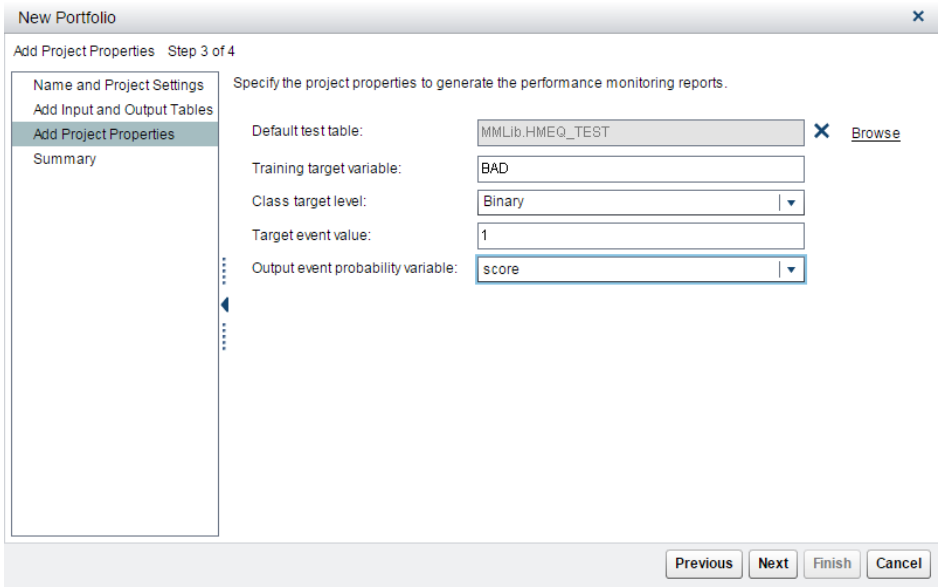
9. Click **Next**.

10. Click **Browse** to select the input and output tables. The input and output variables in the tables are applied to all of the projects.

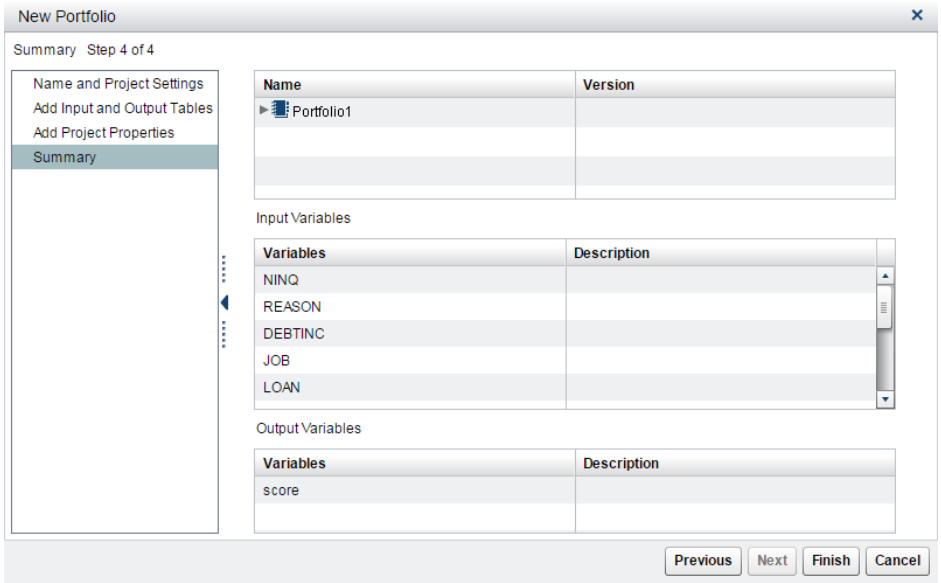


Click **Next**.

11. Specify the project properties to apply to all projects within the portfolio. The properties are used to perform tasks and generate reports.



12. Click **Next** to view the summary of information that has been specified.

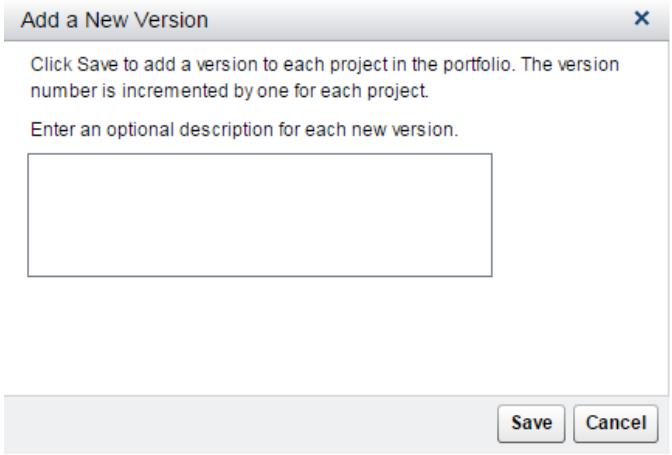


13. Click **Finish**. The new portfolio appears in the list.

Add a New Version

You can add a new version to all projects within a portfolio.

1. Open a portfolio, select the **Projects** page, and click ✚. The Add a New Version window appears.

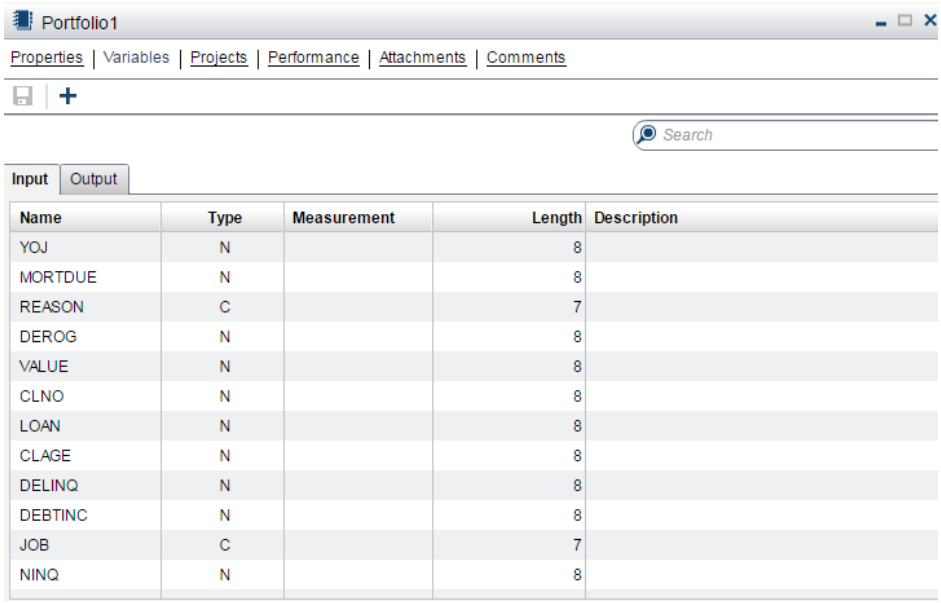


2. (Optional) Enter a description for each new version.
3. Click **Save**. The version number is incremented by one for each project within the portfolio.
4. Click **OK** for the confirmation message.

Add an Input Variable

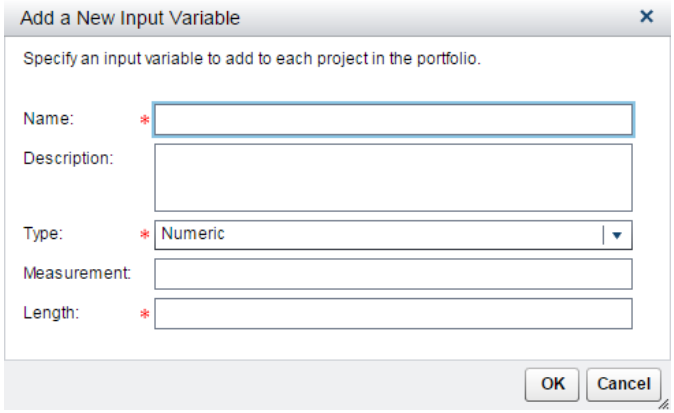
You can add input variables to each project within a portfolio.

1. Open a portfolio.
2. Select the **Variables** page and click the **Input** tab.



Name	Type	Measurement	Length	Description
YOJ	N		8	
MORTDUE	N		8	
REASON	C		7	
DEROG	N		8	
VALUE	N		8	
CLNO	N		8	
LOAN	N		8	
CLAGE	N		8	
DELINQ	N		8	
DEBTINC	N		8	
JOB	C		7	
NINQ	N		8	

3. Click +.



4. Enter a name.
5. (Optional) Enter a description.
6. Select a type.
7. (Optional) Enter a measurement.
8. Enter a length.
9. Click **OK**. The input variable is added to the portfolio and to all projects within the portfolio.

Publishing Models from a Portfolio

About Publishing Models

To publish the champion models and challenger models for projects within a portfolio, you must have already set the models that you want to publish as project champion models or challengers. SAS Decision Manager examines the projects and always publishes the champion models. When the champion model for a project changes and you publish the model again to the same location, the scoring application automatically uses the latest score code. In the Portfolios category view, when you select a portfolio, you only can publish the project champion models to the SAS Metadata Repository. When you open a portfolio, the **Projects** page offers you the option of publishing a project champion model and its challengers to the SAS Metadata Repository, a SAS Channel, and to a configured database or Hadoop.

Note: SAS Decision Manager cannot publish R models. SAS analytic store models within a project segment can be published only to Hadoop and Teradata. Publishing SAS analytic store models to the SAS Metadata Repository is not supported.

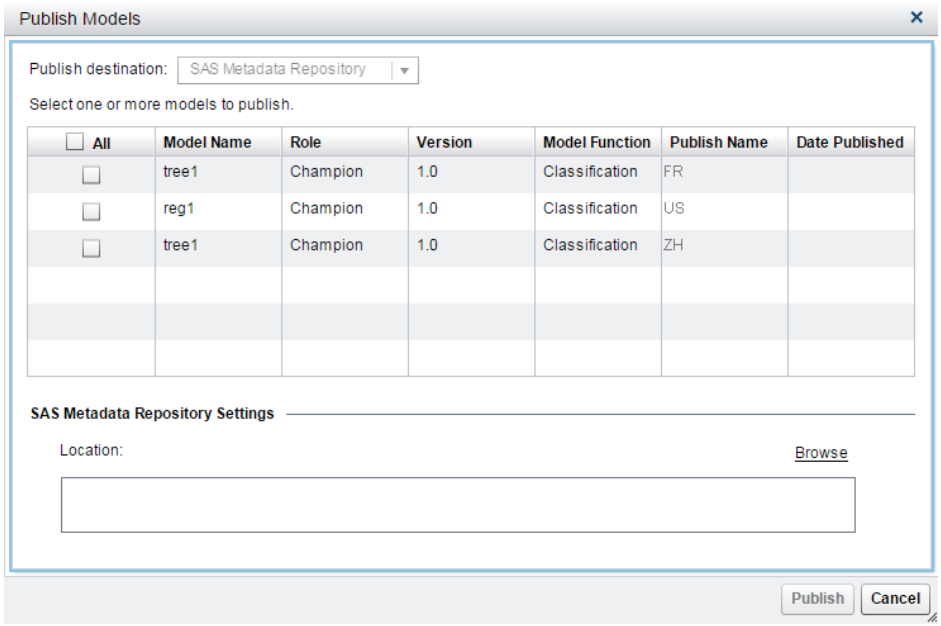
To verify that a champion model has been assigned to all of the projects within a portfolio that you want to publish. Open a project and select **Properties** ⇨ **Specific**. The **Champion version** property contains the name of the champion version. For more information, see "Champion Models" on page 286.

Publishing Project Champion Models

In the Portfolios category view, you can publish the champion models for projects within a portfolio to the SAS Metadata Repository.

To publish champion models for projects in a portfolio:

1. Select a portfolio and click .



2. Select one or more champion models that you want to publish from the models list.
3. Click **Browse** and select the location to publish the model to.
4. Click **Publish**.
5. Click **Close** in the confirmation message.

Note: Alternatively, you can right-click a portfolio and select **Publish**.

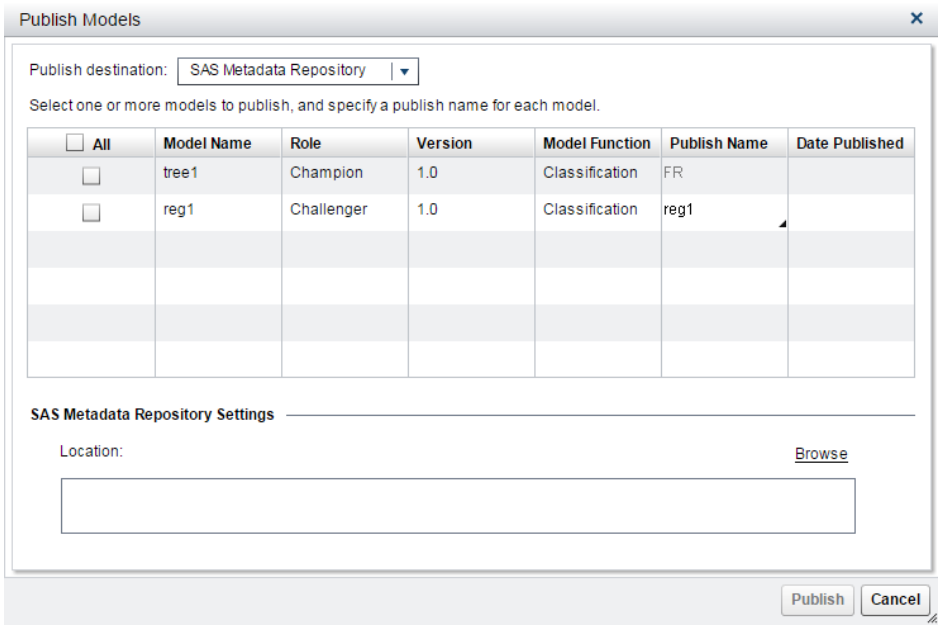
See Also

"Publishing Models to the SAS Metadata Repository" on page 293

Publish Champion and Challenger Models

Publish to the SAS Metadata Repository

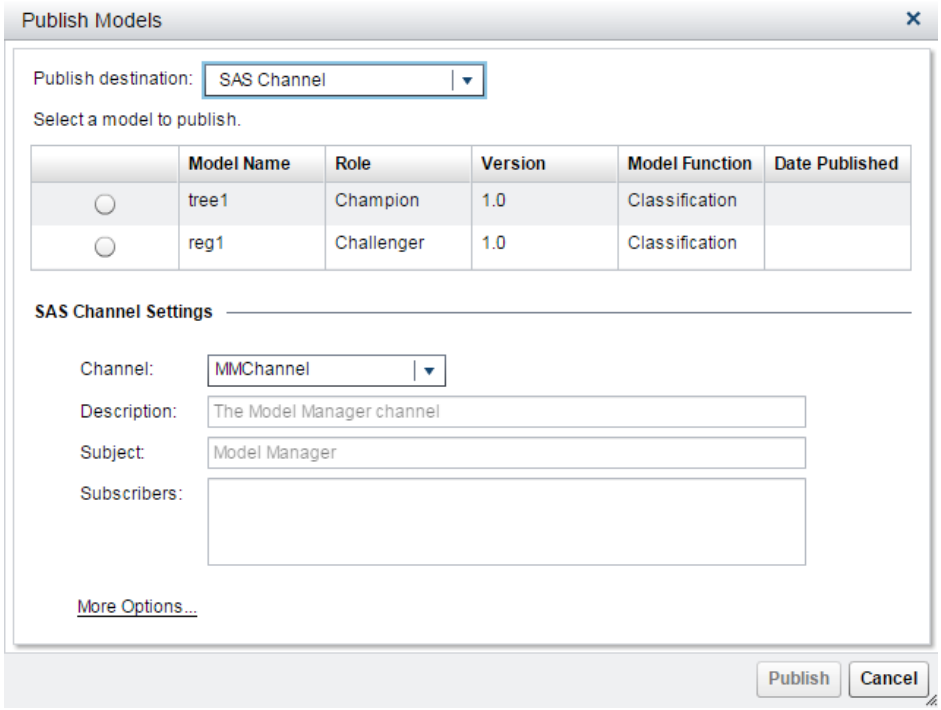
1. Open a portfolio and select the **Projects** page.
2. Select a project and click .
3. Select **SAS Metadata Repository** from the publish destination list.



4. Specify a **Publish Name** for the challenger models. The publish name for a champion model cannot be modified.
5. Click **Browse** and select the location to publish the model to.
6. Click **Publish**.

Publish to a SAS Channel

1. Open a portfolio and select the **Projects** page.
2. Select a project and click .
3. Select **SAS Channel** from the publish destination list.

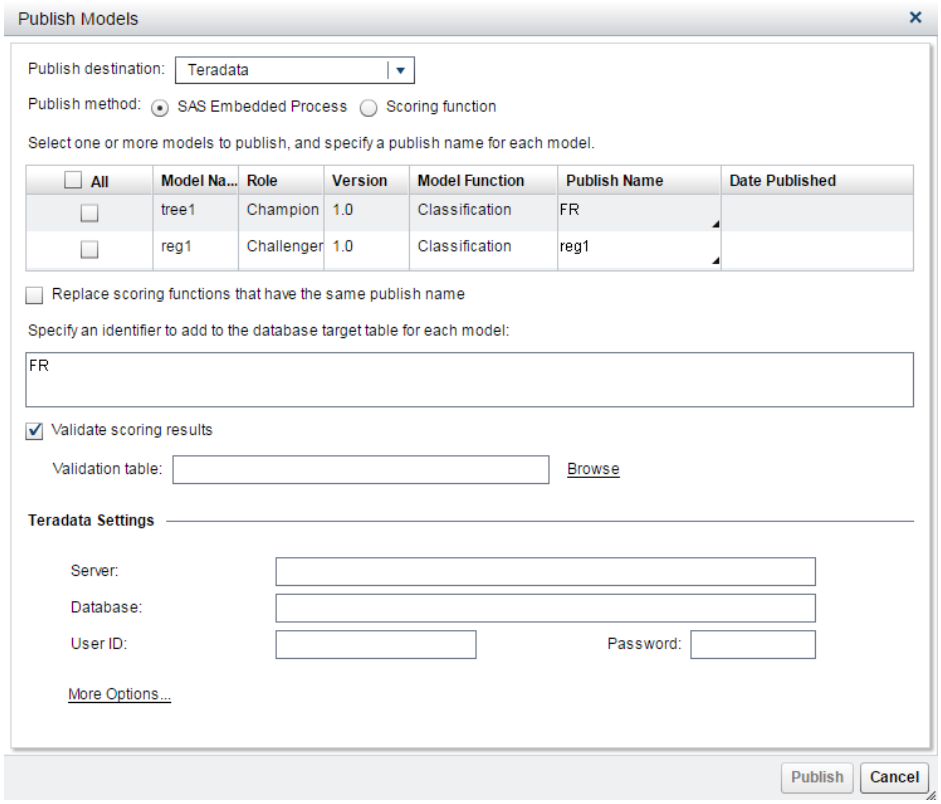


4. Select the model that you want to publish from the models list.

5. Select a publication channel from the channel drop-down list.
6. (Optional) Click **More Options** to specify a message subject, notes, and user-defined properties. Click **Save**.
7. Click **Publish**.

Publish to a Database

1. Open a portfolio and select the **Projects** page.
2. Select a project and click .
3. Select a database from the publish destination list.



4. Select a publish method.
5. Select one or more models that you want to publish from the models list.

   Note: SAS analytic store models can be published only to Hadoop and Teradata.
6. Specify a **Publish Name** for each model.

   Note: The default format of the publish name is configured by the SAS administrator.
7. (Optional) Select whether to **Replace scoring files that have the same publish name**.
8. Specify an identifier to add to the database target table for each model.
9. (Optional) Select whether to **Validate scoring results**. If selected, click **Browse** to navigate to the appropriate train table.
10. Specify the database settings.
11. Click **More Options** to specify other options for the database.

12. Click **Publish**.

Remove Published Models from a Database

The SAS Embedded Process publish method enables you to replace the model scoring files, but the scoring function publish method publishes the model as a separate entry in the database each time. If you modify the previously published models or change the champion model or challenger models, the Remove Models from a Database feature enables you to remove the previously published models, so that you can clean up the test or production database.

To remove models from a database:

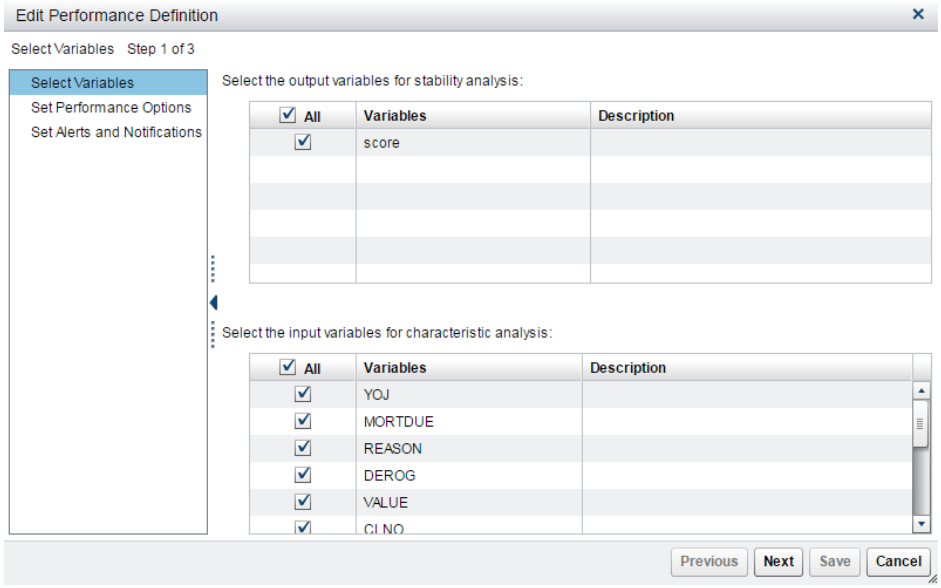
1. Open a portfolio and select the **Projects** page.
2. Select a project, and click .
3. Specify the database settings and click **Log On**.
4. Select the models that you want to remove from the database.
5. Click **Remove Models**. A warning message appears.
6. Click **Yes**.

Monitor Performance of Project Champion Models

To create performance monitoring reports for all projects within a portfolio, you create and execute a performance definition for all projects within a portfolio. Execution of the generated code creates the SAS data sets that are used to display the performance monitoring reports on the **Performance** page of each project.

To monitor the performance of the champion models for all projects:

1. On the **Performance** page of a portfolio, click .
2. Select one or more output variables for stability analysis. To select all output variables, click **All**.
3. Select one or more input variables for characteristic analysis. To select all input variables, click **All**.

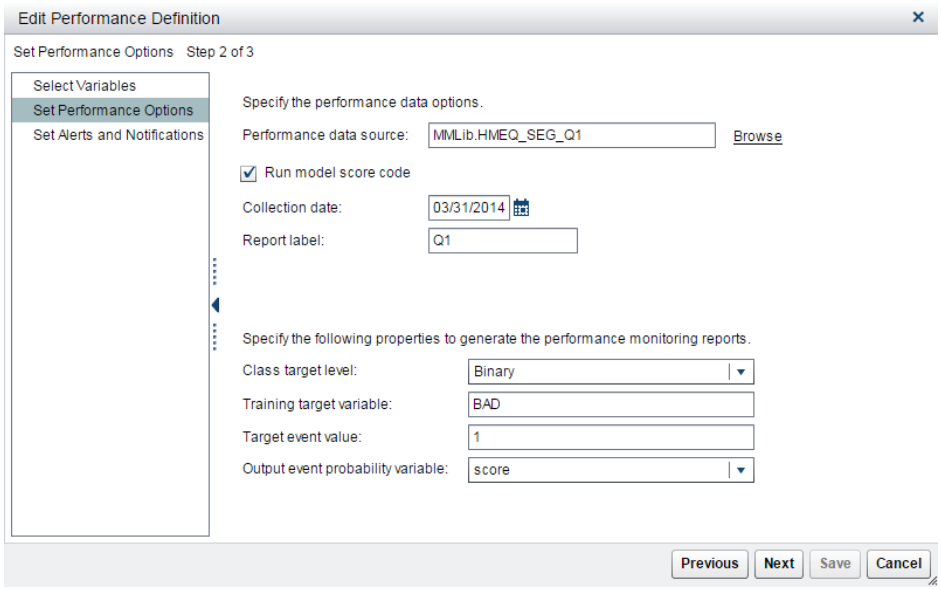


Click **Next**.

4. Specify the performance data options.
   - Click **Browse** to select the performance data source.

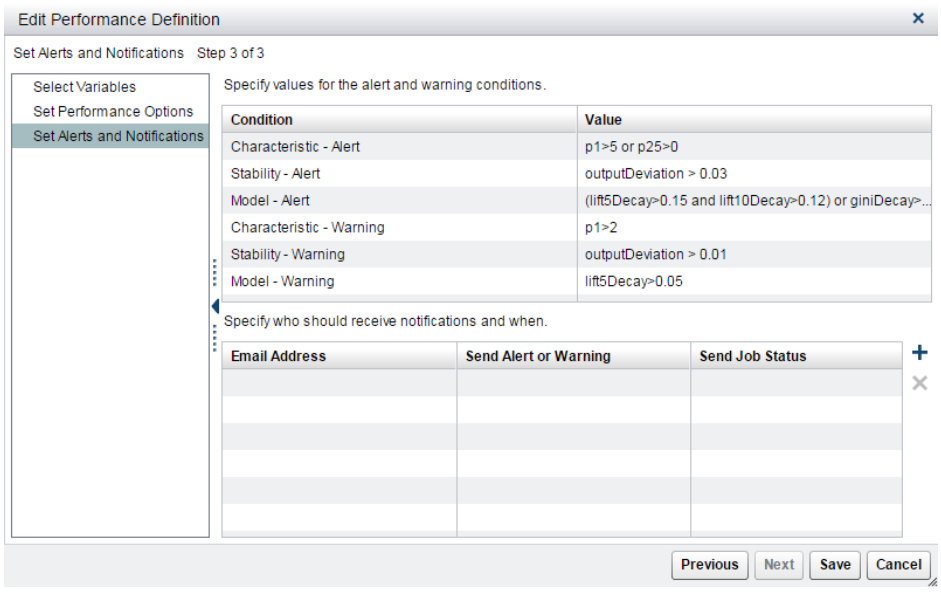
     Note: The performance data source must contain the same segment identifier variables as the control table.
   - To run the score code in the performance monitor job, select the **Run model score code** check box. If the check box is not selected, all of the output variables for stability analysis must be in the performance data source.
   - Click and select a date. The date can be any date in the time period when the performance data was collected.
   - Enter a report label to associate with the performance data. The report label represents the time point of the performance data source. Because the report label appears in the performance charts, use a label that has not been used for another time period, is short, and is understandable (for example, Q1).

     Note: If you duplicate report labels, previous performance results are overwritten.
5. Specify the properties that are used to generate the performance monitoring reports. The properties default to the values that were set when you created a portfolio.

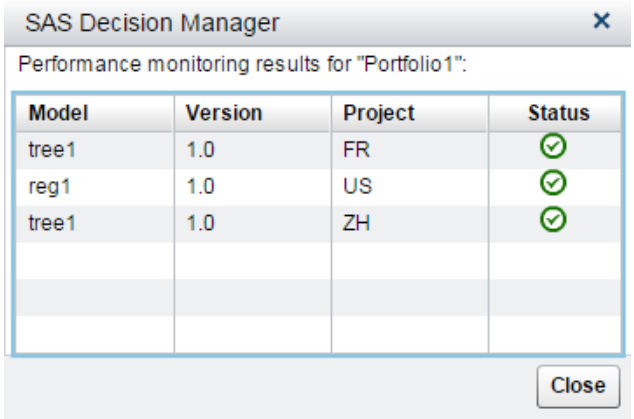


Click **Next**.

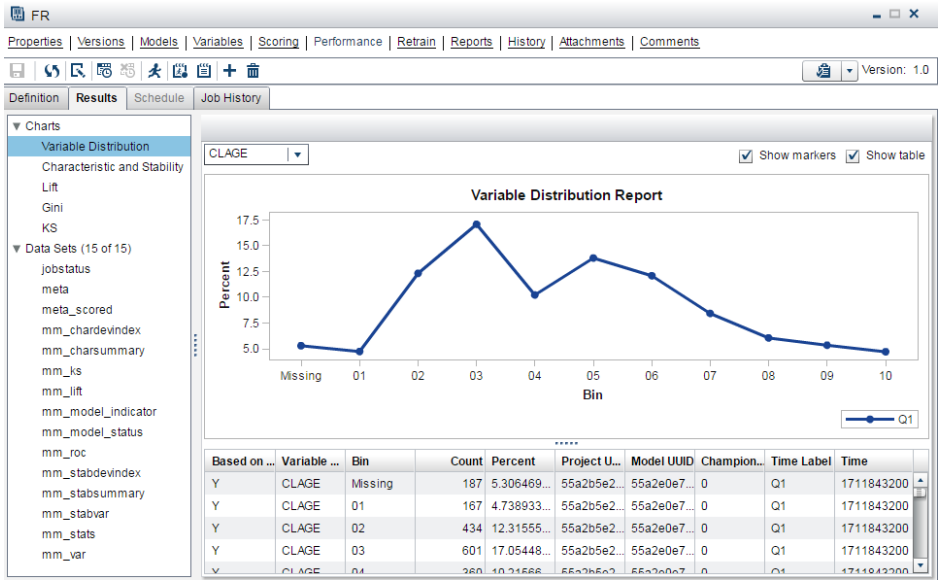
6. (Optional) Specify values for the alert and warning conditions or accept the defaults.
7. (Optional) To send the results by email, click +. A new row is added to the table.
   a. Enter an email address.
   b. Select **Yes** if you want an alert or warning to be sent by email when alert or warning thresholds have been exceeded.
   c. Select **Yes** if you want a completion notice with the job status to be sent by email every time the report runs.



8. Click **Save**.
9. Click .
10. After the performance monitoring is complete, a confirmation message appears. Click **Close**.



11. To view the performance results, select the **Projects** page, and open a project. Select the **Performance** page to view results.



See Also

"Prerequisites for Editing a Performance Definition" on page 257

Add Attachments

You can view and add attachments such as images or documents. Attachments can be added at the object-level for portfolios, projects, and models.

To add an attachment:

1. Select the **Attachments** page.
2. Click ➕.
3. Select a file to attach and click **Open**.

   Note: Depending on your site's settings, you might be able to attach executable files to models, projects, and portfolios. For more information, see "Configure

Support for Executable Attachments" in *SAS Decision Manager: Administrator's Guide*.

Note: Click ✕ to remove an attachment.

See Also

"Add Attachments to a Project" on page 137

Add Comments

You can add new topics or respond to an existing topic. You can also search the comments. Comments can be added at the object-level for portfolios, projects, and models.

To add a comment:

1. Select the **Comments** page
2. Enter a topic name and a comment.
3. (Optional) Click 📎 to attach a file to the new topic. Repeat this step to attach multiple files.

   Note: You cannot attach executable files to a topic.

   Note: You can also click **Remove** to remove an attachment.
4. Click **Post**.

See Also

"Add Comments to a Project" on page 137

Part 5

Evaluating Models and Monitoring Performance

Chapter 15

Scoring Models

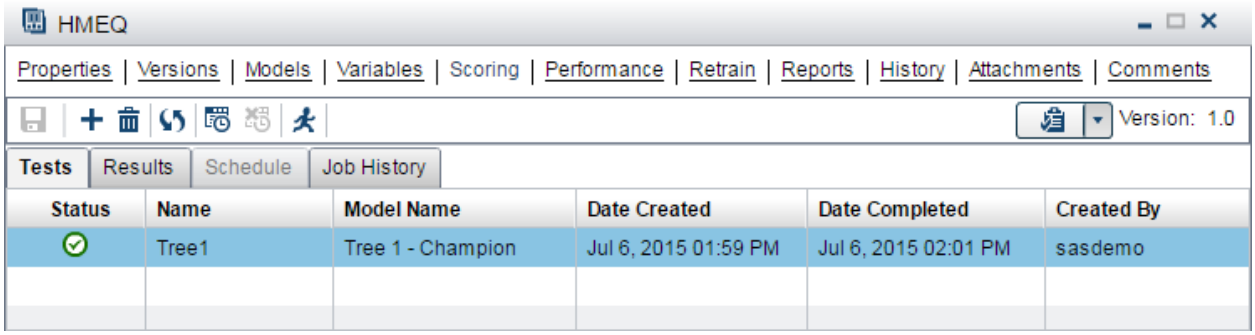
Overview of Scoring Tests

The purpose of a scoring test is to run the score code of a model and produce scoring results that you can use for scoring accuracy and performance analysis. The scoring test uses data from a scoring test input table to generate the scoring test output table. The following types of score code for a model can be imported: analytic store, DATA step fragment, and ready-to-run SAS code.

If your environment has its own means of executing the score code, then your use of the SAS Decision Manager scoring tests is mostly limited to testing the score code. Otherwise, you can use the scoring tests both to test your score code and execute it in a production environment. Scoring results for a model in a test environment are stored on the SAS Content Server. Scoring results for a model in a production environment are written to the location that the output table metadata specifies. In Windows, the scoring test output table in a SAS library must have Modify, Read and Execute, Read, and Write security permissions. For more information, see "Configuring Users, Groups, and Roles" in *SAS Decision Manager: Administrator's Guide*.

CAUTION:
Executing a scoring test in production mode overwrites the scoring test output table, which might result in a loss of data.

You create a new scoring test in the **Scoring** page of your project.



These are the tests that you perform as part of the scoring test workflow:

- Before creating a scoring test, you must create and register scoring test input and output tables. For more information, see "Create Scoring Output Tables" on page 194.
- When a new scoring test is successfully created, the scoring test is selected on the **Scoring** page. The scoring test displays the various scoring test information. For more information, see "Create a Scoring Test" on page 196.
- Before you execute the scoring test, it is recommended that you verify the scoring test output variable mappings on the Scoring Output Table view. For more information, "Create Scoring Output Tables" on page 194.
- To execute a scoring test, you can select and run a test. For more information, see "Execute a Scoring Test" on page 197.
- To run a scoring test at a scheduled time, you can specify the date, time and frequency that you want the scoring test to run. For more information, see "Schedule a Scoring Test" on page 197.
- After the successful execution of the scoring test, you can view the results on the Results tab. For more information, see "Execute a Scoring Test" on page 197.

Create Scoring Output Tables

What Is a Scoring Output Table?

A scoring output table is a SAS data set that contains the data from executing a scoring test. The scoring output table cannot be a database table. You can provide a scoring output table or you can create a scoring output table definition using SAS Decision Manager. When you create a scoring test, you specify either the scoring output table that you provide or the scoring test output definition as the scoring test output table. A SAS data set that you provide as a scoring output table must be registered in the SAS Metadata Repository and made available to SAS Decision Manager in the Data category view.

You can create a scoring output table definition by using the Create Scoring Output Table function directly from the model. You select variables from a scoring test input table as well as variables from the model's output. The variables in the input variables table are variables from the scoring test input table when one is specified for the **Default scoring input table** property on the **Model Properties** page for the selected model, or on the **Project Properties** page for the project that contains the selected model. Otherwise, the input variables table is empty. The output variables that appear are model

output variables. You use the variables from both tables to create the scoring output table. For more information, see "Set Model Properties" on page 157.

If you create a scoring output table on the **Model Properties** page, it is automatically saved in the SAS Metadata Repository. You then have to add it to the desired library in the Data category view. If you add an existing scoring output table to a library in the Data category view, it must be available in the SAS Metadata Repository.

SAS Decision Manager saves the table definition as metadata in the SAS Metadata Repository. The location of the metadata is defined by the SAS library that you specify when you create the output table definition. After the table definition is created, the table can be selected as the output table for subsequent scoring tests.

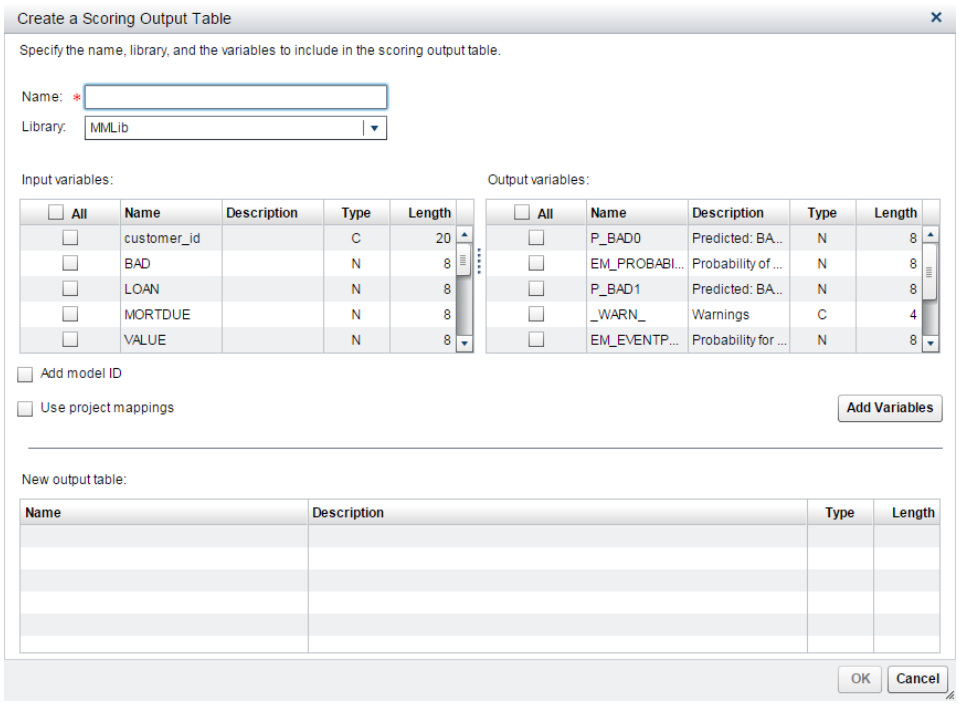
You can view a scoring output table definition in the Data category view. Scoring test results are stored in the **Results** tab on the **Scoring** page.

Create a Scoring Output Table

To create a scoring output table:

1. Select a model on the **Models** page and click . The Create a Scoring Output Table window appears.

   Note: You can also open a model and then select **Model Properties** ⇨ **Variables** ⇨ **Output** to create a scoring output table.



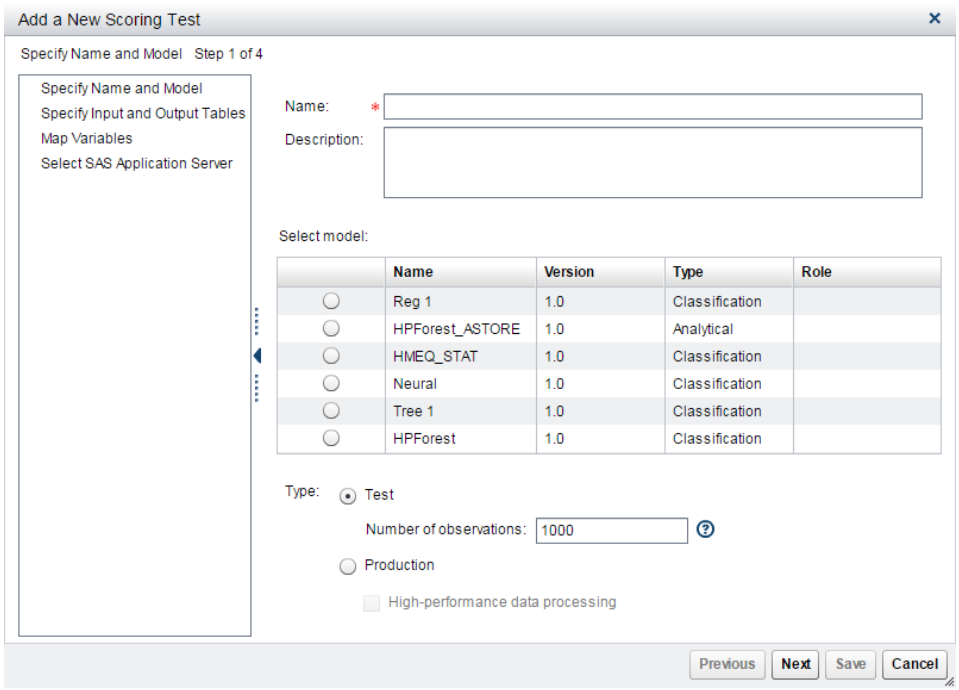
2. Enter a name for the scoring output table.
3. Select a library.
4. Select the input variables.
5. Select the output variables.
6. Select whether to add the model ID variable to the output table. The model UUID appears in all rows of the output table.

7. Select whether to use project's output variable names in the output table for model variables that are mapped to project variables.
8. Click **Add Variables**. The new output table variables appear below.
9. Click **OK**.
10. You must then make the new scoring output table available to SAS Decision Manager. For more information, see "Add Tables That Are Registered in Metadata" on page 48.

Create a Scoring Test

To create a scoring test:

1. On the **Scoring** page of a model, click ✚. The Add a New Scoring Test window appears.



2. Enter a name for the scoring test.
3. (Optional) Enter a description of the scoring test.
4. Select a model from the list.
5. Select a type of scoring test:
   - **Test**

     Specify the number of observations to be read from the scoring input table (default is 1000 rows).
   - **Production**

     To run the scoring test in a High-Performance Analytics production environment, select **High-performance data processing**.

Note: A best practice is to select **Test** before beginning all scoring tests. Later, when you are satisfied with the results of running the scoring test and you are ready to put the test into production, you can change the type to **Production**.

6. Click **Next**.
7. Specify an **Input table**. Use the default scoring input table that is set on the **Properties** page of a project, or select a new input table. To select a table, click **Browse** and select a table. Click **OK**.
8. Specify an **Output table**. To select a table, click **Browse** and select a table. Click **OK**.
9. Click **Next**.
10. For each scoring output variable, select an available variable. The scoring output variables are mapped automatically if their names match those in the **Available Variables** column.
11. Click **Next**.
12. Select a SAS Application Server from the list.
13. Click **Save**.

Execute a Scoring Test

To execute a scoring test:

1. Select a scoring test from the list and click .
2. Click the **Results** tab or double-click the scoring test to view the scoring test results.

   Note: You can check the status of a job by clicking and then selecting the **Tests** tab, the **Results** tab, or the **Job History** tab.

Schedule a Scoring Test

Instead of executing a scoring test, you can schedule a scoring test to run on a particular date and time. You can also schedule how often you want the scoring test to run. Advanced settings enable you to set the scheduling server, the batch server to run the scoring test, and the location of the scoring results.

Before you can schedule a scoring test, your user ID and password must be made available to the SAS Metadata Repository. You must also sign in to SAS Decision Manager using your full user credentials that were specified for your user account in SAS Management Console. For user accounts where a Microsoft Windows user ID is specified, you must enter your user ID in the format of *domain\userID*. Contact your system administrator to add or update your password, and to determine the correct user credentials for your account.

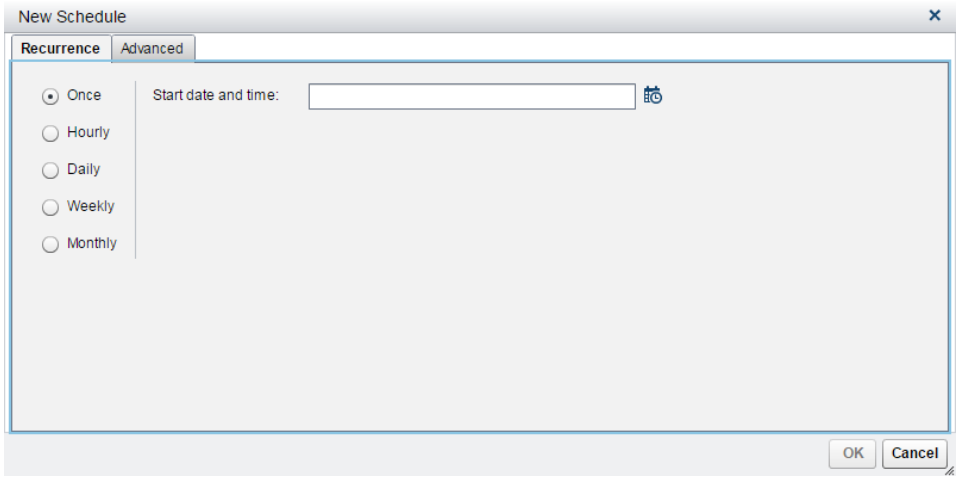
Note: You must have already created a scoring test before you can schedule a job to run the scoring test.

CAUTION:

Check your configuration settings before scheduling jobs. Users who are configured for single sign-on authentication by Kerberos cannot schedule jobs.

To schedule a scoring test:

1. Select the scoring test that you want to schedule from the list and click .



2. On the **Recurrence** tab, select the recurrence pattern.
3. Specify the criteria for when and how often the job should be run.
4. (Optional) Click the **Advanced** tab.
   a. Select the server that schedules the job from the **Scheduling server** list box.

      Note: The SAS Distributed In-Process Services Scheduling Server is not supported.

   b. Select the batch server that runs the job from the **Batch server** list box.
   c. Click **Browse** to select a location for the output and click **OK**.
5. Click **OK**.
6. After the job has been scheduled, a confirmation message appears. Click **Close**.
7. Click the **Job History** tab to view the job status.
8. Click the **Results** tab to view the scoring test results.

Note: Scoring test job schedules cannot be edited. To change the schedule, delete the schedule and create a new schedule.

To delete a schedule, select the schedule and then click .

Scoring Model Properties

Scoring Test Properties

Here is a list of the **Scoring test** properties that provide information that is specific to the scoring test.

Property Name	Description
Scoring test type	Specifies a value of **Test** or **Production** for the type of scoring test.
SAS Application Server	Specifies the name of the SAS Application Server to which SAS Decision Manager is connected. This value is taken from the SAS Metadata Repository.
Model	Specifies the name of the model whose score code is to be executed on the SAS Application Server. This value is set when the scoring test is created and cannot be modified.
Input table	Specifies the name of the input table (data source) to be used in scoring. This value is set when the scoring test is created and cannot be modified.
Output table	Specifies the name of the output table to be used in scoring. This value is set when the scoring test is created. If the scoring test type is **Test**, the output file is stored on the SAS Content Server. If the scoring test type is **Production**, then this setting identifies the output table where the results of the scoring are written.

Result Set Properties

The following property provides information that is specific to the scoring test.

Property Name	Description
Number of observations	When **Scoring test type** is set to **Test**, this property specifies how many observations are to be read from the scoring test input table. This setting enables you to limit the number of records that are written to the scoring test output table on the SAS Content Server in order to reduce operation costs. If a value is not specified, the default value of 1000 rows is used for the number of observations. When **Scoring test type** is set to **Production**, this property specifies how many observations are to be read from the scoring test input table and displayed when you select **Result Set** from the **Results** tab. The default value is 0, indicating that there is no limit. This value cannot be changed in SAS Decision Manager. The administrator can modify the value by using SAS Management Console. For more information, see *SAS Decision Manager: Administrator's Guide*.

Chapter 16

Using Reports to Evaluate and Validate Models

Overview of Model Comparison, Validation, and Summary Reports

What Are Model Comparison, Validation, and Summary Reports?

The SAS Decision Manager model comparison, validation, and summary reports are tools that you can use to evaluate and compare the candidate models in a version or across versions to help you select and approve the champion model that moves to production status. The model comparison reports are analytical tools that project managers, statisticians, and analysts can use to assess the structure, performance, and resilience of candidate models. The model validation reports use statistical measures to validate the stability, performance, and calibration of risk models and parameters. The training summary data set report creates frequency and distribution charts that summarize the train table variables.

The reports present information about a number of attributes that can affect model performance. Together, the reports provide qualified information that can serve as the analytical basis for choosing and monitoring a champion model.

Here is a description of the comparison reports:

Model Profile Report
For a single model, this report displays the profile data that is associated with input, output, and target variables. Profile data includes the variable name, type, length, label, SAS format, measurement level, and role.

Delta Report
This report compares the profile data for two models and notes the differences.

Dynamic Lift Report
The Dynamic Lift report provides visual summaries of the performance of one or more models for predicting a binary outcome variable.

Interval Target Variable Report
The Interval Target Variable report creates two plots for you to view the actual versus predicted values for a model and the actual versus residual values for a model. Interval Target Variable report can be created only for prediction models.

The following are the Basel III model validation reports:

Loss Given Default Report
The Loss Given Default (LGD) report calculates the amount that might be lost in an investment and calculates the economic or regulatory capital for Basel III compliance.

Probability of Default Model Validation Report
The Probability of Default (PD) Validation report estimates the probability of defaulting on a debt that is owed. Probability of default is used to calculate economic or regulatory capital for Basel III compliance.

The model validation reports use statistical measures that report on these model validation measures:

- The model stability measures track the change in distribution for the modeling data and scoring data.
- The model performance measures check the model's ability to distinguish between accounts that have not defaulted and accounts that have defaulted, as well as report on the relationship between actual default probability and predicted default probability.
- The model calibration measures check the accuracy of the selected models for the LGD and PD reports by comparing the correct quantification of the risk components with the available standards.

This is the train table data set summary report:

Training Summary Data Set Report
The Training Summary Data Set report creates frequency and distribution charts for a training data set.

After you execute a performance definition, you can generate performance monitoring results and compare the champion and challenger models:

Monitoring Report
After you execute a performance definition, SAS Decision Manager stores the output data sets in the project folder. You can format the performance monitoring results and then view the performance monitoring results report.

Champion and Challenger Report
After you execute a performance definition for the champion model, you can execute a performance definition for the challenger model using the same performance data sets. You can then create a Champion and Challenger Performance report that compares the performance of the two models.

You create the reports using the New Report window that you start from a project's **Reports** page.

Model Comparison, Validation, and Summary Report Input Files

SAS Decision Manager uses a test table as the input table for the Dynamic Lift report and the Interval Target Variable report.

Before you can create a Dynamic Lift report or the Interval Target Variable report, make sure that a test table has been added to the SAS Metadata Repository and registered in the Data Tables category or SAS Management Console. The test table can be viewed in the Data Tables category view. Then, specify the test table in the project property **Default test table**.

You specify the input table for validation reports in the New Report window. The input file for the validation reports can contain only input variables or it can contain input and output variables. If the input table contains input and output variables, the report generation does not need to run a scoring test to obtain the output variables.

When you create a train table summary report, the train table or specified input table is used to create the training summary data sets. The train table must be available in the SAS Metadata Repository. The train table must then be specified in the project property for the **Default train table**.

The Model Comparison, Validation, and Summary Report Output Files

The Reports page stores the model comparison, validation, and summary report output files in the **Model Evaluation** tab. The name of the report is the value of the **Name** box that you specified in the New Report window.

Each time you create a report, these files are generated:

- the report in either HTML, PDF, or RTF format

  Note: The Loss Given Default and Probability of Default Model Validation reports can be created only in PDF format.

- taskCode.log
- taskCode.sas

Here is a description of the model comparison output files:

Report File	Description
report-name.html	This file is the report output in HTML format.
report-name.pdf	This file is the report output in PDF format.
report-name.rtf	This file is the report output in RTF format.
taskCode.log	This file is the log file that contains messages from running the SAS code to create the report.
taskCode.sas	This file is the SAS code that is used to create the report.

After you create a report, you can view the report from the **Reports** page.

Note: If you are configured for single sign-on web authentication and you try to view a report, the report is downloaded to your local drive. To view an HTML report, you must unzip the file.

Model Profile Reports

About Model Profile Reports

A Model Profile report displays the profile data that is associated with the model input variables, output variables, and target variables. The report creates three tables, one each for the model input, output, and target variables.

Here is a description of the model profile data:

Profile Data	Description
Name	The name of the variable.

Profile Data	Description
Type	The data type of the variable: character (C) or numeric (N).
Length	The length of the variable.
Label	A label that is associated with the variable.
Format	The SAS format that is associated with formatting the variable.
Level	The measurement level: nominal, ordinal, interval, or binary.
Role	The type of variable: input, output, or target.

The reports are created using these auxiliary model files:

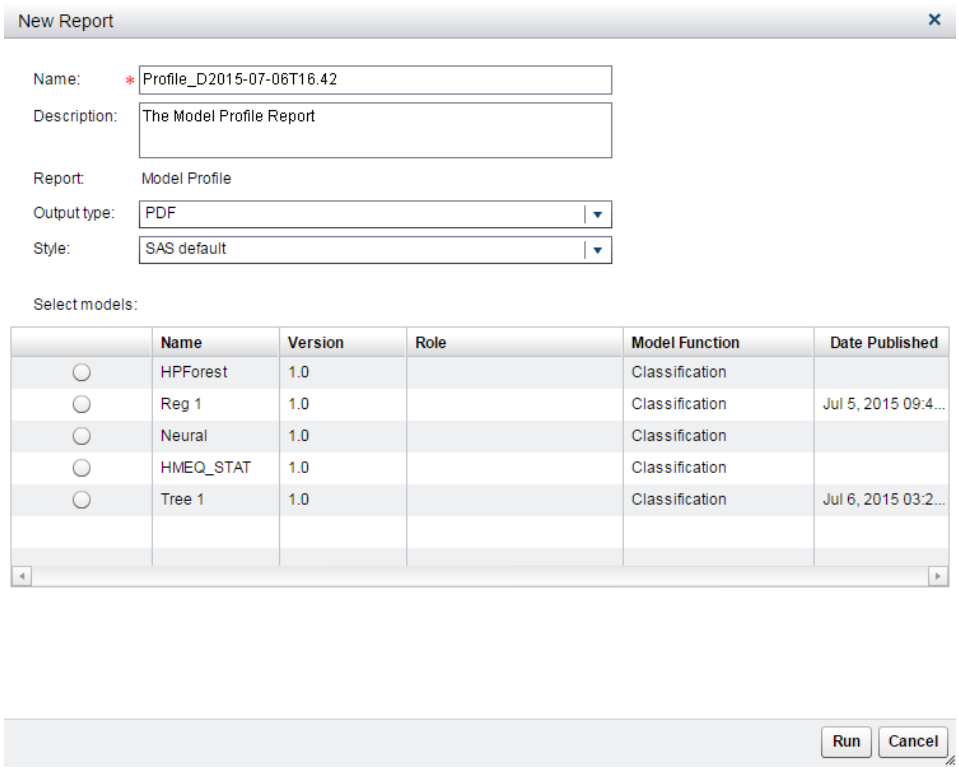
- inputvar.xml
- outputvar.xml
- targetvar.xml

Note: The target variable property must be set in order for a model to be able to run the Model Profile report.

Create a Model Profile Report

To create a Model Profile report:

1. Click and select **Model Profile**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. From the list, select the model that you want to include in the report.
6. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Delta Reports

About Delta Reports

A Delta report compares the input, output, and target variable attributes for each of the variables that are used to score two candidate models. Delta reports display the differences in the variables of competing candidate models. The report output is a table that groups the variables by the variable name. For each variable, the reports lists the attribute value for each model and whether the attribute value is the same or different from the other attribute values.

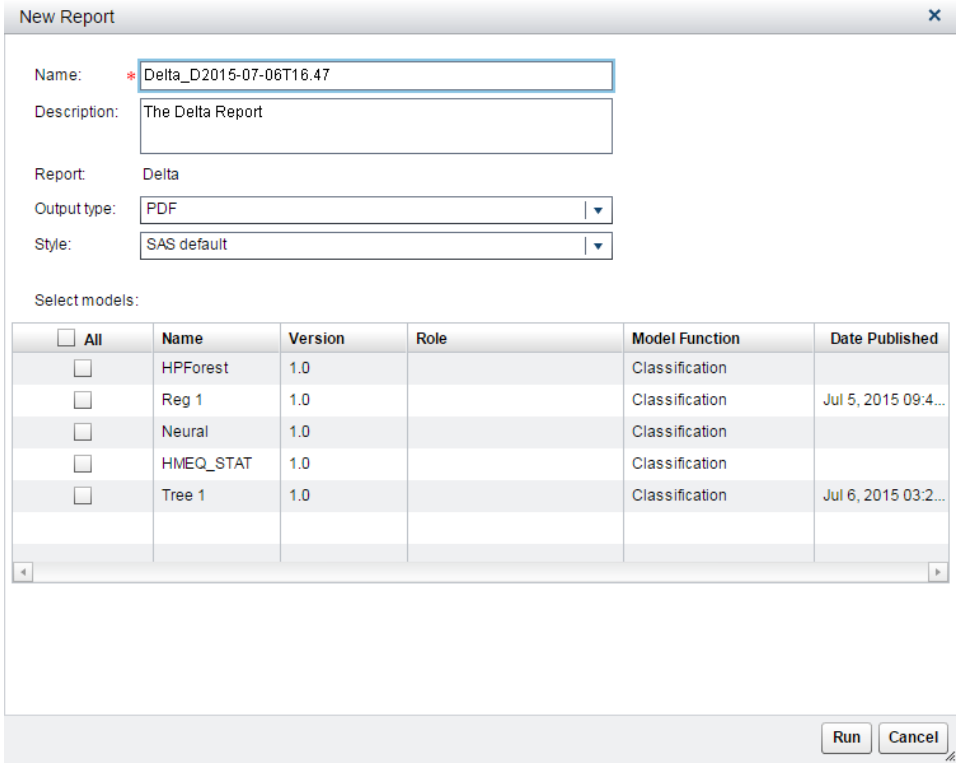
Here is a description of each of the columns in the output of a Delta report:

Column	Description
Role	Specifies the function that a variable performs in determining a score code.
Name	Specifies the name of the variable that is being compared.
Variable Attribute	Specifies the name of the variable attribute that is being compared.
Model Name-1	Contains the value of the attribute for the first model.
Model Name-2	Contains the value of the attribute for the second model.
Difference	Specifies an X if the value of the variable attribute is different from the value of the variable attributes in the other model. If the value of the variable attribute is the same, this column is blank.

Create a Delta Report

To create a Delta report:

1. Click and select **Delta**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. From the list, select the models that you want to include in the report.
6. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Dynamic Lift Reports

About Dynamic Lift Reports

The Dynamic Lift report enables you to view a model's lift at a given point in time or to compare the lift performance of several models on one chart. The Dynamic Lift report creates the following charts:

- Lift
- Cumulative Lift
- Percent Response
- Cumulative Percent Response
- Captured Response
- Cumulative Captured Response

A Dynamic Lift report can be created only for classification models with a binary target.

The charts that are created for a Dynamic Lift report are also created in the Monitoring Report, which creates multiple types of model comparison reports. Before you can create a Dynamic Lift report, certain project and model property settings must be set.

For models that are created with PMML 4.2, the **Valid variable name** option in SAS Management Console must be set to **Yes** by a SAS Decision Manager administrator. In addition, the PMML variable names cannot be more than 20 characters. For more information, see *SAS Decision Manager: Administrator's Guide*.

Verify Project and Model Property Settings

Verify Project Properties

Select the project name and verify that the following project properties are set:

Training target variable
Specifies the name of the target variable that was used to train the model. The model must have the same training target variable as the project.

Target event value
Specifies the value for the desired target variable event or state. For example, if a model predicts when RESPONSE=YES, then the target event value is **`YES`**.

Output event probability variable
Specifies the name of the output event's probability variable.

Verify Model Properties

For each model in the Dynamic Lift report, open the model and verify the following properties on the **Model Properties** page:

Properties

Property	Description
Target variable	Specifies the name of the target variable. For example, if a model predicts when RESPONSE=YES, then the target variable is **`RESPONSE`**.
Score code type	Specifies whether the score code type is Analytic store, DS2, DATA step, PMML, or SAS program.

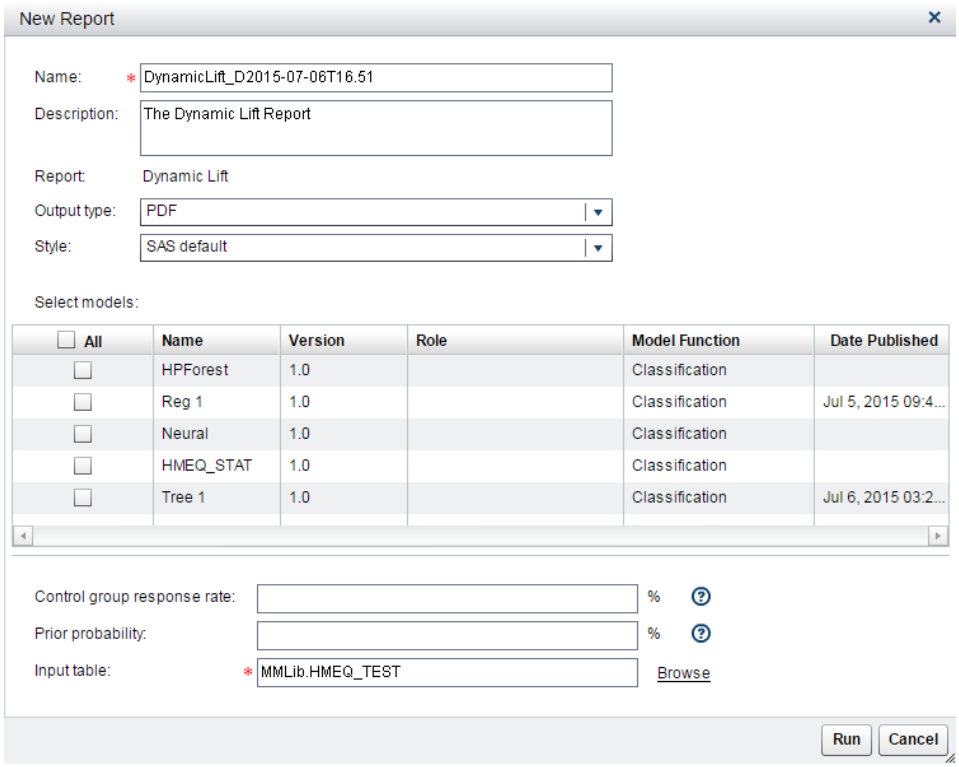
Note: Dynamic Lift reports are not applicable to models whose **Score code type** property has a value of PMML or DS2. A Dynamic Lift report can be created for a PMML model that was created using PMML 4.2 and whose **Score code type** is DATA step.

Create a Dynamic Lift Report

After ensuring that the appropriate project and model properties have been set, create the report.

To create a Dynamic Lift report:

1. Click and select **Dynamic Lift**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. From the list, select the models that you want to include in the report.
6. Specify the **Control group response rate**. The control group response rate calculates the adjusted lift values for a model. If the control group response rate is not specified, the default response rate in the test table is used to calculate the adjusted lift values.
7. Specify the **Prior probability**. The prior probability is the proportion of event observations to the total observations in the whole population. In this case, the whole population is the entire train table. Specify a value for the prior probability to be used as the true event proportion when assessment values are computed for the lift of a model.
8. Accept the default value for **Input table** or click **Browse** to navigate to the appropriate folder to select an input table. Click **OK**.
9. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Interval Target Variable Report

About Interval Target Variable Reports

The Interval Target Variable report creates two plots for you to view the actual versus predicted values for a model and the actual versus residual values for a model. The Interval Target Variable report can be created only for prediction models. Before you can create an Interval Target Variable report, certain project and model property settings must be set.

Verify Project and Model Properties

Before you can run an Interval Target Variable report, you must set the following project properties:

Default test table
Specifies a test table that is registered in the SAS Metadata Repository. You can view the table in the Data category view. The test table must contain the target variable, as well as values for the variables that are defined by the project input variables.

Training target variable
Specifies the name of the target variable that was used to train the model. The model must have the same training target variable as the project.

Output prediction variable
Specifies the name of the output prediction variable.

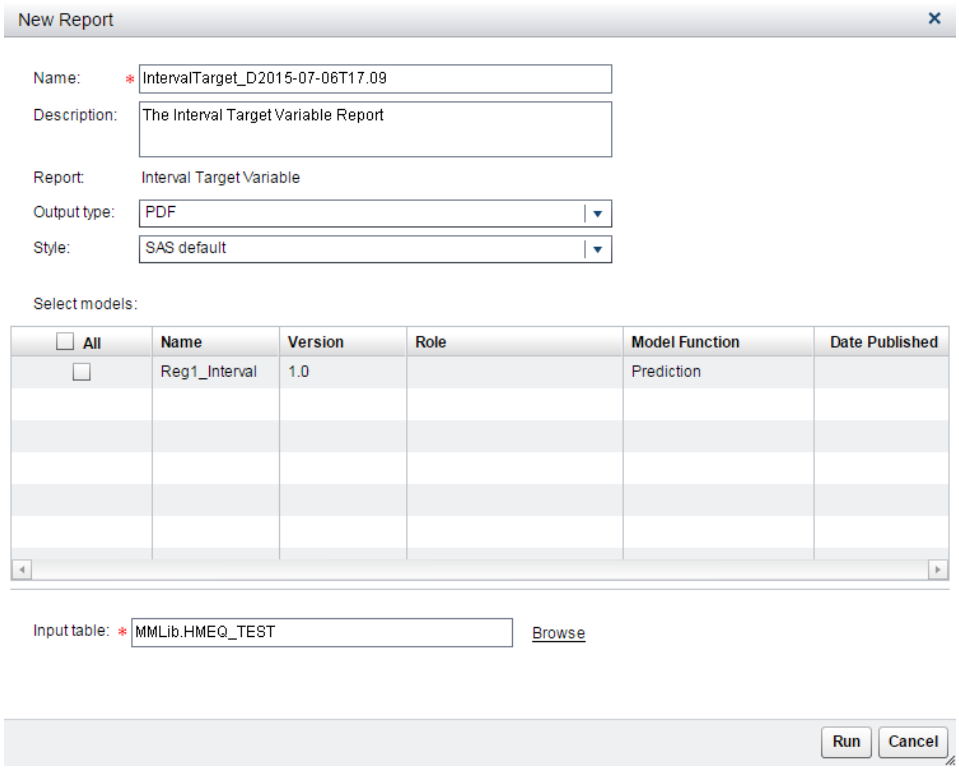
To verify the model mapping, select and open the model from the Models page. Select **Model Properties** ⇨ **Variables** to verify that the model variables are mapped to the project variables. If the variable names are the same, you do not need to map the variables. If they are not mapped, for each project variable, select the project variable and select a variable name.

Create an Interval Target Variable Report

You must have a prediction model with an interval target variable in order to create the Interval Target Variable report. After ensuring that the appropriate project properties have been set and the model mapping is set, create the report.

To create an Interval Target Variable report:

1. Click and select **Interval Target Variable**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. From the list, select the models that you want to include in the report.
6. Accept the default value for **Input table** or click **Browse** to navigate to the appropriate folder to select an input table. Click **OK**.
7. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Loss Given Default Reports

About Loss Given Default Reports

Loss Given Default (LGD) models help validate the stability, performance, and calibration of models with the following statistical measures and tests:

Model stability measures
: The model stability measures track the change in distribution of the modeling data and the scoring data.

Model performance measures
: The model performance measures report this information:

- The model's ability to discriminate accounts that have defaulted with those that have not defaulted. The score difference between the accounts that default and those that do not helps determine the cut-off score, which is used to predict whether a credit exposure is a default.
- The relationship between the actual default probability and the predicted probability. This information is used to understand a model's performance over a period of time.

Model calibration measures
: The model calibration measures check the accuracy of the LGD models by comparing the correct quantification of the risk components with the available standards.

For a description of the statistical measures, see "Statistical Measures Used in Basel III Reports" on page 375.

The Loss Given Default Report Properties

In order to create the reports, SAS Decision Manager must know the input and output variables for the model. The input table can contain only input variables, or it can contain input and output variables. If the input table contains only input variables, a scoring test must be run to obtain the output variable. If the input table contains the input and output variables, no scoring is necessary. You specify whether a scoring test must be run by setting the **Run score code** property in the New Report window. If the input table contains output variables, the value of the **Run score code** can be **No**. If the input table contains only input variables, the **Run score code** property must be set to **Yes**.

The report properties require the names of the variables from the input and output tables in order to map these variables to variables that are used to create the reports. The LGD report properties map these variables:

Time period variable	specifies the variable that is used to indicate a time period. The first time period begins with 1 and typically increments by 1. The default is **period**.
Time label variable	(optional) specifies a label for the time period. If this variable exists in the input table, the report output contains a table that maps time periods to time labels.
Actual variable	specifies the actual LGD variable. The default is **lgd**.
Predicted variable	specifies the output prediction variable that is used only if scoring for the report is not performed by SAS Decision Manager. If the report scoring is done by SAS Decision Manager, this variable should be excluded by the input data set. The default is **p_lgd**.
Pool variable	specifies the variable that names pool IDs. The default is **pool_id**.

Prerequisites for Loss Given Default Reports

Before you run an LGD report, select the project name and verify that the following project properties are set:

Training target variable
Specifies the name of the target variable that was used to train the model. The model must have the same training target variable as the project.

Model function
Specifies the type of model function. For an LGD report, the model function must be Prediction.

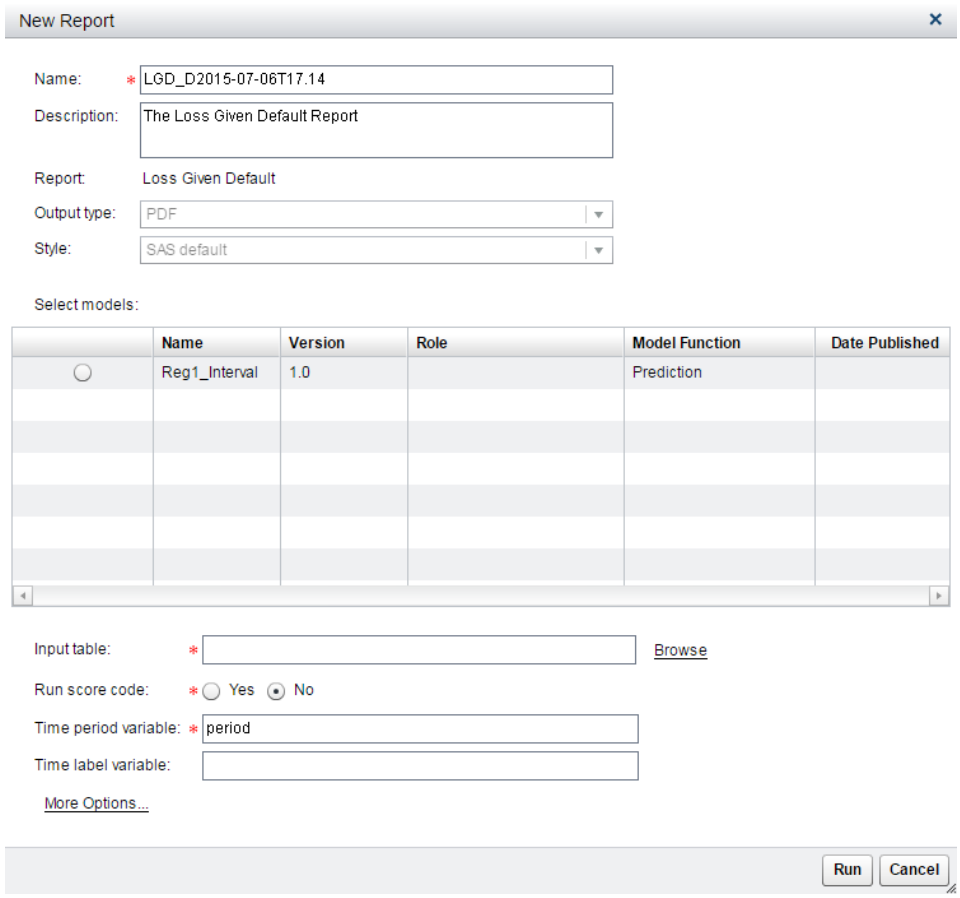
Class target level
Specifies an **Interval** class target level.

Output prediction variable
Specifies the name of the output prediction variable.

Create a Loss Given Default Report

To create a Loss Given Default report:

1. Click and select **Loss Given Default**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.

   Note: The default output type is PDF.

3. From the list, select the model that you want to include in the report.

4. Select an **Input table**. Click **Browse** to navigate to the appropriate folder. Select an input table and click **OK**. The table can contain only input variables or it can contain input and output variables.

5. Select whether to run the score code. If the input table contains only input variables, set **Run score code** to **Yes**. If the input table contains output variables, set **Run score code** to **No**.
6. The **Time period variable** specifies the variable from the input table whose value is a number that represents the development period. This value is numeric. The time period for PD reports begin with 1. The default is **period**.
7. (Optional) In the **Time label variable** field, enter the variable from the input table that is used for time period labels. When you specify the time label variable, the report appendix shows the mapping of the time period to the time label.
8. Click **More Options** to set the following:

   Actual variable
   Specifies the actual LGD variable. The default is **lgd**.

   Predicted variable
   Specifies the project scoring output variable. If the scoring for the LGD report is performed outside SAS Decision Manager, the input data set must include this variable. If the scoring for the LGD report is done by SAS Decision Manager, the input data set should not include this variable. The default is **p_lgd**.

   Pool variable
   Specifies the variable from the input table that is used to identify a two-character pool identifier. The default is **pool_id**.

   Note: The variable names that you specify can be user-defined variables. A variable mapping feature maps the user-defined variables to required variables.
9. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Probability of Default Model Validation Reports

About Probability of Default Model Validation Reports

Probability of Default (PD) models help validate the stability, performance, and calibration of models with the following statistical measures and tests:

Model stability measures
The model stability measures track the change in distribution of the modeling data and the scoring data.

Model performance measures
The model performance measures report this information:

- The model's ability to discriminate accounts that have defaulted with those that have not defaulted. The score difference between the accounts that default and those that do not helps determine the cut-off score, which is used to predict whether a credit exposure is a default.
- The relationship between the actual default probability and the predicted probability. This information is used to understand a model's performance over a period of time.

Model calibration measures
: The model calibration measures check the accuracy of the PD model by comparing the correct quantification of the risk components with the available standards.

For a description of the statistical measures, see "Statistical Measures Used in Basel III Reports" on page 375.

Default Model Validation Report Properties

In order to create the reports, SAS Decision Manager must know the input and output variables for the model. To run the reports, the New Report window requires the name of an input table. The input table can contain only input variables, or it can contain input and output variables. If the input table contain only input variables only, a scoring test must be run to obtain the output variable. If the input table contains the input and output variables, no scoring is necessary. You specify whether a scoring test must be run by setting the **Run score code** property in the New Report window. If the input table contains output variables, the value of the **Run score code** can be **No**. If the input table contains only input variables, the **Run score code** property must be set to **Yes**.

The report properties require the names of the variables from the input and output tables in order to map these variables to variables that are used to create the reports. The report properties map these variables:

Time period variable	specifies the variable that is used to indicate a time period. The first time period begins with 1 and typically increments by 1. The default is **period**.
Time label variable	(optional) specifies a label for the time period. If this variable exists in the input table, the report output contains a table that maps time periods to time labels.
Scorecard bin variable	specifies the scoring output variable that names the scorecard bins. The input table must include this variable if scoring for the PD report is performed outside SAS Decision Manager. If scoring is done by SAS Decision Manager, do not include this variable in the input data set. The default is **scorecard_bin**.
Scorecard points variable	specifies the scoring output variable that names the scorecard points. The input table must include this variable if scoring for the PD report is performed outside SAS Decision Manager. If scoring is done by SAS Decision Manager, do not include this variable in the input data set. The default is **scorecard_points**.
Cut-off value	specifies the variable that is used to derive whether a credit exposure is a default. The cut-off value is also used to compute accuracy, sensitivity, specificity, precision, and error rate measures. You can use the score difference between accounts that default on loans and those that do not default on loans to determine the cut-off value. The default is **100**.

Prerequisites for Probability of Default Model Validation Reports

Before you can create a Probability of Default Model Validation report, verify that the following project settings are specified and that the output variables have been mapped:

Training target variable
: Specifies the name of the target variable that was used to train the model. The model must have the same training target variable as the project.

Class target level
Specifies a **Binary** class target level.

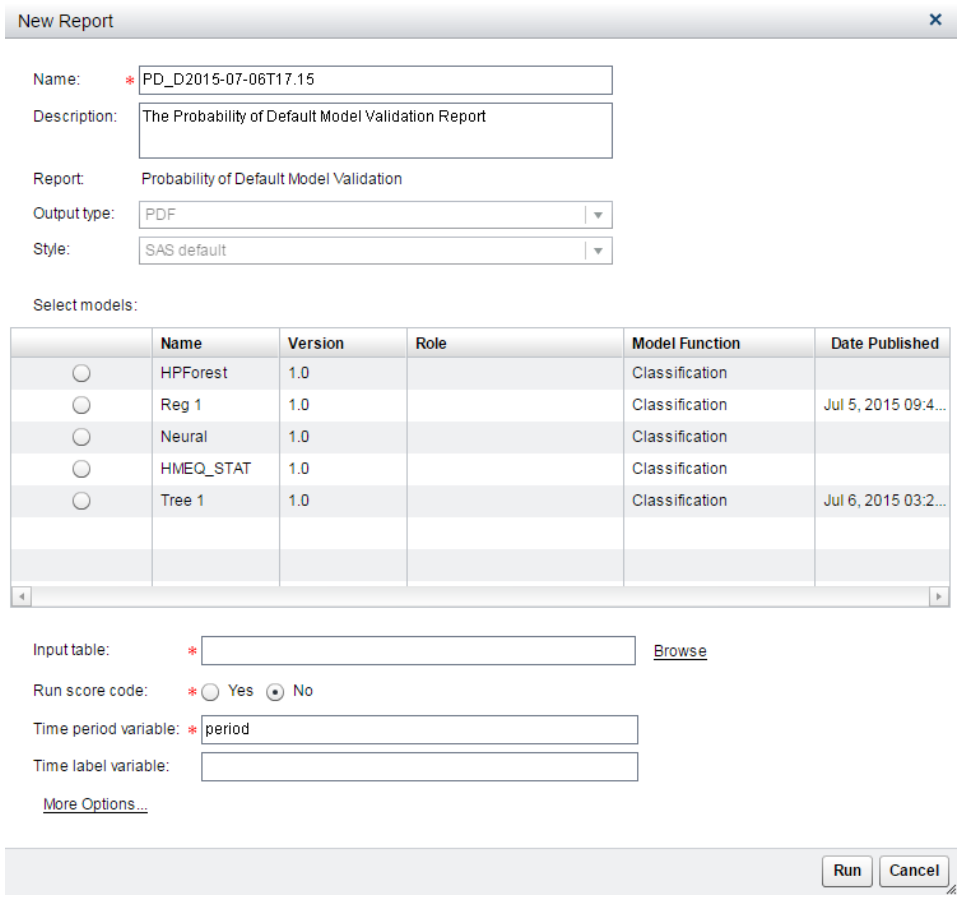
Output event probability variable
Specifies the name of the output event probability variable.

Model function
Specifies the type of model function. The model function must be `Classification` in order to create a PD report.

Create a Probability of Default Model Validation Report

To create a Probability of Default Model Validation report:

1. Click and select **Probability of Default Model Validation**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.

   Note: The default output type is PDF.

3. From the list, select the model that you want to include in the report.

4. Click **Browse** to navigate to the appropriate folder and select an input table and click **OK**. The table can contain only input variables or both input and output variables.

   Note: When a scoring input table for a PD report contains data and one or more time periods do not contain default or non-default loan information, these time periods are not used to calculate the PD measurements. In a chart, time periods that are not used to calculate the PD measurements are represented with dashed lines.

5. Select whether to run the score code. If the input table contains only input variables, set **Run score code** to **Yes**. If the input table contains output variables, set **Run score code** to **No**.
6. The **Time period variable** specifies the variable from the input table whose value is a number that represents the development period. This value is numeric. The time period for PD reports begin with 1. The default is **period**.
7. (Optional) In the **Time label variable** field, enter the variable from the input table that is used for time period labels. When you specify the time label variable, the report appendix shows the mapping of the time period to the time label.
8. Click **More Options** to set the following:

   Scorecard bin variable
   Specifies the variable from the input table that contains the scorecard bins. If the scoring job for the PD report is run outside SAS Decision Manager, the scorecard bin variable must be a variable in the input table. If scoring is done within SAS Decision Manager, do not include the variable in the input table. The default is **scorecard_bin**.

   Scorecard points variable
   Specifies the variable that contains the scorecard points. If the scoring job for the PD report is run outside SAS Decision Manager, the scorecard points variable must be a variable in the input table. If scoring is done within SAS Decision Manager, do not include the variable in the input table. The default is **scorecard_points**.

   Cut-off value
   Specifies the maximum value that can be used to derive the predicted event and to further compute accuracy, sensitivity, specificity, precision, and error rate. The default is **100**.
9. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Training Summary Data Set Reports

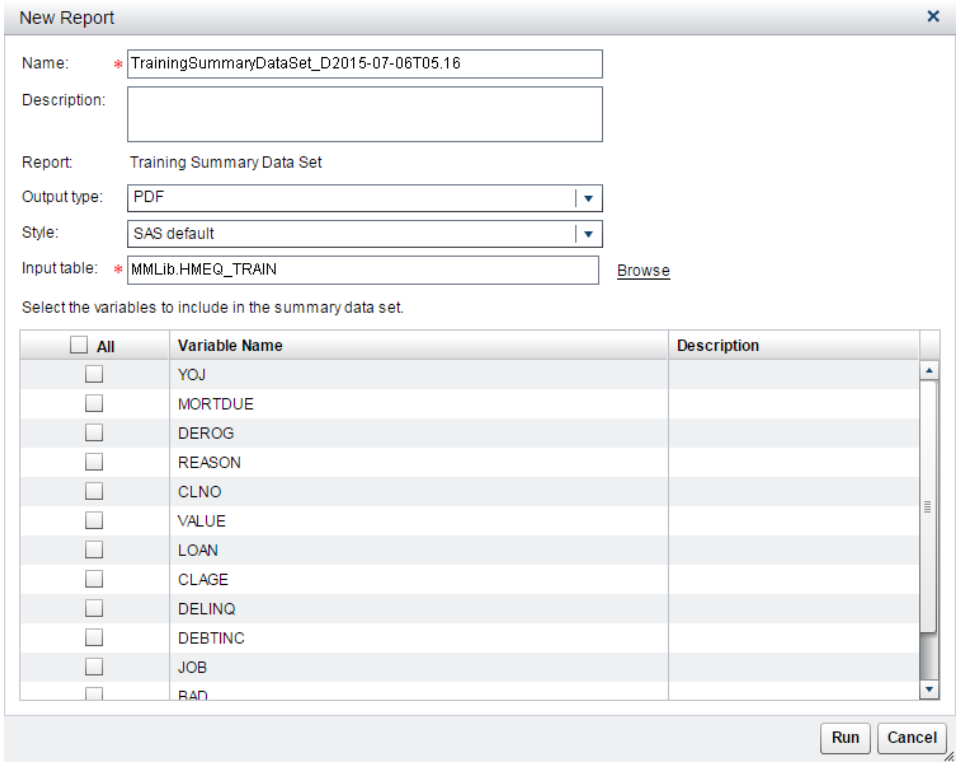
About Training Summary Data Set Reports

A Training Summary Data Set report creates frequency and distribution charts that summarize the train table variables. Using the default train table, SAS Decision Manager generates data sets that contain numeric and character variable summaries, and variable distributions. These data sets are used to create the summary report. Before you can create the report, you must generate the training summary data sets.

Create a Training Summary Data Set Report

To generate a training summary data set report:

1. Click and select **Training Summary Data Set**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. Select an **Input table**. Click **Browse** to navigate to the appropriate folder and select an input table. Defaults to the value of the default train table project property.
6. Select the variables to include in the summary data set.
7. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Monitoring Reports

About Monitoring Reports

After you execute a performance definition or run the %MM_RunReports() macro in production mode, as a batch job, SAS Decision Manager stores the output data sets on the SAS Content Server. You can view the performance monitoring results on the **Performance Results** tab or on the **Attachments** page.

When you create monitoring reports using the New Report window, the report creates the following charts:

Assessment charts

Assessment charts summarize the utility that you can expect by using the respective models, as compared to using only baseline information. Assessment charts can present a model's lift at a given point in time or the sequential lift performance of a model's lift over time. A monitoring report creates the following assessment charts:

- Lift
- Cumulative Lift
- Percent Response
- Cumulative Percent Response
- Captured Response
- Cumulative Captured Response
- Actual versus Predicted for prediction models
- Actual versus Residual for prediction models
- Population Stability Trend for prediction models

Assessment charts are created for the Monitoring Report.

Lift Trend chart

A Lift Trend chart displays the cumulative lift of the champion model, over time.

Gini - ROC chart

Sensitivity is the proportion of true positive events, and specificity is the proportion of true negative events. The Gini - ROC chart plots Sensitivity on the Y axis and 1 - Specificity on the X axis.

Gini - Trend Chart

When the Gini - ROC chart is created, the Gini index for each ROC curve is also created. The Gini index represents the area under the ROC curve and is a benchmark statistic that can be used to summarize the predictive accuracy of a model. The Gini - Trend chart plots a model's Gini index scores over time, and these are used to monitor model degradation over time.

KS Chart

The KS chart uses the Kolmogorov-Smirnov statistic to measure the maximum vertical separation, or deviation between the cumulative distributions of events and non-events.

KS Trend Chart

When you create a Kolmogorov-Smirnov report, the underlying KS statistic and the corresponding probability cutoff are read from a summary data set in the Resources folder. The KS Trend chart uses a summary data set that plots the KS Statistic over time. The KS Trend chart is used to monitor model degradation over time.

Actual versus Predicted

You use the Actual versus Predicted plot to see how predicted values match actual values.

Actual versus Residual

You use the Actual versus Residual plot to determine how good the model is at predicting values by examining errors and error trending, and comparing them to the actual values.

Population Stability Trend

The Population Stability Trend chart measures the shift of the scoring output variable distribution over time. Scoring output that is based on a development sample is used

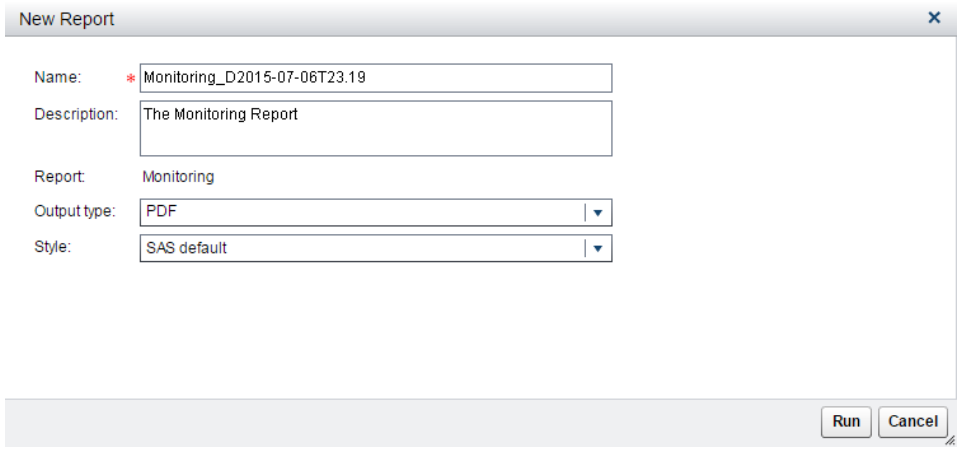
as the baseline distribution. The deviation index is used to indicate the shift for a given point in time.

Before you create a Monitoring Report or a Champion and Challenger Performance Report, you must ensure that certain project and model properties are set. For more information, see "Verify Project and Model Property Settings" on page 208.

Create a Monitoring Report

To create a Monitoring report:

1. Click and select **Monitoring**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Champion and Challenger Performance Reports

About the Champion and Challenger Performance Report

After you execute a performance definition for the champion model, you can execute a performance definition for the challenger model using the same performance data sets. SAS Decision Manager updates the output data sets with the performance data for the challenger model. You can create a Champion and Challenger Performance report that compares the performance of the two models.

The Champion and Challenger Performance report contains these charts:

Number of Predictors Exceeding Deviation Threshold
: This characteristic report creates a chart for each index that exceeds a deviation threshold (either 0.1 or 0.25) as indicated in the define performance definition. The characteristic report detects shifts in the distribution of input variables over time.

Lift Trend Chart
: A Lift Trend chart displays the cumulative lift of the champion model over time.

Gini - Trend
: When the Gini - ROC Chart is created, the Gini index for each ROC curve is also created. The Gini coefficient represents the area under the ROC curve and is a benchmark statistic that can be used to summarize the predictive accuracy of a model. The Gini - Trend Chart plots a model's Gini index scores over time, and these are used to monitor model degradation over time.

Gini - ROC Chart
: Sensitivity is the proportion of true positive events, and specificity is the proportion of true negative events. The Gini - ROC Chart plots Sensitivity on the Y axis and 1 - Specificity on the X axis.

KS Trend Chart
: When you create a Kolmogorov-Smirnov report, the KS statistic and the corresponding probability cutoff are computed for each Kolmogorov-Smirnov table. The KS Trend Chart uses a summary data set that plots the KS Statistic and the probability cutoff values over time. The KS Trend Chart is used to monitor model degradation over time.

KS Chart
: The KS Chart uses the Kolmogorov-Smirnov statistic to measure the maximum vertical separation, or deviation between the cumulative distributions of events and non-events.

Score Histogram
: The Score Histogram compares the scoring result distribution at different time periods using a histogram.

Score Distribution Line Plot
: The Score Distribution Line Plot compares the scoring result distribution at different time periods using a line plot.

Before you create a Champion and Challenger Performance report, verify the performance data and model status.

Verify Performance Data and Model Status

Before you can create a Champion and Challenger Performance report:

1. Select the **Models** page and verify that a champion model has been set. The champion model is designated as **Champion** in the **Role** column. If a champion has not been set, select a model from the list, and click ✔ to set the model as the project champion model.
2. Ensure that a challenger model is flagged. The challenger model is designated as **Challenger** in the **Role** column. If it is not, select a model from the list, and click ⚑ to flag a model as a challenger to the project champion model.
3. Verify that performance monitoring data is available for the champion model and the challenger model. Performance monitoring results must exist for the same performance data using the same time periods and data labels. Navigate to **Performance** ⇨ **Results** ⇨ **Data Sets** and select the file **jobstatus.sas7bdat**. The **Content** tab displays performance monitoring status data.

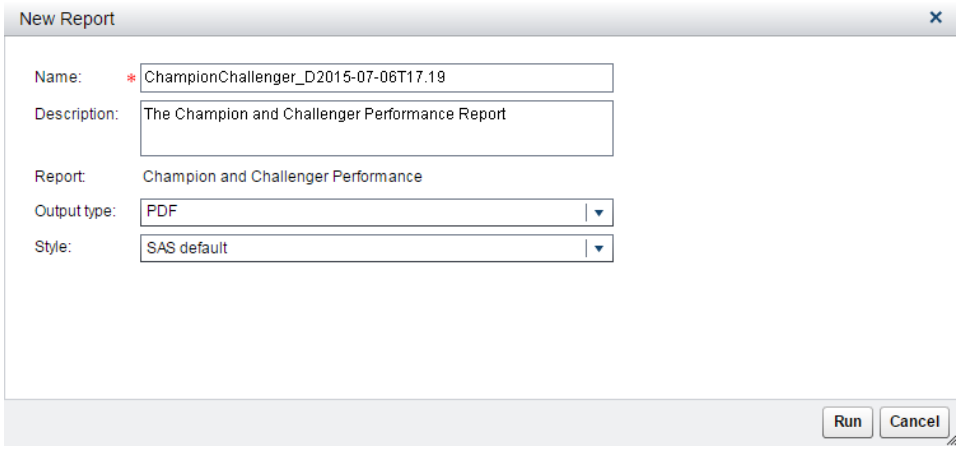
a. Verify that the UUIDs for the champion and challenger models are in the **Model UUID** column.

b. Using the **name** column and the **time** column, verify that matching date labels exist for the champion and challenger models for each type of report. If there are multiple date labels for a model for any given report, SAS Decision Manager uses the most recent job.

Create a Champion and Challenger Performance Report

To create a champion and challenger performance report:

1. Click and select **Champion and Challenger Performance**. The New Report window appears.



2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

View Reports

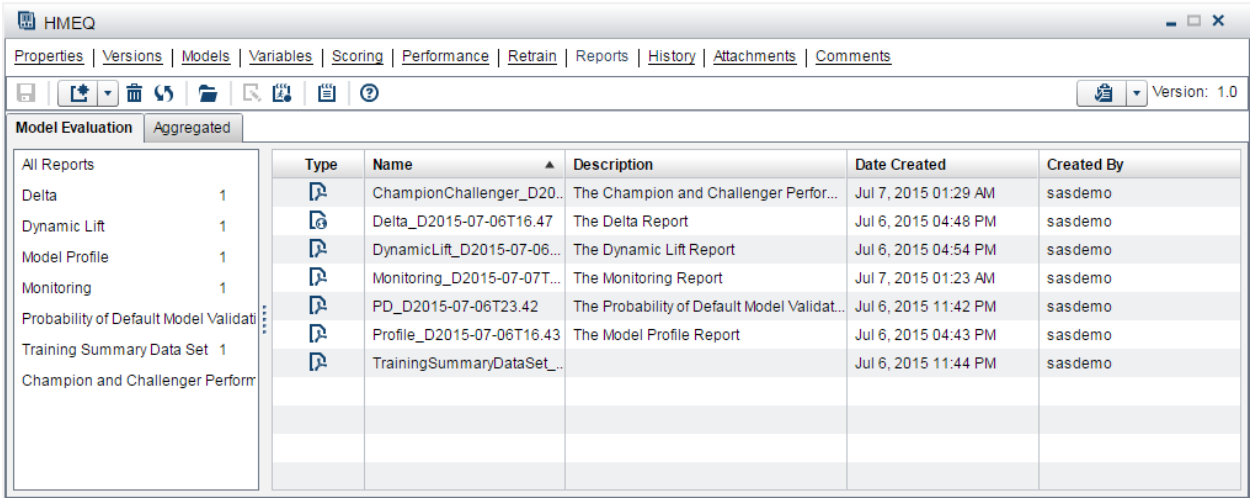
To view a report:

1. On the **Reports** page, in the **Model Evaluation** tab, select a type of report from the left navigation menu.
2. You can view a report in several ways:
   - Double-click a report in the list.
   - Select a report from the list and click .

- Right-click a report from the list and select **Open**.

  Note: You can also view the SAS code and SAS log.

Note: If you are configured for single sign-on web authentication and you try to view a report, the report is downloaded to your local drive. To view an HTML report, you must unzip the file.



Chapter 17

Validating Models Using User Reports

Overview of User Reports

Ad Hoc Reports and User-Defined Reports

User reports are SAS programs that you create and import to SAS Decision Manager so that you can customize reports to meet your business requirements. The ad hoc report enables you to develop, test, and run your report within SAS Decision Manager. The user-defined report can be developed either within or external to SAS Decision Manager. It requires a SAS program and the associated auxiliary files to be installed in a directory that is available to SAS Decision Manager. Using ad hoc reports, you modify and submit your code from the SAS Editor within the Create an Ad Hoc Report window.

A user-defined report is a report that is available for reporting on all models in SAS Decision Manager. The user-defined report requires three files to be installed in your server's file structure:

- a SAS program to create the report

- a report template XML file that specifies the report requirements, such as report name and the number of required models to run the report
- a SAS program file that lists the SAS Decision Manager global macro variables and macros that are used in your report

After you have these three files, you use the Manage Templates function to upload the files to the SAS Content Server.

The ad hoc report can be used to develop, test, and debug user-defined reports. When your ad hoc report is ready for a production environment, you can create the report template XML file and the macro file, and install the three files in the user-defined report file structure.

Comparison of Ad Hoc and User-Defined Reports

Report Difference	Ad Hoc Report	User-Defined Report
Version	An ad hoc report is defined and can be run only under the version where it was created.	A user-defined report can be run under any project version.
Report template	An ad hoc report does not require a template.	A user-defined report requires a template to define the report parameters.
Report results	Each time an ad hoc report is run, the existing report is overwritten.	Each time a user-defined report is run, a new report is created on the **Reports** page.
Location of SAS files used to generate the report	The ad hoc report SAS program is stored on the **Reports** page for the version where it was created.	The user-defined report SAS files are uploaded to the SAS Content Server.

Output Created by User Reports

The first time you create a report, SAS Decision Manager creates a report on the **Reports** page.

Each time you create a new ad hoc report, the following files are created:

- the report in either HTML, PDF, or RTF format
- smm_userCode.sas
- taskCode.log
- taskCode.sas

Each time you create a new user-defined report, the following files are created:

- the report in either HTML, PDF, or RTF format
- taskCode.log
- taskCode.sas

CAUTION:

The wizard overwrites the output files if an output file of the same name already exists.

Here is a description of the ad hoc report output files:

Report File	Description
report-name.html	This file is the report output in HTML format.
report-name.pdf	This file is the report output in PDF format.
report-name.rtf	This file is the report output in RTF format.
smm_userCode.sas	This file contains the SAS program report code that was submitted in the Create an Ad Hoc Report window.
taskCode.log	This file is the log file that contains messages from running the SAS code to create the report.
taskCode.sas	This file is the SAS code that is used to create the report. The file contains the user-defined report code as well as code that was generated by SAS Decision Manager to create the report.

You can see the contents of these files by selecting them on the **Reports** page. You can also see the taskCode.sas file and the taskCode.log files.

Ad Hoc Reports

Overview of Ad Hoc Reports

To create an ad hoc report, you must first write a SAS report program. When the report code is ready, you copy your code to the **SAS Editor** tab in the Create an Ad Hoc Report window. You then submit your program. Unlike the user-defined report, the ad hoc report does not require auxiliary files to be uploaded to the SAS Content Server.

To create your report output in either HTML, PDF, or RTF, or to specify a style other than the default style for your report, you modify your report with code that is provided by SAS and that enables you to specify the report output format and style. The code that you need to add to your program is included in the steps to create an ad hoc program.

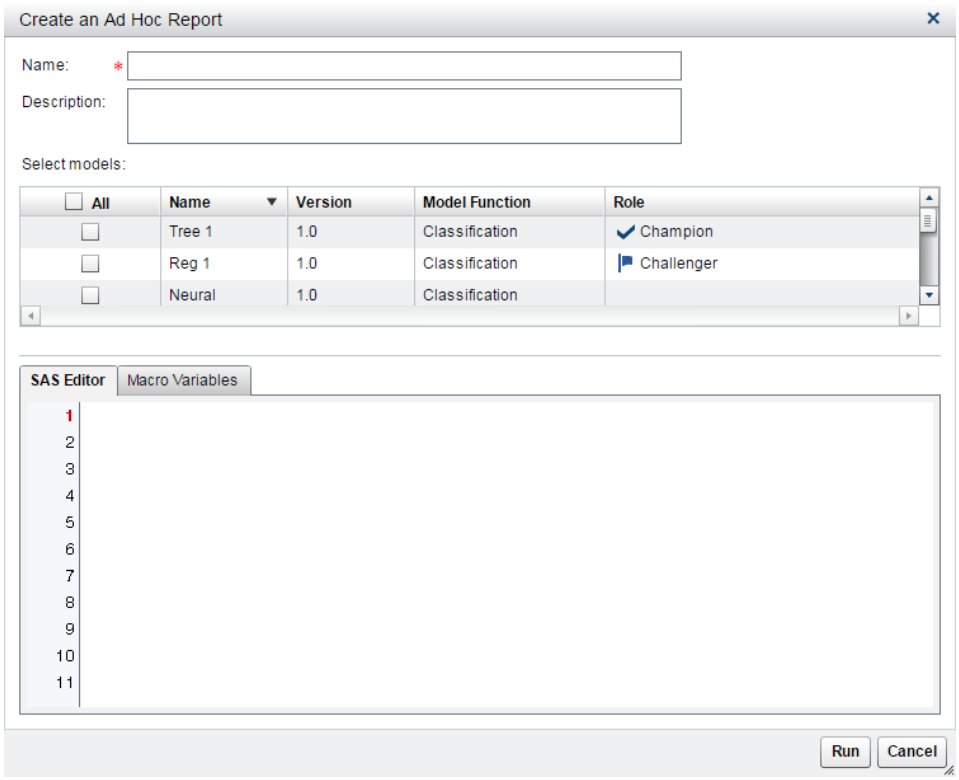
If you find an error in your report code, you must delete the report in the project, fix your code in your source file, and submit the code in the Create an Ad Hoc Report window again.

Create an Ad Hoc Report

To create an ad hoc report, you must first create a SAS program. Test your program in SAS before you run your program as an ad hoc report. After the code runs successfully, you can create the ad hoc report.

To create an ad hoc report:

1. On the **Reports** page, click and select **Ad Hoc**. The Create an Ad Hoc Report window appears.



2. Enter a name and an optional description for the report.
3. Select one or more models.
4. Add or copy SAS code to the **SAS Editor** tab. Make sure that your report program is enclosed by the SAS code that defines the report output format. Click the **Macro Variables** tab to view a list of the variables that can be accessed by your program.
5. Click **Run**. The report is generated and appears in the default viewer for the selected output type.
6. The report appears in a list on the **Model Evaluation** reports tab.

Example Ad Hoc Report

The following example code lists the score results in an HTML output format:

```
Filename mmreport catalog "sashelp.modelmgr.reportexportmacros.source";
%include mmreport;

%MM_ExportReportsBegin(reportFormat=html, reportStyle=Meadow, fileName=PerfDS);
proc print data=myTable.scoretable;
var loan delinq score;
run;
quit;


%MM_ExportReportsEnd(reportFormat=html);
```

After you click **Run**, the report is created and placed on the **Reports** page. The following HTML output displays selected rows of the output.

Obs	LOAN	DELINQ	score
1	1100.00	0	0.08918
2	162.06	2	0.08918
3	1292.02	0	0.08918
4	783.13	.	0.08918
5	1700.00	0	0.08918

User-Defined Reports

Overview of User-Defined Reports

User-defined reports require the following files to be uploaded to the SAS Content Server:

- the SAS program that creates the report.
- a SAS program file that lists the SAS Decision Manager global macro variables that are used in your report.
- a report template XML file that specifies the report requirements, such as report name and the number of required models to run the report.

After these three files have been uploaded to the SAS Content Server, the user-defined report type is included as a report type in the new report drop-down list on the **Reports** page.

The New Report window includes controls to specify the type of output that the report creates, such as HTML or PDF, and a style for the report. You can modify your report to include the SAS code so that the New Report window offers the report output controls for your report.

Create a User-Defined Report

To create a user-defined report:

1. Write and test your SAS program that creates a report.
2. To format the output for a user-defined report, add the SAS code below to your report code in order to select the **Output type** and the **Style** in the New Report window. The **Output type** enables you to select a report output format of HTML, PDF, or RTF. The **Style** enables you to select a report output style for your report.

   Replace *report-name* with the name of your user-defined report. The name can contain letters, the underscore (_), hyphen (-), and the period (.). End your user-defined report with the %MM_ExportReportsEnd macro.

   ```
   Filename mmreport catalog "sashelp.modelmgr.reportexportmacros.source";
   %include mmreport;
   %MM_ExportReportsBegin(fileName=report-name);
   ```

```
            …
    your-user–defined-code
            …
%MM_ExportReportsEnd;
```

3. In the report XML file, add this SAS program name to the FILENAME= argument of the <Code> element (for example, **<Code filename="myUserReport.sas"/>**). For more information, see "Report Templates" on page 231.

For an example of a report, see "Example User-Defined Report" on page 235.

Defining Macro Variables for a User-Defined Report

Executing a user-defined report requires a SAS program that lists the report code's macro variables. If you do not have macro variables in your report, create a SAS program file with a comment in it. This file is required.

Here is an example program to define macro variables:

```
%let _MM_User=miller;
%let _MM_Password=Rumpillstillskin3;
```

In the report XML file, add this SAS program name to the FILENAME= argument of the <PreCode> element (for example, **<PreCode filename="myMacroDefs.sas"/>**). For more information, see "Report Templates" on page 231.

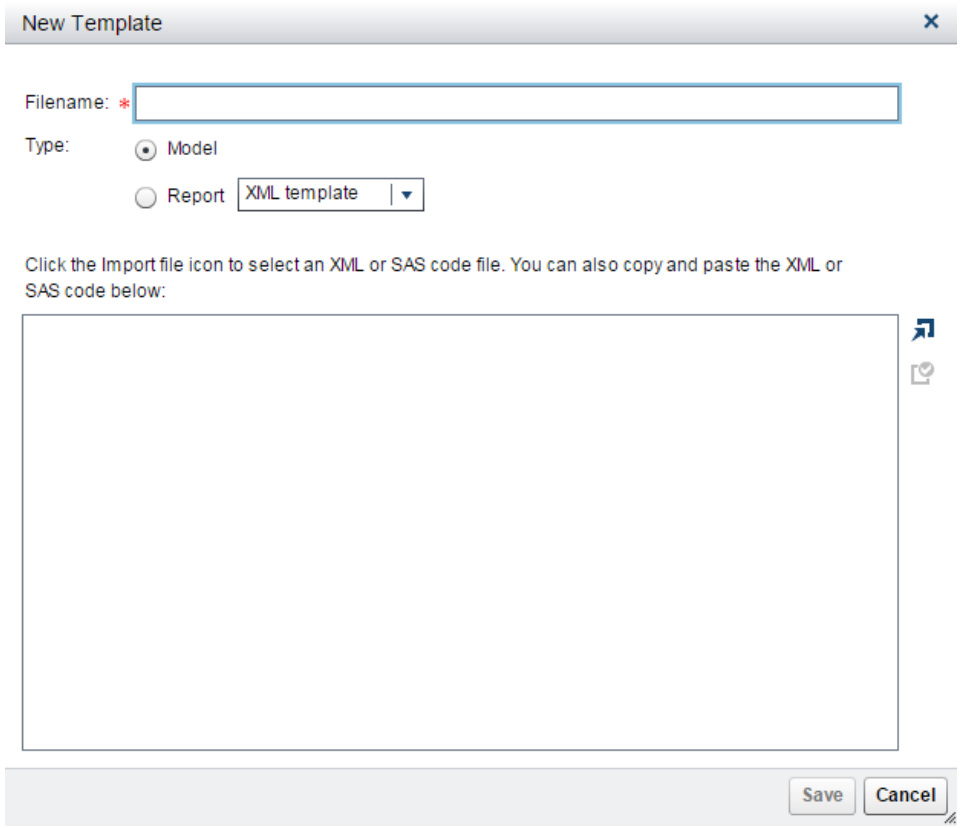
For an example of a macro variable program, see "Example User-Defined Report" on page 235.

For a list of macro variables, see "Macro Variables" in *SAS Model Manager: Macro Reference*.

Upload SAS Programs to the SAS Content Server

After you have the two SAS programs for your user report, follow these steps to upload them to the SAS Content Server:

1. From the Projects category view, click , and select **New Template**.



2. Enter a filename.
3. Select **Report** (SAS code).
4. Click to select a SAS code file. Click **Open**. You can also copy and paste the SAS code in the text box.
5. Click **Save**.
6. Repeat the steps to upload the second file.

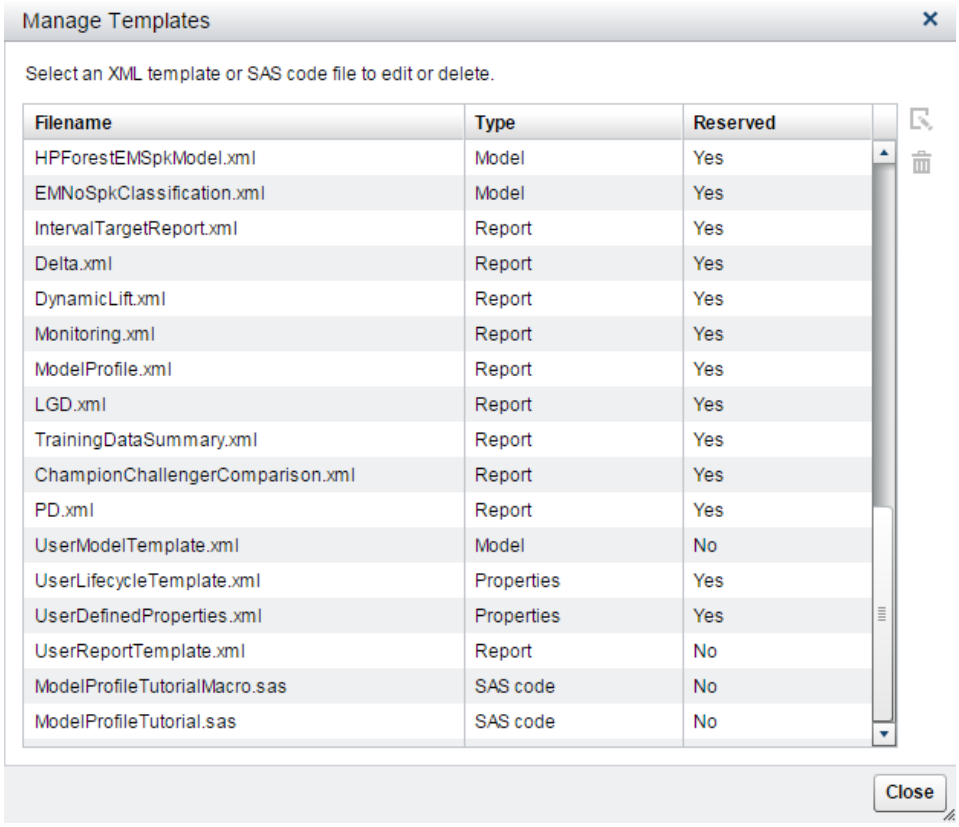
Report Templates

You create a report template XML definition file to describe your user-defined report. After you create the report template, upload the template to the SAS Content Server.

SAS Decision Manager provides a sample report template that you can use as a model for your XML template. You can use any template as a model or you can create an XML file with the required XML elements. A best practice is to open the model XML template and save the template using another name.

To open a sample report template:

1. From the Projects category, click , and select **Manage Templates**.



Filename	Type	Reserved
HPForestEMSpkModel.xml	Model	Yes
EMNoSpkClassification.xml	Model	Yes
IntervalTargetReport.xml	Report	Yes
Delta.xml	Report	Yes
DynamicLift.xml	Report	Yes
Monitoring.xml	Report	Yes
ModelProfile.xml	Report	Yes
LGD.xml	Report	Yes
TrainingDataSummary.xml	Report	Yes
ChampionChallengerComparison.xml	Report	Yes
PD.xml	Report	Yes
UserModelTemplate.xml	Model	No
UserLifecycleTemplate.xml	Properties	Yes
UserDefinedProperties.xml	Properties	Yes
UserReportTemplate.xml	Report	No
ModelProfileTutorialMacro.sas	SAS code	No
ModelProfileTutorial.sas	SAS code	No

2. Select **UserReportTemplate.xml** and click . The UserReportTemplate.xml file has arguments in quotation marks that you modify for your report. Replace the text in quotation marks with values that are appropriate for your report. See the argument descriptions below. Make your changes and click **Save** to upload the report template to the SAS Content Server.

3. Click **Close**.

Here is the report template XML definition:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ReportTemplate
    name="report-name"
    type="UserDefinedReport"
    displayName="display-name"
    description="model-description"
    >
    <Report>
        <Data datasetName="input-data-set-name"/>
        <Models expectedModelType="model-type"
           requiredNumberOfModels="1"
           level="level">
   </Models>
        <SourceCode>
            <PreCode filename="pre-code-filename.sas"/>
            <Code filename="score-code-filename.sas"/>
        </SourceCode>
        <Output format="output-format" filename="output-name"/>
    </Report>
    <Parameters>
```

```
        <Parameter name="parameter-name" value="parameter-value" />
    </Parameters>
</ReportTemplate>
```

<ReportTemplate> element arguments

name="*report-name*"
specifies the name of the report. The characters @ \ / * % # & $ ( ) ! ? < > ^ + ~ ` = { } [ ] | ; : ‘ " cannot be used in the name.

displayName="*display-name*"
specifies the name of the report that is displayed in the **Report** section of the New Report window.

description="*model-description*"
specifies a description of the report that is displayed at the bottom of the New Report window when the report is selected in the window.

<Report> element arguments

<Data datasetName="*input-data-set-name*"/>
specifies the name of a data source data set that is used for input to the report. The data set must be in the form *libref.filename*. You can use the following global macro variables as a value for input-data-set-name as long as the value of the macro variable is in the form of *libref.filename*:

- &_MM_InputLib
- &_MM_OutputLib
- &_MM_PerformanceLib
- &_MM_TestLib
- &_MM_TrainLib

<Models
expectedModelType="*model-type*"
requiredNumberOfModels="*number-of-models*"
level="*level*">
</Models>
specifies information about the model.

expectedModelType="*model-type*"
specifies the model type.

Valid values: ANALYTICAL, CLASSIFICATION, PREDICTION, SEGMENTATION, ANY

requiredNumberOfModels="*number-of-models*"
specifies the number of models that are processed in this report.

level="folder"
specifies where the report is to obtain a list of models. If folder is VERSION, the report creates a list of models in the version. If folder is PROJECT, the report creates a list of models from all versions in the project.

Valid values: VERSION, PROJECT

<SourceCode>
<PreCode filename="*pre-code-filename*.sas"/>
<Code filename="*report-code-filename*.sas"/>
</SourceCode>
specifies the files that are used to execute the report.

<PreCode filename="*pre-code-filename*.sas"/>
specifies the name of the SAS program that contains macro variable definitions.

<Code filename="*report-code-filename*.sas"/>
specifies the name of the SAS program that creates the report.

<Output format="*output-format*" filename="*output-report-name*"/>
specifies the output format arguments:

format="*output-format*"
specifies the format of the report output.

Valid values: HTML, PDF, or RTF

filename="*output-report-name*"
specifies the name of the output report.

<Parameters> Element Argument

<Parameter name="*parameter-name*" value="*parameter-value*" />
This element is not used. It is reserved for future use.

Edit a SAS Program on the SAS Content Server

To edit the program after the file has been uploaded to the SAS Content Server:

1. Click and select **Manage Templates**.
2. Select an XML template, SAS code file, or user-defined properties template to edit. In order for the template to be editable, the **Reserved** column must be marked as **No**. Life cycle templates cannot be edited but can be viewed.
3. Click . Make your changes and click **Save**.
4. Click **Close**.

Delete a SAS Program from the SAS Content Server

Deleting a User Report SAS Content Server is a two-step process. You must delete the SAS program and the report template.

To delete a user report:

1. Click and select **Manage Templates**.
2. Select an XML template, SAS code file, or user-defined properties template to delete. The **Reserved** column must be marked as **No** to delete a file. The user-defined properties template file cannot be deleted.
3. Click . A confirmation window appears.
4. Click **Yes** to delete the file.
5. Click **Close**.

Run a User-Defined Report

To run a user-defined report:

1. Click and select your user-defined report. The New Report window appears.

2. Enter a name and description if you do not want to use the default values.
3. Select an output type. The default is PDF.
4. Select a style for the report. When the SAS default option is selected, the default style and themes are used in generating the report. For example, the SAS default style for the HTML output type is HTMLBLUE.
5. From the list, select the models that you want to include in the report.
6. Click **Run**. The report is generated and appears in the default viewer for the selected output type.

See Also

"View Reports" on page 222

Example User-Defined Report

Overview of the Example User-Defined Report

The example user-defined report categorizes scoring values into score ranges and then graphs the results. The program name is Score Range Report. The following SAS programs and report template file are required to create this report:

- The SAS report program is the file ScoreRange.sas
- The SAS program file that contains macro variables is ScoreRangeMacro.sas
- The report template XML file is ScoreRangeTemplate.xml

SAS Report Program

Here is the SAS code for a user-defined report to categorize score codes:

```
filename mmreport catalog "sashelp.modelmgr.reportexportmacros.source";
%include mmreport;

%MM_ExportReportsBegin(fileName=scoreRange);

options NOmprint NOdate;
%let _MM_PosteriorVar=P_1;


proc format;
  value score
    low - 400 = '400 and Below'
    401 - 450 = '401 - 450'
    451 - 500 = '451 - 500'
    501 - 550 = '501 - 550'
    551 - 600 = '551 - 600'
    601 - 650 = '601 - 650'
    651 - 700 = '651 - 700'
    701 - 750 = '701 - 750'
    751 - 800 = '751 - 800'
    801 - high= '801 and Above';
run;
quit;
```

```
%Macro scoreRange();

  %if &_MM_ScoreCodeType = %str(SAS Program) %then
    %do;
       %let _MM_OutputDS=work.scoreresult;
       %inc &_MM_Score;
    %end;
  %else
    %do;
       data work.scoreresult;
       set &_MM_InputDS;
       %inc &_MM_Score;
       run;
    %end;

  data work.scoreresult2;
    set work.scoreresult;
    keep score;
    if &_MM_PosteriorVar =. then delete;
    score = int (((1-&_MM_PosteriorVar) * 480) + 350 + 0.5);
  run;

  proc freq data=work.scoreresult2;
    table score/out=scoresummary;
    format score score.;
    title 'Credit Score Range';
  quit;

  proc gchart data=work.scoresummary;
    hbar score / sumvar=count discrete;
    title 'Credit Score Range';
  run;
  quit;
%Mend scoreRange;

/*   Reporting section */

ods listing close;

%getModelInfo(0);

/* Define libref and data source for _MM_InputDS */
libname MMLib 'C:\SMM141Tutorials\Data';
%let _MM_InputDS=MMLib.DELINQUENCY_SCORING_INPUT;

%scoreRange();
%closeLibsAndFiles();

%MM_ExportReportsEnd;
```

SAS Program File for Macro Variables

The file ScoreRangeMacro.sas contains only a comment in it because macro variables are not used in the report code:

```
/* ScoreRangeMacro.sas empty file   */
```

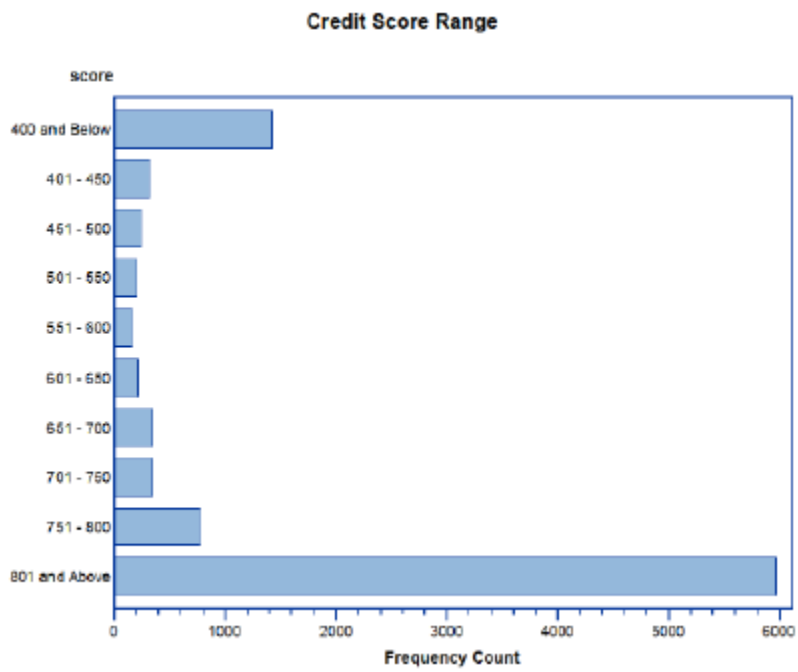
Report Template XML File

Here is the report template XML file for the user-defined Score Range report:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ReportTemplate
    name="Score Range Report"
    type="UserDefinedReport"
    displayName="Score Range Report"
    description="Score Range Report"
    >
    <Report>
        <Data datasetName=""/>
        <Models expectedModelType="ANALYTICAL"
           requiredNumberOfModels="1" level="VERSION">
        </Models>
        <SourceCode>
            <PreCode filename="ScoreRangeMacro.sas"/>
            <Code filename="ScoreRange.sas"/>
        </SourceCode>
        <Output format="PDF" filename="ScoreRange"/>
    </Report>
    <Parameters>
    </Parameters>
</ReportTemplate>
```

Score Range Report Output

The Credit Score Range graph is one of the output pages in the PDF report output.



Chapter 18

Combining Reports

About Aggregated Reports

SAS Decision Manager administrators and advanced users can combine multiple reports from the **Reports** page to create a single, aggregated report. Using reports that reside in the **Reports** page, you select the reports that you want in your aggregated report. The format of the report can be PDF, HTML, or RTF. Aggregated reports are stored on the **Aggregated** tab.

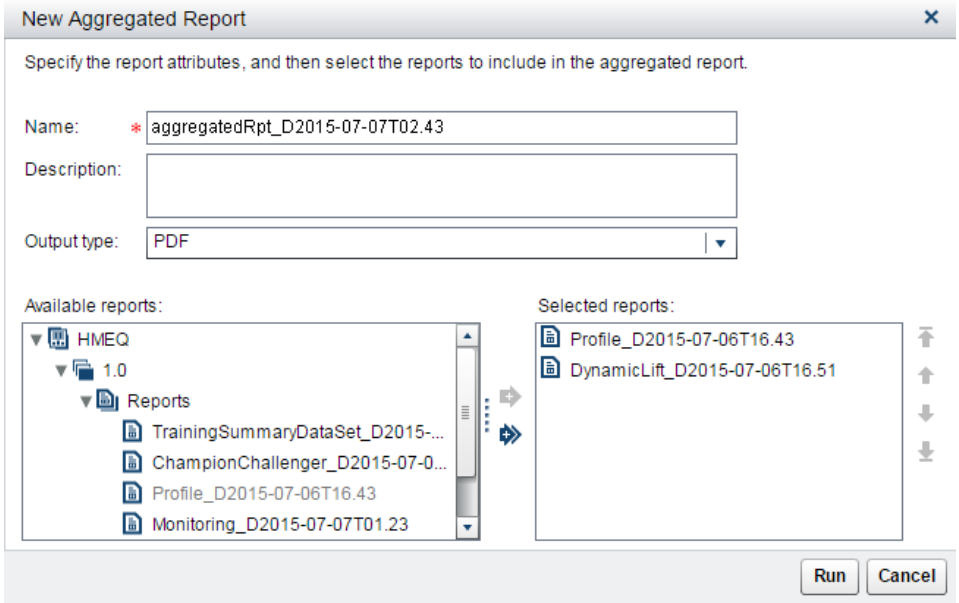
Ad hoc reports, Loss Given Default (LGD) reports, and Probability of Default Model Validation (PD) reports cannot be added to an aggregated report.

Create an Aggregated Report

Note: To create an aggregated report, you must have existing reports on the **Reports** page.

To create an aggregated report:

1. On the **Aggregated** tab on the **Reports** page, click . The New Aggregated Report window appears.



2. (Optional) Enter a name and a description for the report.
3. Select an output type. The default is PDF.
4. In the **Available reports** section, expand the organizational, project, or version folders to show all of the available reports.
5. To add reports from the **Available reports** section, select a report and click to move one report or click to move all reports. The report or reports appear in the **Selected reports** section.
6. To order the reports, select a report and use the up and down arrows.
7. To remove reports from the **Selected reports** section, select a report and click to remove one report or click to remove all reports.
8. When all of the reports are in the **Selected reports** section and in the correct order, click **Run**. The report is generated and appears in the default viewer for the selected output type.
9. The report appears in a list on the **Aggregated** reports tab.

View an Aggregated Report

To view an aggregated report:

1. On the **Aggregated** tab, select a report from the list.
2. View the report in one of several ways:
   - Double-click a report in the list.
   - Select a report from the list and click .
   - Right-click a report from the list and select **Open**.

   Note: You can also view the SAS code and SAS log if the report is not displayed.

Note: If you are configured for single sign-on web authentication and you try to view a report, the report is downloaded to your local drive. To view an HTML report, you must unzip the file.

Delete an Aggregated Report

To delete an aggregated report:

1. On the **Aggregated** tab, select a report from the list.
2. You can delete the report in one of several ways:
   - To delete a file, click . Confirm the deletion.
   - Right-click a report from the list and select **Delete**. Confirm the deletion.

Chapter 19

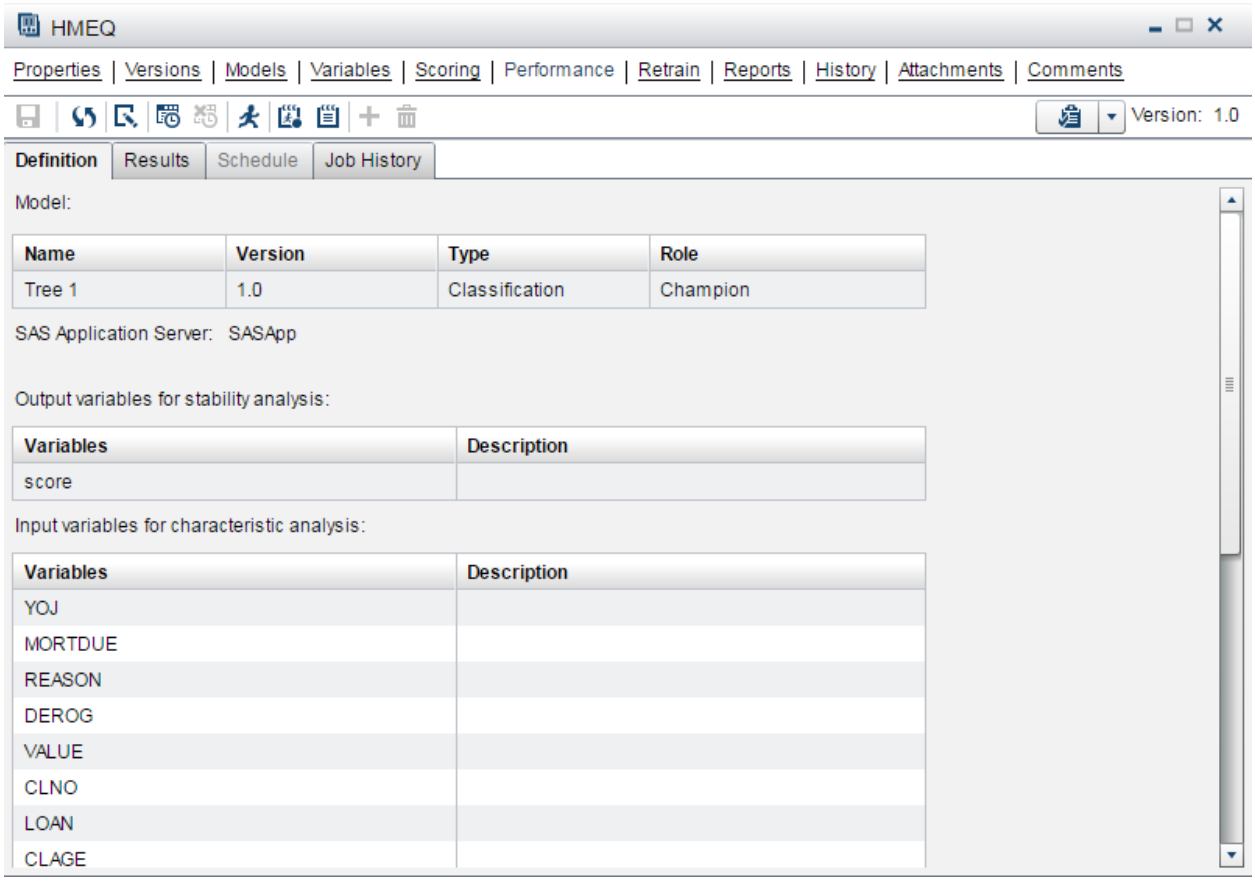
Monitoring Performance of Models

Overview of Performance Monitoring

To ensure that a champion model in a production environment is performing efficiently, you can collect performance data that has been created by the model at intervals that are determined by your organization. A performance data set is used to assess model prediction accuracy. It includes all of the required input variables as well as one or more actual target variables. For example, you might want to create performance data sets monthly or quarterly and then use SAS Decision Manager to create a performance definition that includes each time interval. After you create and execute the performance

definition on the **Performance** page, you can view the performance data through report charts in SAS Decision Manager. These report charts give a graphical representation of the model's performance. SAS Decision Manager also enables you to create performance monitoring reports in PDF, HTML, and RTF output formats from the **Reports** page.

Note: Performance monitoring is designed to work only with a project that is associated with a classification model function and has a binary target, or with a prediction model function and has an interval target. Only models that are associated with the classification and prediction model types and that are set as champion and challenger models can be monitored for performance.



The following types of output for performance monitoring are available:

- Summaries of the types of information in project folders such as the number of models, model age distribution, input variables, and target variables.
- Reports that detect and quantify shifts in the distribution of variable values over time that occur in input data and scored output data.
- Performance monitoring reports that evaluate the predicted and actual target values for a champion model at multiple points in time.

You can create the performance monitoring output, except for summaries, using either of the following methods:

- On the **Performance** page, generate the SAS code that creates the performance output and then execute the generated code.
- Write your own SAS program using the report creation macros that are provided with SAS Decision Manager and submit your program as a batch job. You can run

your SAS program in any SAS session as long as the SAS session can access the SAS Content Server.

After you create and execute a performance definition, you view the report charts by selecting the **Results** tab on the **Performance** page. The report charts are interactive, and you can modify them to help you assess the champion model performance. For example, you can show markers in the charts and show tables for the different types of reports. You can also select different variables for the x-axis and display them in the chart for the Variable Distribution Report.

If you have flagged a challenger model to compare with the champion model, you can use the performance data that you collected for the champion model to create reports for the challenger model. After all of the performance monitoring definitions have been run, you can create a Champion and Challenger Performance report that compares the champion model to the challenger model.

Types of Performance Monitoring

Overview of the Types of Performance Monitoring

After a champion model is in production, you can monitor the performance of the model by analyzing the performance results. You can create the performance output interactively using the Edit Performance Definition wizard on the **Performance** page of a project or you can submit batch programs within SAS.

You can create the following types of performance output:

Summary Results
: The **Summary** results summarize the number of models, the number of versions, the number of scoring tests, and the number of reports. The summary information enables you to compare the contents of folders, projects, and versions. You view the Summary results by selecting **Actions** ⇨ **View Summary**.

Data Composition Reports
: The Variable Distribution report shows you the distributions for a variable in one or more time periods, which enables you to see the differences and changes over time. The Characteristic and Stability reports detect and quantify shifts in the distribution of variable values that occur in input data and scored output data over time. By analyzing these shifts, you can gain insights on scoring input and output variables.

Model Monitoring Reports
: The model monitoring reports are a collection of performance assessment reports that evaluate the predicted and actual target values. The model monitoring reports create several charts:

- Lift
- Gini - ROC (Receiver Operating Characteristic)
- Gini - Trend
- KS
- MSE (Mean Squared Error) for prediction models

When you create Data Composition reports and Model Monitoring reports, you can set performance index warnings and alerts. When certain thresholds are met, SAS Decision Manager can send a warning and alert notification to email addresses that you configure either in the Edit Performance Definition wizard or in a SAS program.

You view the Data Composition reports and the Model Monitoring reports on the **Results** tab on the **Performance** page.

Summary Results

The Summary results summarizes the contents of different folders and projects.

The contents of the Summary results is dynamic and is updated according to the selected project. The scope of the information that is reported is defined by the collection of folders and objects that exist beneath the folder that is selected.

To view the Summary results, select **Actions** ⇨ **View Summary**.

Use the following sections to evaluate and compare the contents of the project:

General
Use the **General** section to browse the number of models, the number of versions, and the number of scoring tests.

Summary of Reports
Use the **Summary of Reports** section to browse the number of reports that are available on the **Reports** page for the selected object.

Model Target Variable Report
Use the **Model Target Variable Report** to see the frequency with which target variables are used in the models that exist for the selected object. Each unique model target variable is reported, listing the number of models that use that variable as a target variable.

Model Input Variable Report
Use the **Model Input Variable Report** to see the frequency with which input variables are used in the models for a folder or project. Each unique model input variable is reported, listing the number of models that use that variable as an input variable.

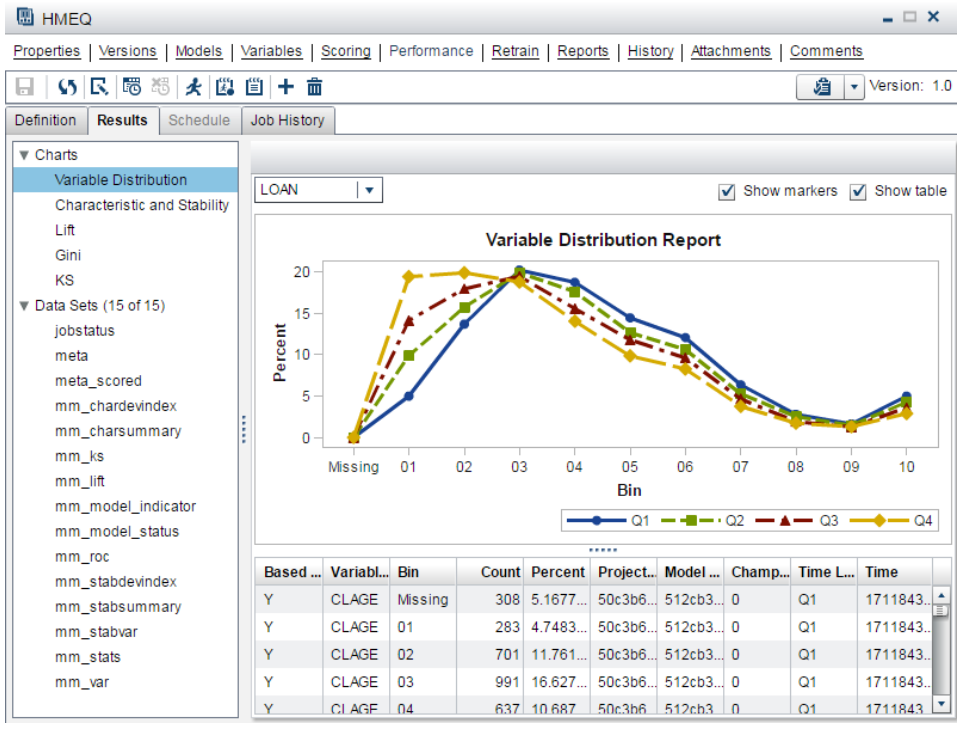
Data Composition Reports

Variable Distribution Report

Select the **Results** tab on the **Performance** page to view the **Variable Distribution** report. The variable distribution chart is a graphical representation of distributions over a period of time for the selected variable. Each line plot represents the data for a specific period of time. The Y-axis is the percentage of observations in a bin that is proportional to the total count.

To change the variable that appears in the chart, select a variable from the drop-down list.

Here is an example of a Variable Distribution report. By placing the cursor over a point in the chart, you can view the data for that point.



Characteristic and Stability Reports

Together, the Characteristic and Stability reports detect and quantify shifts that can occur in the distribution of model performance data, scoring input data, and the scored output data that a model produces.

Note: For each time period that you execute a performance definition, SAS Decision Manager creates a new point on the charts. Line segments between points in time do not appear on the charts unless you specify at least three data sources and collection dates as part of the performance definition.

Characteristic Report

The Characteristic report detects and quantifies the shifts in the distribution of variable values in the input data over time. These shifts can point to significant changes in customer behavior that are due to new technology, competition, marketing promotions, new laws, or other influences.

To find shifts, the Characteristic report compares the distributions of the variables in these two data sets:

- the training data set that was used to develop the model
- a current data set

If large enough shifts occur in the distribution of variable values over time, the original model might not be the best predictive or classification tool to use with the current data.

The Characteristic report uses a deviation index to quantify the shifts in a variable's values distribution that can occur between the training data set and the current data set. The deviation index is computed for each predictor variable in the data set, using this equation:

$$\textit{Deviation\_Index} = \Sigma\,(\%\text{Actual} - \%\text{Expected})\ {}^{*}1n\ (\%\text{Actual} / \%\text{Expected})$$

Numeric predictor variable values are placed into bins for frequency analysis. Outlier values are removed to facilitate better placement of values and to avoid scenarios that can aggregate most observations into a single bin.

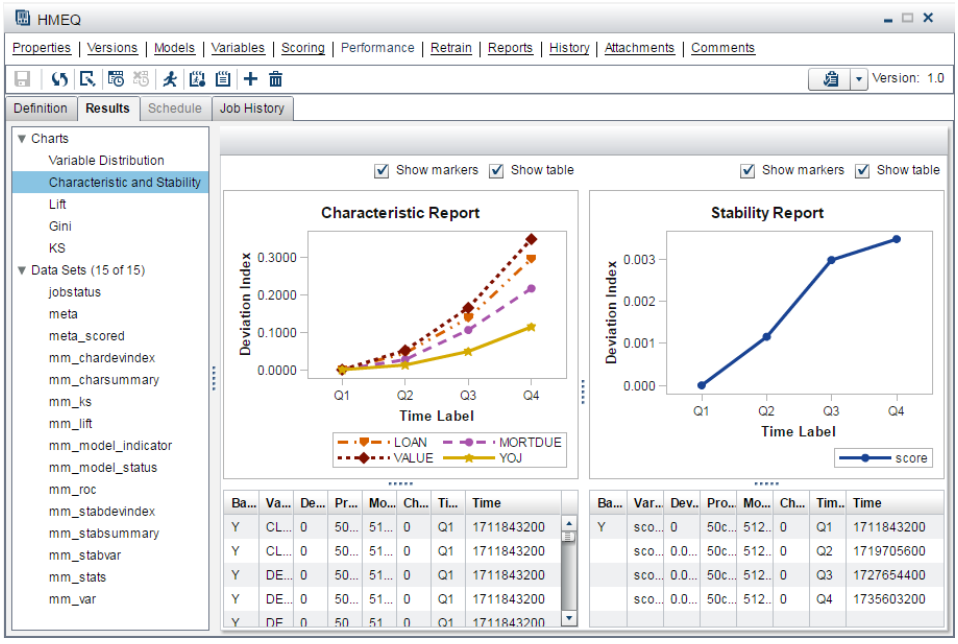
If the training data set and the current data set have identical distributions for a variable, the variable's deviation index is equal to 0. A variable with a deviation index value that is P1>2 is classified as having a mild deviation. The Characteristic report uses the performance measure P1 to count the number of variables that receive a deviation index value that is greater than 0.1.

A variable that has a deviation index value that is P1>5 or P25>0 is classified as having a significant deviation. A performance measure P25 is used to count the number of variables that have significant deviations, or the number of input variables that receive a deviation index score value that is greater than or equal to 0.25.

Stability Report

The Stability report evaluates changes in the distribution of scored output variable values as models score data over time, and detects and quantifies shifts in the distribution of output variable values in the data that is produced by the models. If an output variable from the training data set and the output variable from the current data set have identical distributions, then that output variable's deviation index is equal to 0. An output variable with a deviation index value that is greater than 0.10 and less than 0.25 is classified as having a mild deviation. A variable that has a deviation index value that is greater than 0.30 is classified as having a significant deviation. Too much deviation in predictive variable output can indicate that model tuning, retraining, or replacement might be necessary.

Here is an example of Characteristic and Stability reports. By placing the cursor over a point in the chart, you can view the data for that point.



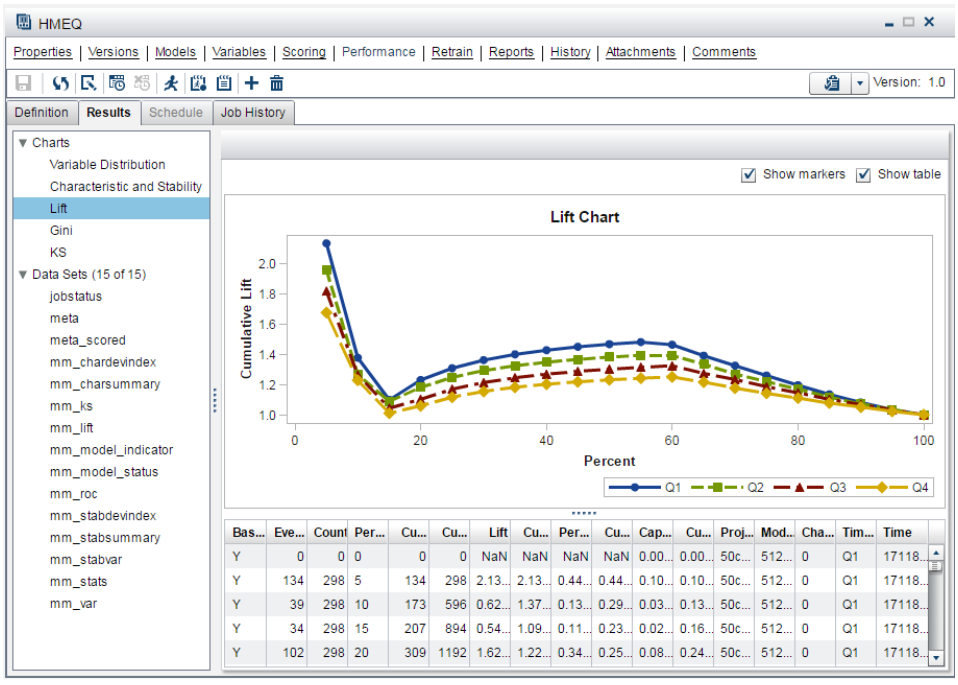
Model Monitoring Reports

Lift Report

The Lift report provides a visual summary of the usefulness of the information that is provided by a model for predicting a binary outcome variable. Specifically, the report summarizes the utility that you can expect by using the champion model as compared to using baseline information only. Baseline information is the prediction accuracy performance of the initial performance monitoring definition or batch program using operational data.

A monitoring Lift report can show a model's cumulative lift at a given point in time or the sequential lift performance of a model's lift over time. To detect model performance degradation, you can set the Lift report performance indexes Lift5Decay, Lift10Decay, Lift15Decay, and Lift20Decay. The data that underlies the Lift report is contained in the report file mm_lift.sas7bdat. This file is available on the **Attachments** page.

Here is an example of a monitoring Lift report. By placing the cursor over a point in the report, you can view the data for that point.



Bas...	Eve...	Count	Per...	Cu...	Cu...	Lift	Cu...	Per...	Cu...	Cap...	Cu...	Proj...	Mod...	Cha...	Tim...	Time
Y	0	0	0	0	0	NaN	NaN	NaN	NaN	0.00...	0.00...	50c...	512...	0	Q1	17118...
Y	134	298	5	134	298	2.13...	2.13...	0.44...	0.44...	0.10...	0.10...	50c...	512...	0	Q1	17118...
Y	39	298	10	173	596	0.62...	1.37...	0.13...	0.29...	0.03...	0.13...	50c...	512...	0	Q1	17118...
Y	34	298	15	207	894	0.54...	1.09...	0.11...	0.23...	0.02...	0.16...	50c...	512...	0	Q1	17118...
Y	102	298	20	309	1192	1.62...	1.22...	0.34...	0.25...	0.08...	0.24...	50c...	512...	0	Q1	17118...

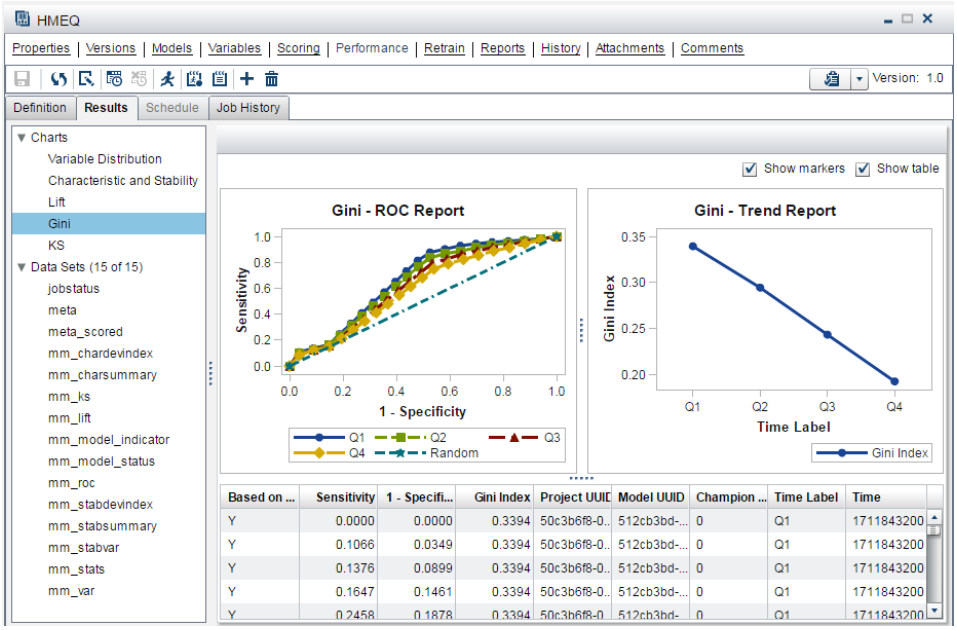
Gini (ROC and Trend) Report

The **Gini (ROC and Trend)** reports show you the predictive accuracy of a model that has a binary target. The plot displays sensitivity information about the y-axis and 1-Specificity information about the x-axis. Sensitivity is the proportion of true positive events. Specificity is the proportion of true negative events. The Gini index is calculated for each ROC curve. The Gini coefficient, which represents the area under the ROC curve, is a benchmark statistic that can be used to summarize the predictive accuracy of a model.

Use the monitoring **Gini (ROC and Trend)** report to detect degradations in the predictive power of a model.

The data that underlies the monitoring **Gini (ROC and Trend)** report is contained in the report component file mm_roc.sas7bdat.

The following chart is an example of a monitoring **Gini (ROC and Trend)** report. By placing the cursor over a point in the chart, you can view the data for that point.



KS Report

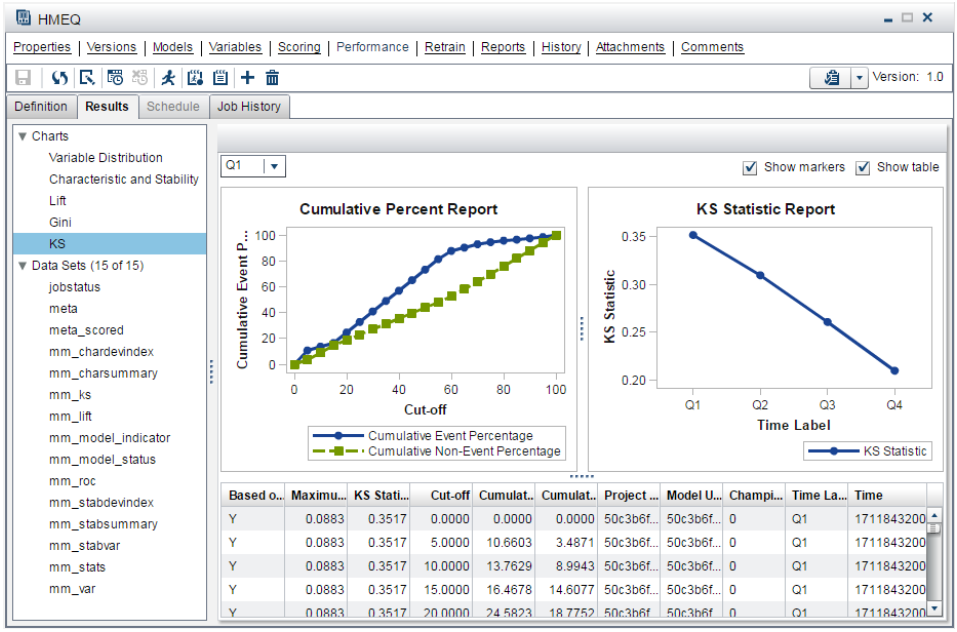
The KS report contains the Kolmogorov-Smirnov (KS) test plots for models with a binary target. The KS statistic measures the maximum vertical separation, or deviation between the cumulative distributions of events and non-events. This trend report uses a summary data set that plots the KS statistic and the KS probability cutoff values over time.

Use the KS report to detect degradations in the predictive power of a model. To scroll through a successive series of KS performance depictions, select a time interval from the **Time Interval** list box. If model performance is declining, it is indicated by the decreasing distances between the KS plot lines.

To detect model performance degradation, you can set the ksDecay performance index in the KS report.

The data that underlies the KS chart is contained in the report component file mm_ks.sas7bdat.

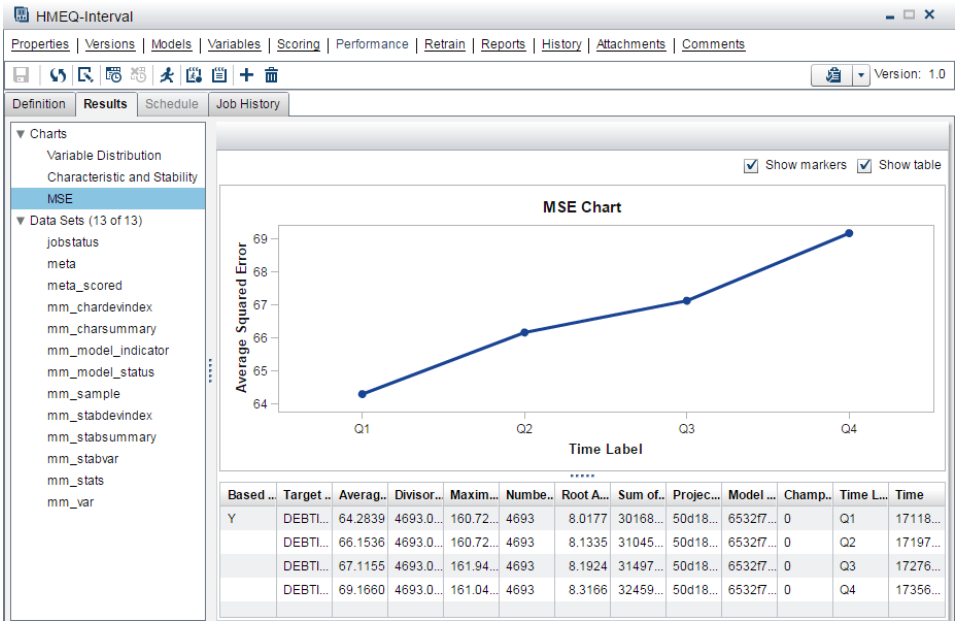
The following report is an example of a KS report. By placing the cursor over a point in the chart, you can view the data for that point.



Mean Squared Error Report

The Mean Squared Error (MSE) report checks the accuracy of a prediction model with an interval target by comparing the estimation derived from the test data and the actual outcomes that are associated with the test data for different time periods.

The following report is an example of an MSE report.



Performance Index Warnings and Alerts

The production model performance reports use performance measurement thresholds to benchmark and gauge the performance of a predictive model. When one of the

performance measurements exceeds one or more specified indexes or thresholds, warning and alert events occur. When warning or alert events occur, warning and alert notifications are automatically sent by email to recipients whose email address is configured either in the Edit Performance Definition wizard or in the batch program that runs the reports.

Use the following assignment statements to set warning and alert conditions:

```
alertCondition='alert-condition';
warningCondition='warning-condition';
```

Note: The condition must be enclosed in quotation marks if you use SAS code to create the report. An error occurs if you enclose the condition in quotation marks in the Edit Performance Definition wizard.

The following indexes and thresholds can be configured in either the Edit Performance Definition wizard or in a batch program that creates the report specifications:

Characteristic report
: You can configure the thresholds for the performance indexes P1 and P25. The P1 and P25 indexes represent the count of input variables with deviation index scores exceeding 0.1 and 0.25, respectively. Here is an example of alert and warning thresholds:

```
alertCondition='p1>5 or p25>0';
warningCondition='p1>2';
```

Stability report
: You can configure output deviation index scores for a model's output variable. The output deviation index scores represent the deviation levels in the distribution of the model's scored output variables. Here is an example of alert and warning thresholds:

```
alertCondition='outputDeviation>0.03';
warningCondition='outputDeviation>0.01';
```

Model Assessment reports
: For the Lift, Gini (ROC and Trend), and KS reports, you can configure threshold values for the following decay statistics.

lift5Decay	is the lift performance decay based on the top 5% of the target population of interest from time A to time B.
lift10Decay	is the lift performance decay based on the top 10% of the target population of interest from time A to time B.
lift15Decay	is the lift performance decay based on the top 15% of the target population of interest from time A to time B.
lift20Decay	is the lift performance decay based on the top 20% of the target population of interest from time A to time B.
giniDecay	is the performance decay of the Gini index from time A to time B.
ksDecay	is the performance decay of the KS statistic from time A to time B.

For the prediction model MSE report, you can configure the mseDecay statistic threshold values. The mseDecay statistic is the performance decay of the MSE statistic from time A to time B.

Here is an example of alert and warning thresholds:

```
alertCondition='(lift5Decay>0.15 and lift10Decay>0.12)
```

```
                                or giniDecay>0.1 or ksDecay>0.1';
warningCondition='lift5Decay>0.05';
```

Monitoring Champion Models

Your project plan might include a schedule to monitor the champion model performance, or your plan might require that you monitor the performance at any time. For each time period that you monitor the champion model, you take a snapshot of the data for that time period and use that data as the performance data source for creating the monitoring reports.

You can create monitoring reports by creating and executing a performance definition, or you can submit batch programs to create the reports. Both methods require the same information. Both methods can process one or more performance data sources. When you create a performance definition, you can specify one or more data sources to process. When you use a batch program, you use a separate DATA step to process each data source.

If you run batch programs, you can find example programs in the sashelp.modelmgr.source catalog. These reports' filenames are **reportexample***x*, where *x* is a number from 1 to 4.

The following table lists the definitions that are required to create performance reports:

Definition	Reports Created by Using the Edit Performance Definition Wizard	Reports Created Using SAS Programs That Run in Batch
Create a folder structure for report files	The folder structure is inherent in the Project. No action is necessary.	Create a folder structure on a local drive of the SAS Model Manager server.
Obtain performance data	The performance data is one or more SAS data sets that are a snapshot of model output. They can be registered in SAS Management Console or they can be accessed by using a libref that has been defined by using the Edit Start-up Code window.	The performance data is used to assess model prediction accuracy. It includes all of the required scoring input variables as well as one or more actual target variables. You can store performance data sets anywhere as long as they can be accessed by the SAS session that runs the batch program. The data sets do not need to be registered with SAS Management Console.
Ensure access to the champion or challenger model	This process is performed by the Edit Performance Definition wizard. No action is necessary.	Run the %MM_GetModels() macro to extract the champion model in a channel to a local drive of the SAS Model Manager server.

Definition	Reports Created by Using the Edit Performance Definition Wizard	Reports Created Using SAS Programs That Run in Batch
Map model and project output variables.	Map the model and project output variables using the Project Tree.	Map the model and project output variables using the Project Tree.
Define report specifications	The report specification are derived from project data and input that you specify in the Edit Performance Definition wizard. The wizard generates the SAS code to create the performance reports.	Write the following DATA steps: • mm_jobs.project • mm_jobs.emailaddr • mm_jobs.reportdef • mm_jobs.jobtime
Specify the report execution operational environment	The operational environment is known to SAS Decision Manager. No action is necessary.	Define the required macro variables that are used by the %MM_RunReports() macro.
Run the reports	Execute the code from the **Performance** page that was generated by the Edit Performance Definition wizard or schedule the performance definition from the **Performance** page. The data sets that underlie the monitoring reports are stored in the **Results** tab.	Create a DATA step that points to the performance data sets and execute the %MM_RunReports() macro. The data sets that underlie the monitoring reports are stored in the **Results** tab when the reports are created in production mode. In Test mode, the monitoring reports data sets reside in the location specified in the mm_jobs.project data set.
View the reports	Select the **Performance** page for the champion model to view the reports.	Select the **Performance** page for the champion model to view the reports.

Creating Reports Using a Performance Definition

Overview of Creating Reports Using a Performance Definition

You define and execute a performance definition for a project. The model that you monitor is either the project champion model or a challenger model that is flagged in any version for the project. The process of creating performance reports is a two-step process. First, you run the Edit Performance Definition wizard to generate the code that creates the performance data results. Then, you execute the generated code. You can execute the code immediately, or you can schedule a date and time at which the definition is to run. Information about performance definitions is recorded and can be viewed on the **Results** tab of the **Performance** page.

To create performance reports:

- Ensure that one or more performance data sources are registered using SAS Management Console or that a libref has been defined for the location where the performance data sets are stored.
- Ensure that all prerequisites have been completed.
- Run the Edit Performance Definition wizard to generate the SAS code that creates the performance reports.
- Execute the generated code or schedule when the generated code is to be executed.
- To view the reports, select the **Results** tab on the **Performance** page.

Determine How to Use the Performance Data Sets

Before you run the Edit Performance Definition wizard, the performance data sets must be registered in the SAS Metadata Repository. You can register the data sets in the Data category view or you can add tables to an existing library that have already been registered using SAS Management Console. For each project, you can set up your environment to use the performance data source that is most appropriate for your business process. Here are two methods of collecting performance data:

- Method 1: You periodically take a snapshot of an operational data set to create a performance data set. Each time you take a snapshot, you give the performance data set a new name. Each performance data set must be registered in the SAS Metadata Repository and it must be available in the **Data** category view. You can create and execute a performance monitoring definition each time you take a snapshot, or you can create a performance monitoring definition to execute multiple performance data sets in the same definition. The best practice is to use the dynamic data sources in the performance definition.
- Method 2: You take a snapshot of the operational data set to create a performance data set over time, and you reuse the same name for each performance data set every time you take a snapshot. You register the performance data set in the SAS Metadata Repository only once. The performance data set must be available in the **Data** category view. Each time you take a snapshot, you replace the performance data set at the location where the performance data set is registered.

  When you run the Edit Performance Definition wizard, the name of the performance data source does not change. The **Default performance table** project property is not populated in the Edit Performance Definition wizard. You modify only the **Collection Date** and **Report Label** columns in the table.

The following table summarizes the definitions that are performed if performance reports are run after six months or for reports that are run every month. Use this definition and example table to help you determine how you want to name your performance data sets and your performance data sources.

Definition	Method 1: The Performance Data Set Name Changes	Method 2: The Performance Data Set Name Remains Static
Create a performance data set from model output data	Each month, take a snapshot of the operational data and create a performance data set with a different name: • Jul13 • Aug13 • Sep13 • Oct13 • Nov13 Dec13	Every month, take a snapshot of the operational data and name the performance data set using the same name: 2013perf
If you are registering the performance data sets in the SAS Metadata Repository, register the performance data sets using SAS Management Console	Register the data sets monthly or register them all at once before you run the reports.	Register the data sets the first month only.
If the performance data set is accessed by using a libref, store the data set in the SAS library.	Save the performance data set in the SAS library that is defined by a libref in SAS Decision Manager.	Save the performance data set in the SAS library that is defined by a libref in SAS Decision Manager.
Modifications to make in the Edit Performance Definition wizard	In Step 3, select one or more performance data sources. For each data source, select a data collection date and enter a date label.	In Step 3, select a data collection date and enter a date label. The **Performance data source** field contains the static name of the performance data source name because it was specified for the previous execution of the definition for this project.
Create the reports	Run the Edit Performance Definition wizard and execute the reports from the **Performance** page or schedule when the definition is to execute. Because each performance data source has a different name, you can run the performance definition as desired; the definition does not need to be run monthly.	Monthly, run the Edit Performance Definition wizard and execute the reports from the **Performance** page or schedule when the definition is to execute. To ensure that you do not write over important performance data, run the performance definition before a new snapshot of the operational data is taken.

Prerequisites for Editing a Performance Definition

Overview of Prerequisites

Before you edit a Performance Definition, the environment must be set appropriately as follows:

- Ensure that the champion model is set or the challenger model is flagged.
- Ensure that the champion or challenger model is within a project that is associated with a classification model function and has a binary target, or that is associated with a prediction model function and has an interval target.
- Ensure that the champion or challenger model contains a score.sas file. If the performance data set contains the predicted values, the score.sas file can be empty. For more information, "Monitoring Performance of a Model without Score Code" on page 265.
- Ensure that the performance data sets for the time period that you want to monitor are registered in SAS Management Console or that a libref has been defined for the SAS library where the performance data sets are saved.
- Ensure that the appropriate project and model properties are set.

After the environment is set, you can run the Edit Performance Definition wizard.

Ensure That Champion and Challenger Models Are Set

The Edit Performance Definition wizard generates report code for the champion model in the champion version.

You can determine the champion version and the champion model by looking for the ✓ icon next to the champion version name and the champion model name on the **Versions** page. The status is also indicated in the **Role** column on the **Models** page.

If the champion model is not set, select a model and click ✓ or right-click the champion model name and select **Set as Champion**. The ✓ icon appears next to the champion model name and the version for the champion model.

You can determine the challenger model by looking at the **Role** column on the **Models** page. View the number of challengers on the **Details** view of the **Versions** page.

If the challenger model is not set, click ⚑ or right-click the challenger model name and select **Flag as Challenger**.

Ensure That the Champion Model Function and Class Target Level Are Valid

Performance monitoring is valid only for a project that is associated with a classification model function and has a binary target, or for a prediction model function that has an interval target. You should define only performance definitions for classification and prediction models. The champion model must either have a function type of classification and must contain a binary target, or have a function type of prediction and must contain an interval target.

From the Projects category view, select the champion model name and verify that the **Model function** property in the specific properties section is set to `Classification` or `Prediction`. For models that are created using SAS Enterprise Miner, verify that **Class target level** is set to `BINARY` for a classification model or to `INTERVAL` for a prediction model.

Ensure That the Performance Data Source Is Available

The Edit Performance Definition wizard requires that the performance data be registered in the SAS Metadata Repository. You can register the data sets in the Data category view or you can add tables to an existing library that have already been registered using SAS Management Console.

If your performance table is not available for selection, contact your administrator to add the table to the Data Library Manager using SAS Management Console. For more information, see the *SAS Decision Manager: Administrator's Guide*.

Ensure That Project and Model Properties Are Set

Several properties must be defined in order to generate the model performance reports. Verify that the appropriate project and model properties are set. Here is a list of properties.

Classification Project Properties

- Training target variable
- Target event value
- Class target level
- Output event probability variable

Prediction Project Properties

- Training target variable
- Class target level
- Output prediction variable

Model Properties

- Score code type

Map Model and Project Output Variables

In order to create the model performance reports, you must map the model output variable to the project output variable if the corresponding project variable and the model variable have different names.

To map the model variables to the project variables:

1. Select and open a model.
2. Select **Model Properties** ⇨ **Output Mapping**.
3. Click the box in the **Value** column beside the variable in the **Property** column to display a list of project variables.
4. Select a model output variable.
5. Repeat steps 3 and 4 for each model variable that requires mapping.

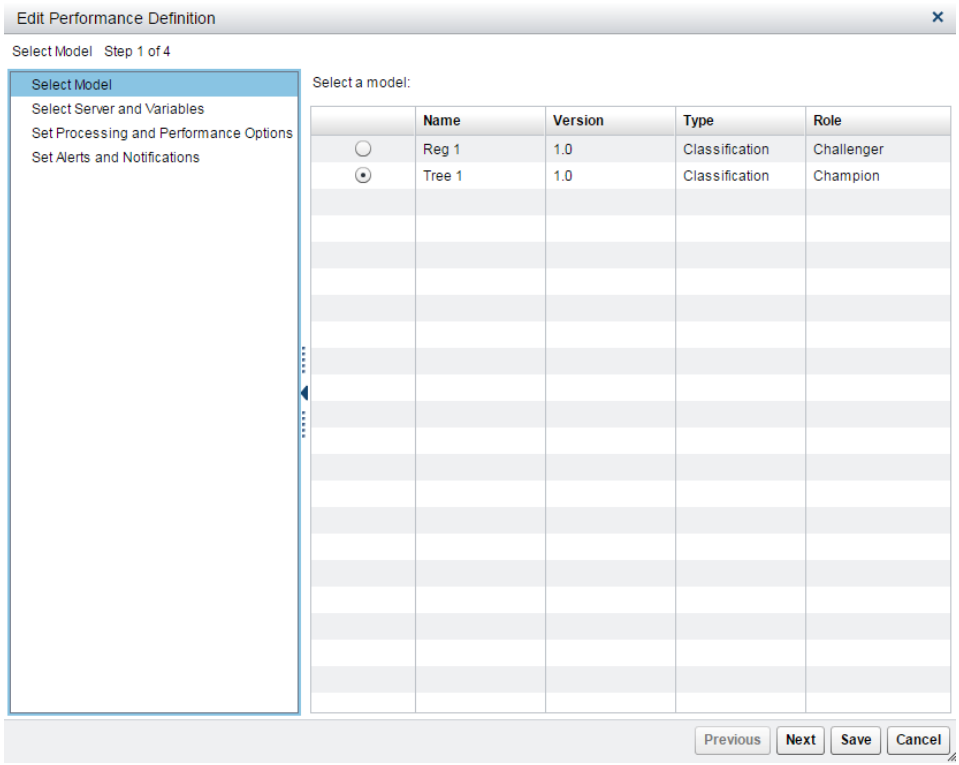
6. Click .

Edit and Execute a Performance Definition

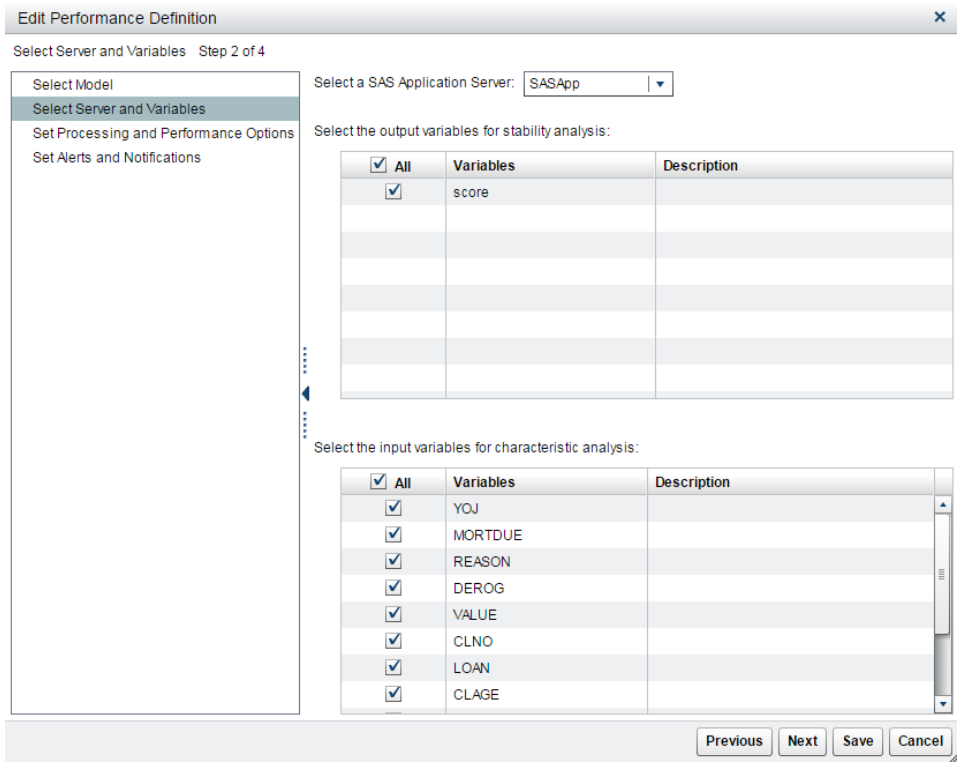
To create the monitoring reports, you specify a performance definition to generate SAS code. You then execute the generated code or create a schedule to execute the generated code on a specific day and time. Execution of the generated code creates the SAS data sets that are used to display reports: either the monitoring reports from the version **Performance** page, or the Monitoring report or Champion and Challenger Performance report that you create from the New Report window.

To edit the performance definition:

1. Click and select a champion or challenger model. Click **Next**.



2. Select a SAS Application Server.
3. Select one or more output variables for stability analysis. To select all output variables, click **All**.
4. Select one or more input variables for characteristic analysis. To select all input variables, click **All**. Click **Next**.



5. Choose the data processing method:

   - To run a standard environment, select **Standard configuration**.

     To run the score code in the performance monitor job, select the **Run model score code** check box.

   - To run the performance monitoring definition in a High-Performance Analytics environment, select **High-performance configuration**.

     Note: The score code is not run when **High-performance configuration** is selected.

6. Decide to use either the static or dynamic data sources, and then specify the data source information.

   Note: Ensure that the data source information is complete before saving the definition. If you start adding information for static data sources and then decide to use dynamic data sources instead, be sure to delete the information added for static data sources before adding the dynamic data source information, and vice versa.

   To use static data sources:

   a. Click +.

      Note: If you are adding multiple tables in the first performance definition, the data table whose collection date is the earliest is set as the baseline performance data table.

   b. Click the empty cell in the **Data Source** column.

   c. Click **Browse** and select a performance data source. Click **OK**.

   d. Click the empty cell in the **Collection Date** column and click . Select a date. The date can be any date in the time period when the performance data was collected.

e. To add a label for the date, enter the label name in the **Report Label** column. The report label represents the time point of the performance data source. Because the report label appears in the performance charts, use a label that has not been used for another time period, is short, and is understandable (for example, Q1).

Note: Duplicate report labels result in previous performance results being overwritten.

f. (Optional) Select a data source and click to verify that the selected input variables and target variable are included in the performance data source, and that the data source is not empty.

g. (Optional) Repeat the above steps to add multiple performance data sources to the performance definition.

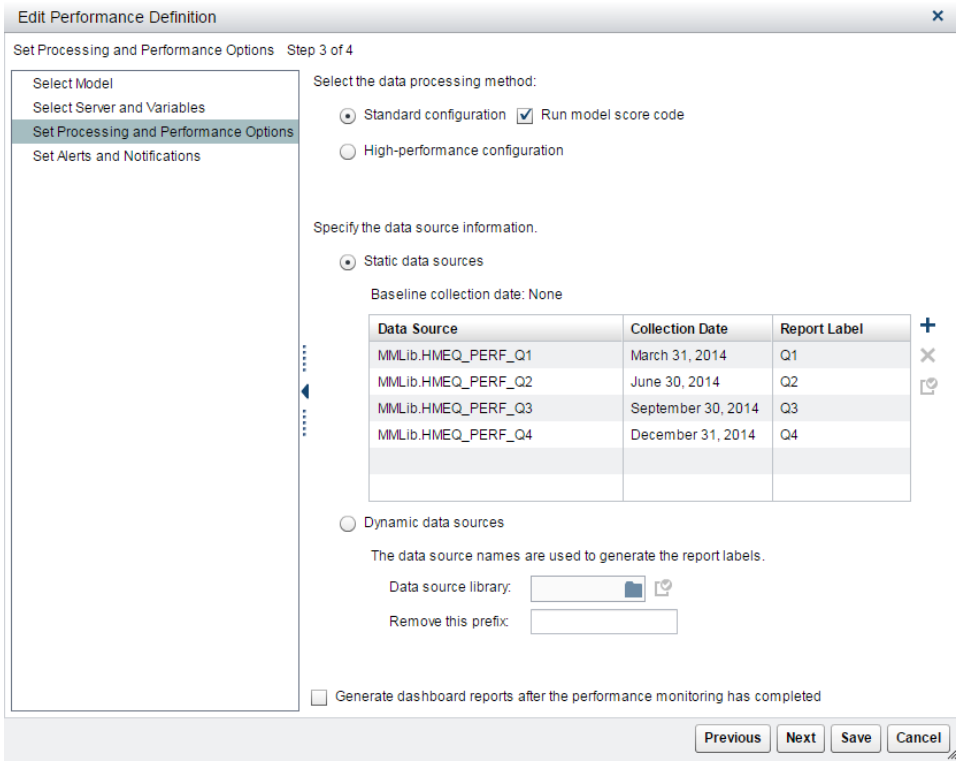
h. (Optional) To delete a data source from the performance definition, select the data source and click .

To use dynamic data sources:

a. Click to select a data source library.

b. (Optional) Specify the prefix to remove from the data source names in the selected library. The data source name is used for the report label. You can remove the prefix so that it does not show as part of a report label on the charts.

7. (Optional) Select **Generate dashboard reports after the performance monitoring has completed**. The dashboard definition must already exist for this option to work.



8. Click **Next**.

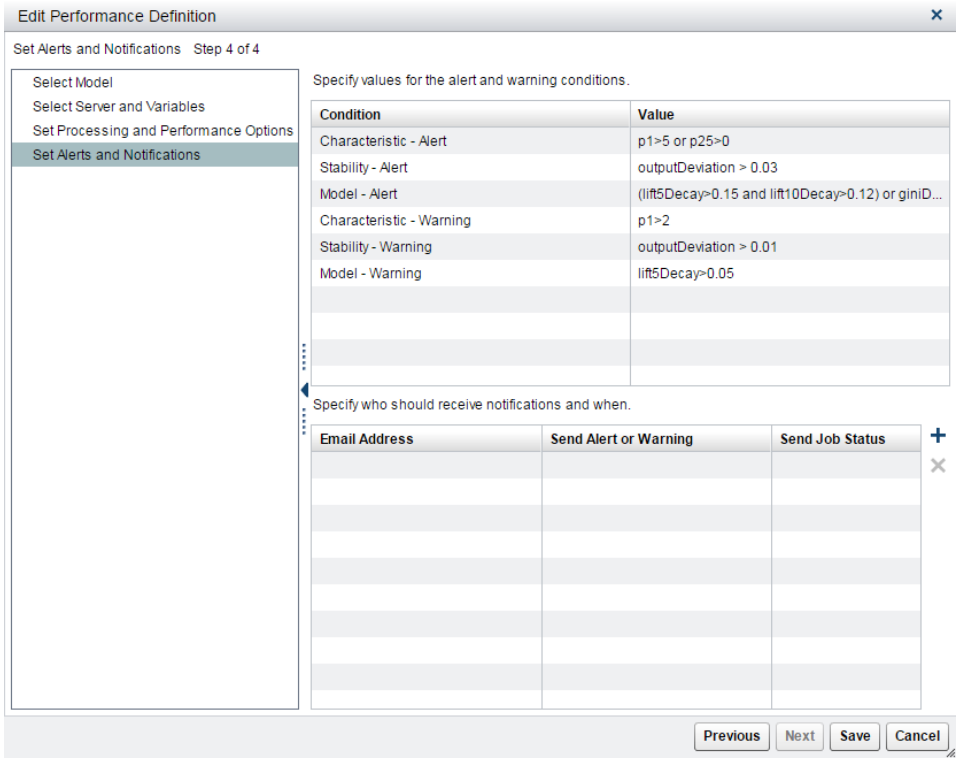
9. (Optional) Either specify values for the alert and warning conditions or accept the defaults.

10. (Optional) To send the results by email, click . A new row is added to the table.

a. Enter an email address.

b. Select either **Yes** or **No** if you want an alert or warning to be sent by email when alert or warning thresholds have been exceeded.

c. Select either **Yes** or **No** if you want a completion notice with the job status to be sent by email every time the report runs.



11. Click **Save**.

To execute a performance definition:

1. Select the **Performance** page for the project.
2. Click .
3. After the performance monitoring has been completed, a confirmation message appears. Click **Close**.
4. Click the **Results** tab to view the performance results.

   Note: You can check the status of a job by clicking and then selecting the **Results** tab or the **Job History** tab.

Note: You can overwrite or delete previously created performance data sets.

Schedule Performance Definitions

After you create a performance definition, you can create a schedule to execute the definition to run on a specific day and at a specific time. You can schedule the definition to run hourly, daily, weekly, monthly, or yearly.

Before you can schedule a performance definition, your user ID and password must be made available to the SAS Metadata Repository. You must also sign in to SAS Decision Manager using your full user credentials that were specified for your user account in SAS Management Console. For user accounts where a Microsoft Windows user ID is specified, you must enter your user ID in the format of *domain\userID*. Contact your system administrator to add or update your password, and to determine the correct user credentials for your user account.

CAUTION:

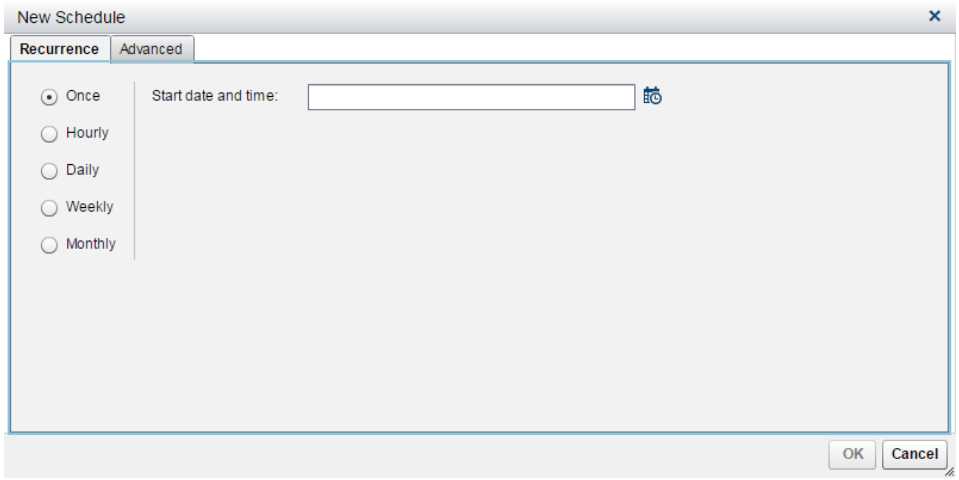
Check your configuration settings before scheduling jobs. Users who are configured for single sign-on authentication by Kerberos cannot schedule jobs.

You cannot edit a schedule for a performance definition. To modify a schedule, delete the schedule and create a new schedule.

After performance monitoring jobs execute, you can view the job history using the **Job History** tab on the **Performance** page.

To schedule a performance monitoring definition:

1. Click .



2. On the **Recurrence** tab, select the recurrence pattern.
3. Specify the criteria for when and how often the job should be run.
4. (Optional) Select the **Advanced** tab.
   a. Select the server that schedules the job from the **Scheduling server** list box.

      Note: The SAS Distributed In-Process Services Scheduling Server is not supported.

   b. Select the batch server that runs the job from the **Batch server** list box.
   c. Click **Browse** to select a location for the performance monitoring output. Click **OK**.
5. Click **OK**.
6. After the job has been scheduled, a confirmation message appears. Click **Close**.
7. Click the **Results** tab to view the performance results.

Note: Performance schedules cannot be edited. To change the schedule, delete the schedule and create a new schedule.

Here is a list of the **Schedule** properties for **Performance**:

Property Name	Description
Job Name	Specifies the name of the performance monitoring definition. This name cannot be changed.
Location	Specifies the location of the performance monitoring definition in the SAS Metadata Repository.
Scheduling Server	Specifies the name of the server that schedules the job for the performance monitoring definition.
Batch Server	Specifies the name of the server that executes the job for the performance monitoring definition.
Recurrence	Specifies how often the scheduled job for the performance monitoring definition is to be executed.
SAS Application Server	Specifies the name of the SAS Application Server where the performance monitoring definition is to be executed.

To delete a schedule, select the schedule and then click .

View Performance Monitoring Job History

Use the **Job History** tab on the **Performance** page to verify whether a performance monitoring task was run. The performance monitoring job appears on the **Job History** tab only after the job has begun.

To view the job history of a performance monitoring task:

1. Select a project and click the **Performance** page.
2. Click the **Job History** tab. A table appears that lists the performance monitoring jobs that have been executed.

Here is a description of the columns in the job history table:

Job Name
is the name of the performance monitoring task.

Job Status
specifies whether the job status is **Running** or **Completed**.

Execution Status
shows a green indicator for a successful job execution. A yellow indicator shows that the performance monitoring task ran with warnings. A red indicator shows that the performance monitoring task ran with errors.

Date Started
is the date and time that the performance monitoring task started.

Date Completed
is the date and time that the performance monitoring task ended.

Log
is the revision number for the SAS log.

Output
is the revision number for the job output.

SAS Code
is the revision number for the performance monitoring task program.

Manage Performance Data Sets

After a performance monitoring task has run, the summary data sets reside on the **Results** tab on the **Performance** page.

To add a performance data set:

1. Click the **Results** tab.
2. Click +.
3. Navigate to the location of the data set and select the file to add.
4. Click **Open**.

Note: Fifteen tables are shown for the classification model function and thirteen are shown for the prediction model function. The table name must be the same as one of the shown tables; otherwise the uploaded table cannot be displayed. Tables with the same name are overwritten.

To delete the performance data sets:

1. Click the **Results** tab.
2. Click 🗑. Confirm the deletion.

Monitoring Performance of a Model without Score Code

If you want to monitor the performance of a model for which you no longer have the score code, you can import a model without SAS score code. If the performance data set contains the predicted values, the score.sas file can be empty.

To monitor the performance of a model without score code:

1. Prepare the following model files:
   - XML file that defines the model input variables (inputvar.xml)
   - XML file that defines the model output variables (outputvar.xml)
   - XML file that defines the model target variables (targetvar.xml)

- empty SAS score code file (score.sas)

2. Select **Models** ⇨ **Projects**
3. Create a project that has a model function type of **Classification** or **Prediction**. You can skip this step if you have already created a project.
4. Open a project and verify that the project properties are set.
   a. If it is a project that has a model function property value of **Classification**, verify that the following project properties are set:
      - Training target variable (for example, *bad*)
      - Target event value (for example, *1*)
      - Class target level as **`Binary`**
      - Output event probability variable (for example, **`score`**)
   b. If it is a project that has a model function property value of **Prediction**, verify that the following project properties are set:
      - Training target variable (for example, *lgd*)
      - Class target level as **`Interval`**
      - Output prediction variable (for example, **`p_lgd`**)
5. Select the **Models** page.
6. Click and select **from local files**.

   Note: If the model already exists, you can open a model to add model files to an existing model. For more information, see "Add Model Files to an Existing Model" on page 162.
7. Navigate to the folder on your computer that contains the component files for your model.
8. Select a classification or prediction template from the **Choose a model template** list.
9. Enter a text value in the model **Name** field.
10. Click **Properties** and specify the model properties.
11. Click **Files** and select the local files from the SAS Workspace Server that match the template files. You cannot delete a file after you have added it. To replace the file, select another file or cancel the import and start over. The following files are required:
    - inputvar.xml
    - outputvar.xml
    - targetvar.xml
    - score.sas

    Note: The filenames that you created for the model do not have to match the template filenames. However, the file contents must meet the file property requirements. For more information, see "Model Template Component Files" on page 340 or "Model Template Component Files" on page 340.
12. Click **OK**.
13. Open the model and set the model-specific properties. The value for the **Score code type** property must be set to **DATA step** or **Analytic store**.

14. Expand **Variables** and select **Output Mapping** in order to set the output variable mappings for the model. Select a value for each variable and click .
15. Click to close the model.
16. Select the model and click to set as the champion model. For more information, see "Ensure That Champion and Challenger Models Are Set" on page 257.
17. Before defining performance, verify that the performance data set is registered in the SAS Metadata Repository and is available in the Data category view. Make sure that the data set contains the following variables:
    - model input variables

      Note: You must have the variable columns in the table, but the values can be missing.
    - target variable
    - prediction variables
    - variables for characteristic analysis
18. Edit a project's performance definition on the **Performance** page. Specify the performance data set that contains the predicted values. Also, be sure to clear the **Run model score code** option for the **Data Processing Method** section of the **Edit Performance Definition** wizard. For more information, see "Edit and Execute a Performance Definition" on page 259.

Chapter 20

Using Dashboard Reports

Overview of Dashboard Reports

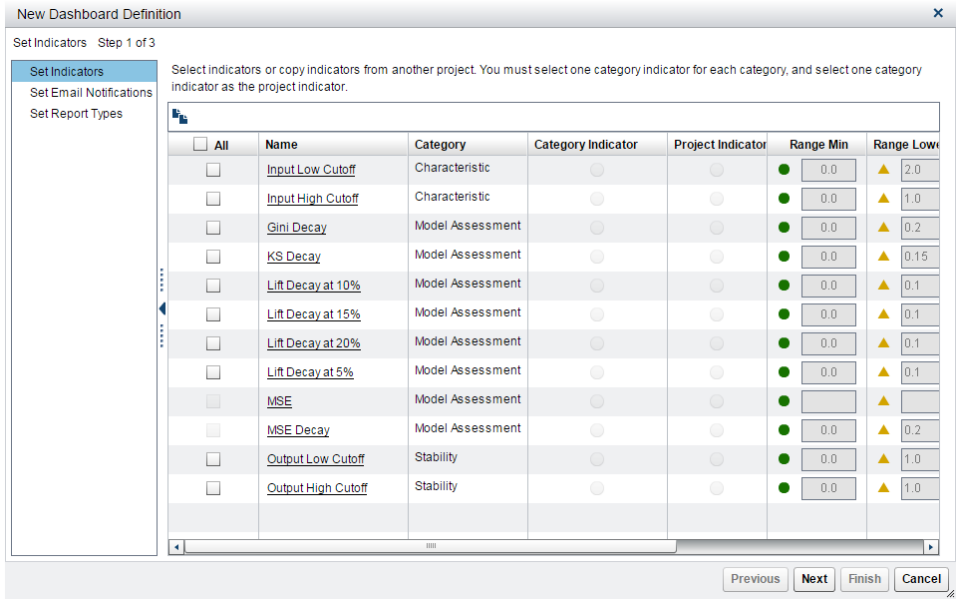
The SAS Decision Manager dashboard can provide reports that show the overall state of projects that are being monitored. The dashboard reports are produced from existing performance monitoring reports. For each project, you can define dashboard report indicators by creating a dashboard report definition. The dashboard report definition is used to create the dashboard reports. You view the dashboard reports through the **Dashboard** menu. These reports are generated in HTML.

Note: The dashboard reports can be defined and generated only by SAS Decision Manager administrators and advanced users.

Create a Dashboard Report Definition

To create a new dashboard definition:

1. Click and select **New Definition**. The New Dashboard Definition window appears.



2. (Optional) Click to copy indicators from another project. The Copy Indicators from a Project window appears.

   Select the indicators that you want to copy and click **OK**.

3. Select the indicators for the new dashboard definition.

4. Enter normal, warning, and alert values for the range definitions for each indicator that you have selected.

   Note: The Mean Squared Error (MSE) indicator does not have default values, since the values are based on the data source. The MSE measures the differences between the actual values and estimated values of the variable. For an ideal model, MSE should be 0. A model with a lower MSE has a smaller difference between the estimated value and the actual value.

5. Click **Next**.

6. Select one **Category Indicator** for each category, and one indicator as the **Project Indicator**.

   Note: The indicator that you select as a project indicator must also be a category indicator.

7. Click **Next**.

8. (Optional) Specify an email address for each recipient who should receive an email notification about the project status. Click +, select a project status, and enter an email address. Click **Save**.

   Note: If you send email notifications as part of the dashboard definition, the email notification contains a ZIP file (DashboardReports.zip) with the Dashboard reports in HTML format with the contents of the reports when they were generated. It also contains a link to the latest dashboard reports and status on the SAS Content Server.

9. Click **Next**.

10. Use the default selections, or select one or more reports to include in the dashboard reports.

11. Click **Finish**.

Note: You must define dashboard report indicators for all projects that you want to be included in your dashboard reports.

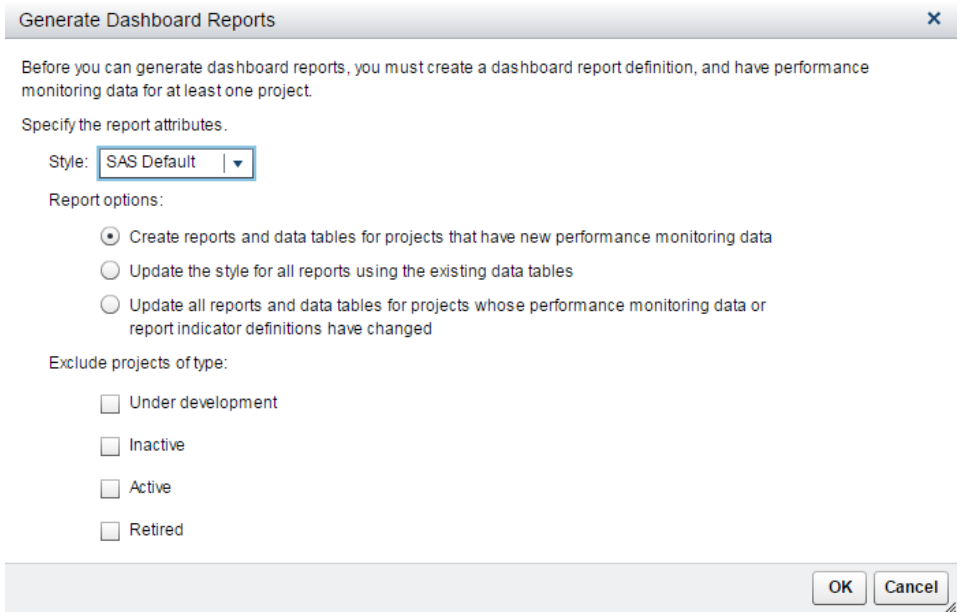
12. Click **Close**.

Generate Dashboard Reports

Note: Before you execute the dashboard report, ensure that at least one project contains performance data. At least one dashboard report indicator must also be defined in that project.

To generate dashboard reports:

1. Click and select **Generate Reports**. The Generate Dashboard Reports window appears.



2. Select a style.
3. Select a report option:
   - Create reports and data tables for projects that have new performance monitoring data.
   - Update the style for all reports, using the existing data tables.
   - Update all reports and data tables for projects whose performance monitoring data or report indicator definitions have changed.

     Note: This option regenerates the dashboard reports, even if the performance monitoring data or report indicator definitions have not changed.
4. (Optional) Select an option if you want to exclude one or more project types from the report.
5. Click **OK**. A confirmation window appears, stating that the dashboard report was created.
6. Click **Close**.

View Dashboard Reports

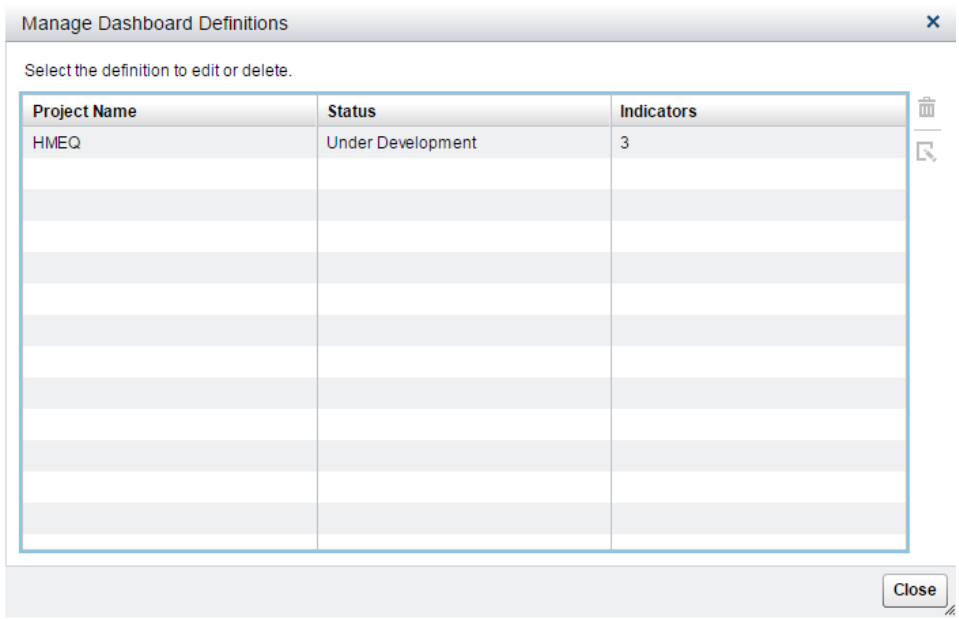
To view the dashboard reports:

1. Click and select **View Reports**. A web page displays all of the dashboard reports for each project that has a dashboard definition.
2. Select a project name or status link to view the associated dashboard report.
3. Select a link from the report column to view the report details.

Note: If you are configured for single sign-on web authentication and you try to view a report, the report is downloaded to your local drive. To view an HTML report, you must unzip the file.

Manage Dashboard Definitions

1. Click and select **Manage Definitions**. The Manage Dashboard Definitions window appears.



2. Select a definition to edit or delete.

   To edit a definition, click .

   The Edit Dashboard Definition window appears. Make your changes. Click **Finish**.

   To delete a definition, click .

   A confirmation message appears. Click **OK** to confirm the definition.

3. Click **Close**.

Edit a Dashboard Report Definition

To edit a dashboard definition for a specific project:

1. Select a project, click , and select **Edit Definition**. The Edit Dashboard Definition window appears.
2. Make your changes. Click **Finish**.
3. Click **Close**.

Chapter 21

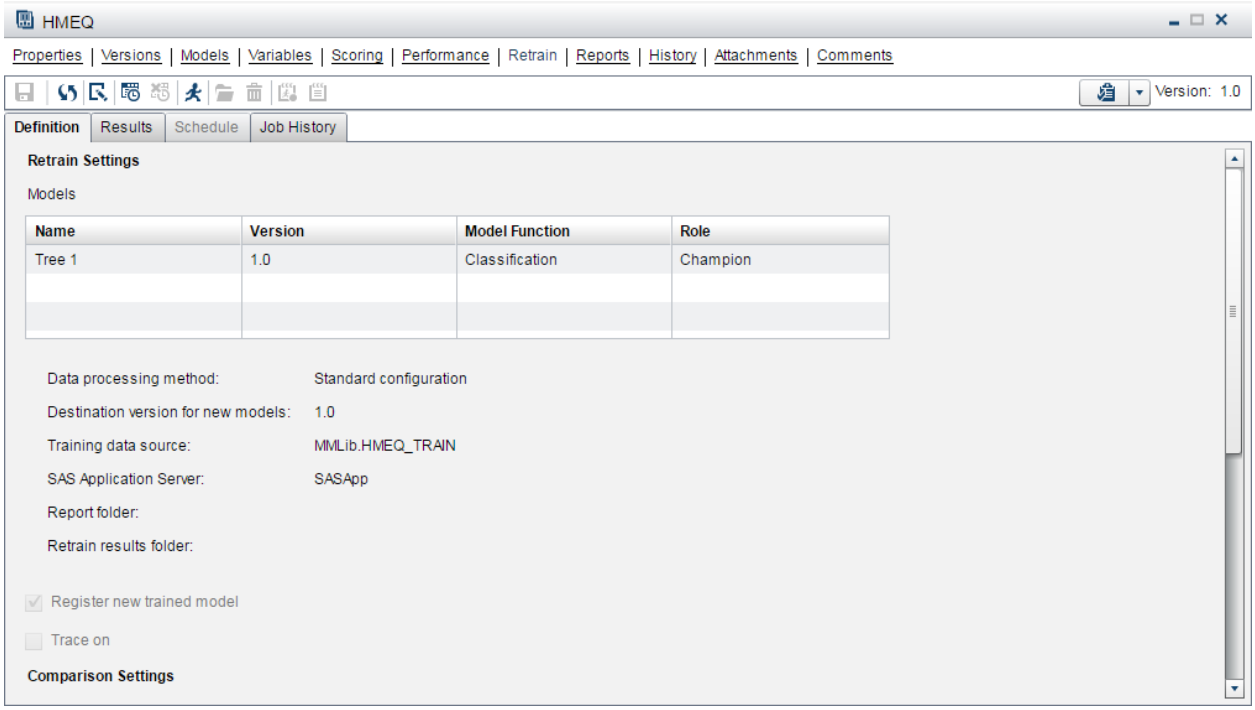
Retraining Models

Overview of Retraining Models

You can retrain models to respond to data and market changes. Retraining models enables you to update out-of-date models and improve model performance. When you edit a model retrain definition, you can select multiple models to be retrained at the same time. The retrain definition for a model includes the destination version and training data source. The destination version is an existing version or new version that is associated with the selected project and stores the retrained model information.

The training data source contains new data for retraining the selected models. You can also specify a location to store the comparison reports and retrain results. When you select the models to include in the comparison report, you can use the training data source or select a different data source to compare the performance of the new models. You can also specify the report options, including the name, format, and style of the comparison report. Email notifications can also be specified as part of a model retrain definition and are sent after you execute a model retrain definition.

By default, the champion model for the selected project is selected for retrain. If the **Register new trained model** option was selected after you execute a model retrain definition, the new models are registered to the destination version. The comparison report is available on the **Results** tab of the **Retrain** page. The definition is executed on the SAS Application Server that is specified. The report folder is stored on the SAS Content Server.



Note: Only R models and those that were created with SAS Enterprise Miner, SAS/STAT, and SAS/ETS can be retrained. SAS Factory Miner models and SAS Viya models that were created with SAS Visual Data Mining and Machine Learning modeling procedures cannot be retrained in SAS Decision Manager. Also note this: if a SAS model package (SPK) file was created with the **Model Comparison** node in SAS Enterprise Miner, the SPK file contains the score code for the best model and the training code for all of the models that were part of the model comparison. Therefore, when you are retraining the model in SAS Model Manager, be aware that the algorithm that is used for the retrained model could be different.

To retrain a model:

- Ensure that all prerequisites have been completed
- Edit the model retrain definition for a project to generate the SAS code that retrains models
- Execute the generated SAS code
- View the new models and comparison report

Prerequisites for Retraining a Model

Before you can retrain a model, complete the following prerequisites:

- If you want to retrain the project champion model, ensure that the champion model is set. For more information, see "Champion Models" on page 286.
- Verify that the training data set that you want to use as the training data source has been registered in the SAS Metadata Repository, and is available in the Data category view.
- Verify that the appropriate project and model properties are set:

Classification Model Project Properties
- Training target variable
- Target event value
- Class target level
- Output event probability variable

Prediction Model Project Properties
- Training target variable
- Class target level
- Output prediction variable

Model Properties
- Score code type

For more information, see "Project Properties" on page 129 and "Scoring Model Properties" on page 198.

- Verify that all of the project output variables are mapped to the corresponding model output variables. For more information, see "Map Model Variables to Project Variables" on page 164.
- Verify that the retrain file that is specified in the model template exists in the list of model files. The retrain file must appear on the **Model Properties** page for the model that you want to retrain. Ensure that the content is correct.
- In order to retrain SAS/STAT linear models, you must modify the training code in the train code file (for example, batch.sas).

To edit the model train code file:

1. Select and open a model.
2. On the **Model Properties** page, select **Advanced** ⇨ **Model Files**.
3. Double-click the train code file (batch.sas) to open it.
4. Select the **Edit** check box and make the following modifications to the code.
   - Replace the name of the training data set with the macro variable **`&_MM_InputDS`**.
   - Replace the name of the outmodel with **`smmmodel.outmodel`**.
   - Make sure that the name of the itemstore is **`work.itemstore`**.

Example Code 1 *SAS/STAT Logistic Model Training Code*

```
proc logistic data=&_MM_InputDS outmodel=smmmodel.outmodel;
class mstr_cstmr_age_grp days_claim_grp odm_last_veh_prchsd_grp
highest_edu_grp;
model bcr_ind=mstr_cstmr_age_grp days_claim_grp odm_last_veh_prchsd_grp
highest_edu_grp;
;
store work.itemstore;
quit;
```

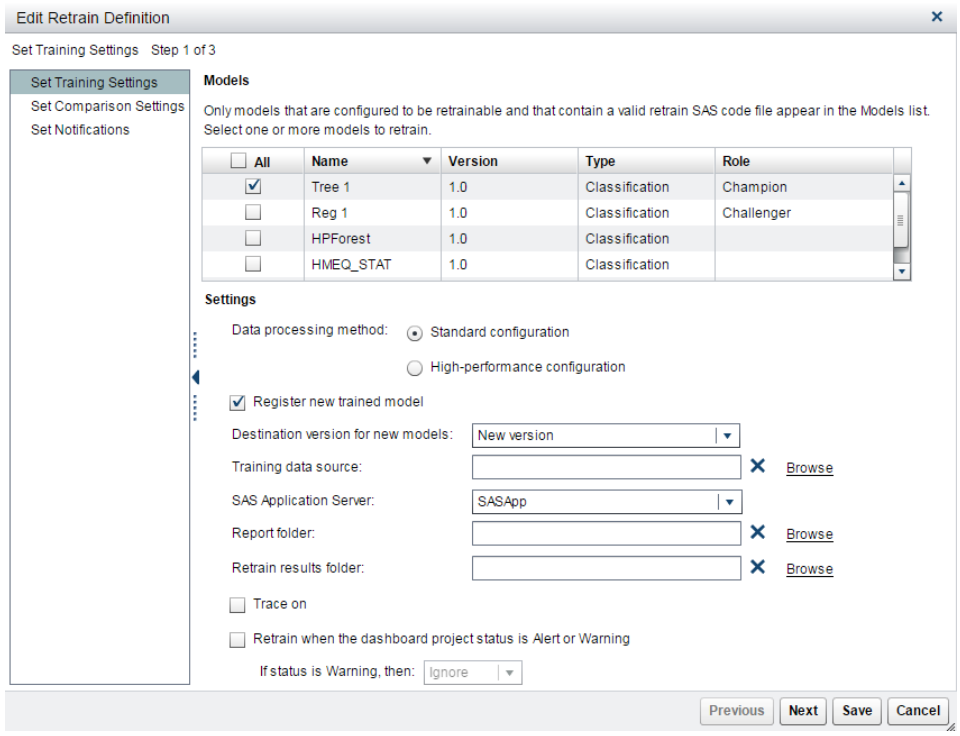
5. Click .

For more information, see "Using Macros to Register Models Not Created by SAS Enterprise Miner" in *SAS Model Manager: Macro Reference*.

Edit a Model Retrain Definition

To define a model retrain definition:

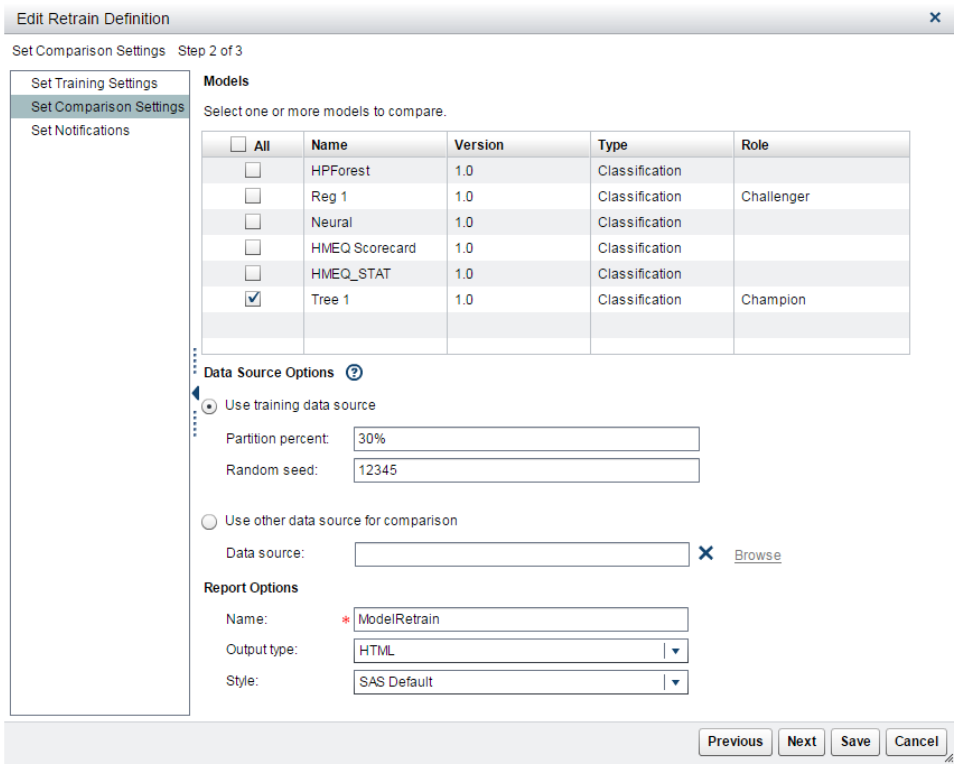
1. On the **Definition** tab on the **Retrain** page, click and select one or more models to retrain. By default, the champion model is selected if it can be retrained.



2. Select a data processing method.
   - To run a standard environment, select **Standard configuration**.
   - To run the performance monitoring definition in a High-Performance Analytics environment, select **High-performance configuration**.
3. (Optional) Select **Register new trained model** to register the new models in the destination version on the SAS Content Server.
4. (Optional) If **Register new trained model** is checked, select a destination version for new models. Select **New version** from the drop-down menu to create a new version for the models.
5. Click **Browse** to select a training data source from a library. Click **OK**.
6. Click the **SAS Application Server** list and select a server.
7. (Optional) Click **Browse** to select a report folder in which to store the comparison report.
8. (Optional) Click **Browse** to select a retrain results folder to store the model training results.
9. (Optional) Select **Trace on** to print trace information to the SAS log file.
10. (Optional) Select **Retrain when the dashboard project status is Alert or Warning**. If the dashboard project status is Alert, the model is automatically

retrained. If the dashboard project status is Warning, select whether to retrain the model or ignore the task. If the dashboard project status is Normal, the model will not be retrained.

11. Click **Next**.



12. Select the models to compare.

    Note: If you do not select a model, the champion model is used to perform the comparison.

13. Specify the data source options:

    - Select **Use training data source** to use the whole training data source to compare or partition it into two parts, based on partition percent and random seed. The percent that is specified is the percentage of data that is used for model comparison; the other part of the data is used for training. The random seed value is used to generate the training data based on the random sampling method.
    - Click **Browse** to select a performance data set as the comparison data source.

14. Specify the report options:

    - Enter a report name.
    - Select a format for the report output. The standard formats that are available are **RTF**, **PDF**, and **HTML**. The default is **RTF**.
    - Select a style for the report. The available styles are **SAS default**, **Seaside**, **Meadow**, and **Harvest**. The default is **SAS default**.

15. Click **Next**.

16. (Optional) To send the retrain results by email, click + and enter an email address.

17. Click **Save**.

Execute a Model Retrain Definition

The prerequisites for retraining a model must be completed and a model retrain definition must exist before you can execute a model retrain definition.

To execute a model retrain definition:

1. Click .
2. After the models are retrained, a confirmation message appears. Click **Close**.
3. Click the **Results** tab to view the results.

Note: You can check the status of a job by clicking and then selecting the **Results** tab or the **Job History** tab.

Schedule a Retrain Definition

After you create a retrain definition, you can create a schedule to execute the definition to run on a specific day and at a specific time. You can schedule the definition to run hourly, daily, weekly, monthly, or yearly.

Before you can schedule a retrain definition, your user ID and password must be made available to the SAS Metadata Repository. You must also sign in to SAS Decision Manager using your full user credentials that were specified for your user account in SAS Management Console. For user accounts where a Microsoft Windows user ID is specified, you must enter your user ID in the format of *domain\userID*. Contact your system administrator to add or update your password, and to determine the correct user credentials for your account.

CAUTION:

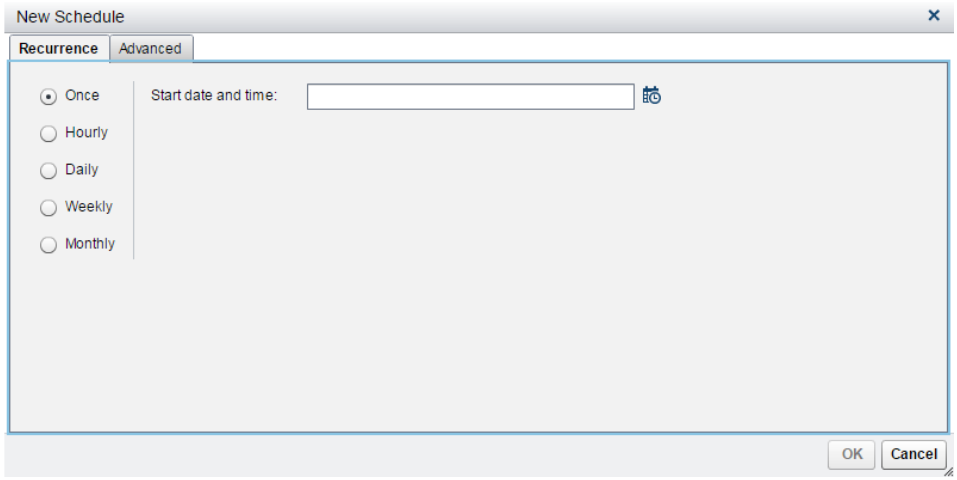
Check your configuration settings before scheduling jobs. Users who are configured for single sign-on authentication by Kerberos cannot schedule jobs.

You cannot edit a schedule for a retrain definition. To modify a schedule, delete the schedule and create a new schedule.

After retrain jobs execute, you can view the job history using the **Job History** tab on the **Retrain** page.

To schedule a retrain definition:

1. On the **Retrain** page, click .



2. On the **Recurrence** tab, select the recurrence pattern.
3. Specify the criteria for when and how often the job should be run.
4. (Optional) Click the **Advanced** tab.
   a. Select the server that schedules the job from the **Scheduling server** list box.

      Note: The SAS Distributed In-Process Services Scheduling Server is not supported.

   b. Select the batch server that runs the job from the **Batch server** list box.
   c. Click **Browse** to select a location for the output and click **OK**.
5. Click **OK**.
6. After the job has been scheduled, a confirmation message appears. Click **Close**.
7. Click the **Job History** tab to view the job status.
8. After the job has completed, click the **Results** tab to view the retrain results.

Note: Schedules cannot be edited. To change the schedule, delete the schedule and create a new schedule.

To delete a schedule, select the schedule and then click .

Viewing Retrained Models and Model Comparison Reports

After a model retrain definition is executed and if you chose to register the retrained models in the model retrain definition, the new retrained models are available in the destination version. In addition, the model retrain job creates a model comparison report, which is available in the **Results** tab on the **Retrain** page.

To view the retrain results:

1. Select the **Results** tab on the **Retrain** page.
2. You can view the model comparison reports in several ways:
   - Double-click a result in the list.
   - Select a result from the list and click .

Note: You can also view the SAS code and SAS log.

Note: If you are configured for single sign-on web authentication and you try to view a report, the report is downloaded to your local drive. To view an HTML report, you must unzip the file.

Part 6

Deploying and Publishing Models

Chapter 22

Deploying Models

Overview of Deploying Models

The goal of a modeling project is to identify a champion model that a scoring application uses to predict an outcome. SAS Decision Manager provides tools to evaluate candidate models, declare champion models, and inform your scoring officer that a predictive model is ready for validation or production.

To deploy a model, you might use the following scenario:

1. Identify the model that outperforms other candidate models and declare this model to be the project champion model. You can also flag challenger models for the champion model.
2. Test and validate the model before you declare the model ready for production.
3. Lock the champion version for the project to prevent changes to the champion model.
4. Publish the champion model and challenger models (optional) so that you can deploy them to a production environment.

Champion Models

About Champion Models

The champion model is the best predictive model that is chosen from a pool of candidate models. Before you identify the champion model, you can evaluate the structure, performance, and resilience of candidate models. When a champion model is ready for production scoring, you set the model as the champion model. The project version that contains the champion model becomes the champion version for the project. A check mark ✔ appears next to the version on the **Versions** page. You can publish the champion model to a database, the SAS Metadata Repository, and a SAS channel.

Requirements for a Champion Model

Before you identify a model as the champion, perform the following tasks:

- Register at least one model.
- Verify that the model is active. If the model expiration date has passed, you cannot set the model as a champion model.

  Note: However, an authorized user can reset the expiration date to a later date in order to set the champion model. To reset the expiration date, select the **Model Properties** page for the model.

You might use the following criteria to identify a champion model:

- model comparison reports that validate and assess the candidate models.
- business decision rules. For example, you might use a decision tree model because of difficulty interpreting results from a neural network model even when the neural network model outperforms the decision tree model.
- regulatory requirements, such as when the champion model should exclude certain specific attributes (age or race).

You can flag and publish a challenger model specifically for the purpose of comparison with the champion model. For example, your champion model for a production environment might omit restricted attributes during operational scoring because of regulatory requirements. You can use a challenger model that includes the restricted attributes in the development environment to evaluate its prediction power against the prediction of the champion model. Then you can determine the amount of predictive power that is lost because of the regulatory requirements.

Set a Champion Model

To set a champion model:

- On the **Models** page of a project, select a model and click ✔ to set the model as the project champion model. The value in the **Role** column changes to **Champion**.

Note: Alternatively, you can right-click a model and select **Set as champion**.

Clear a Champion Model

To clear a champion model:

- On the **Models** page of a project, select a model that is marked as Champion, and click ✕ to clear a flagged champion model.

Note: Alternatively, you can right-click a model that is marked as Champion and select **Clear**.

Challenger Models

About Challenger Models

You use challenger models to test the strength of champion models. The champion model for a project can have one or more challenger models. A model can be flagged as a challenger model only after a champion model for the project has been selected. A challenger model can be flagged in any version of a project.

Verify that the model is active. If the model expiration date has passed, you cannot set the model as a challenger model.

Note: An authorized user can reset the expiration date to a later date so that it is possible to set the challenger model.

To compare a challenger model to a champion model, you can create and run performance monitoring tasks for the champion model and any challenger models. Then, using the performance data, you can create a Champion and Challenger Performance report. You can also compare challenger models to the champion model using other reports such as the Delta report and Dynamic Lift report that are available through the **Reports** page. For more information, see "Champion and Challenger Performance Reports" on page 220.

Note: The batch programs for performance monitoring do not support creating challenger model performance reports.

Challenger models can be published to a database, the SAS Metadata Repository, or to a SAS channel that contains the champion model. They can also be published by themselves. If testing determines that the challenger model is the better model, you can replace the champion model by setting the challenger model as the champion model.

Flag a Challenger Model

To flag a challenger model:

- On the **Models** page of a project, select a model and click ⚑ to flag a model as a challenger to the project champion model. The value in the **Role** column changes to **Challenger**.

Note: Alternatively, you can right-click a model and select **Flag as challenger**.

Clear a Challenger Model

To clear a challenger model:

- On the **Models** page of a project, select a model that is marked as Challenger, and click ✕ to clear a flagged challenger model.

Note: Alternatively, you can right-click a model that is marked as Challenger and select **Clear**. Challenger models can also be cleared when the champion model is cleared or replaced.

Locking Versions

About Locking Versions

You must be a SAS Decision Manager administrator to lock and unlock a version. Administrators can lock a project version to prevent users from modifying some properties and files for the version's models. The champion version can be locked when the project champion model is approved for production or is pending approval. After a project version is locked, users cannot perform the following tasks:

- import, delete, or copy and paste a model
- rename a model
- change the champion model
- flag a model as a challenger

SAS Decision Manager administrators remain authorized to perform these activities. If the champion model is not deployed to an operational environment, then an administrator can unlock a version so that users can change the models. Advanced users can still modify the **Attachments**, **Reports**, and **Scoring** pages after a version is locked.

When the champion model has been used in production scoring, you must unlock the model if you want to change the contents of the champion version. However, use caution in modifying the version content. If the model UUID and revision number for the score code in production scoring environments are always recorded, then you can modify a version even after the version is deployed to production environment.

If you attempt to delete a project that contains a locked version, SAS Decision Manager displays a message indicating that you cannot delete a project that contains locked versions. An administrator must unlock the versions before the project can be deleted.

Lock a Version

Locking a version restricts the activities that you can do with the project. You normally lock a version after you declare a champion model in preparation for deploying the champion model to a production environment.

To lock a version:

1. Open a project and select the **Versions** page.
2. Select a version and click 🔒 to lock the version. The label **Locked** appears after the version name.

Unlock a Version

If changes to a model are required after the version is locked, a SAS Decision Manager administrator can unlock the version.

To unlock a version:

1. Open a project and select the **Versions** page.
2. Select a version and click to unlock the version.

For more information about versions, see "Lock and Unlock a Project Version" on page 136.

Chapter 23

Publishing Models

Overview of Publishing Models

SAS Decision Manager provides a comprehensive publishing environment for model delivery that supports sharing performance and scoring data. SAS Decision Manager publishes models to different channels, and to the SAS Metadata Repository. SAS Decision Manager can also publish classification, prediction, and segmentation (cluster) models that have the score code type of DATA step to a database or Hadoop. Models that have a score code type of analytic store can be published only to Hadoop and Teradata. Application software, such as SAS Data Integration Studio or SAS Enterprise Guide, enables you to access models through the SAS Metadata Server and to submit on-demand and batch scoring jobs.

SAS Decision Manager publishes models to defined publication channels. Authorized users who subscribe to a channel can choose to receive email notifications when updated models are ready to deploy to testing or production scoring servers, and are published to a publication channel. From a publication channel, you can extract and validate the scoring logic, deploy champion models to a production environment, and monitor the performance of your models.

Models can also be published from the **Models** page. You can publish champion and challenger models from a model project to the SAS Metadata Repository. The publish history of models can be viewed on the **Models** page and on the **Published** tab on the **History** page. You can also remove models that have been published to a database.

Publishing Models to a SAS Channel

SAS Decision Manager uses the SAS Publishing Framework to publish models to defined channels. The SAS Publishing Framework notifies subscribers of the publication channel when the models are delivered. You can publish models in the Projects category view. SAS Decision Manager creates a SAS package (SPK) file for the model in a publication channel. A user who subscribes to the publication channel can choose to receive email that includes the SAS package as an attachment.

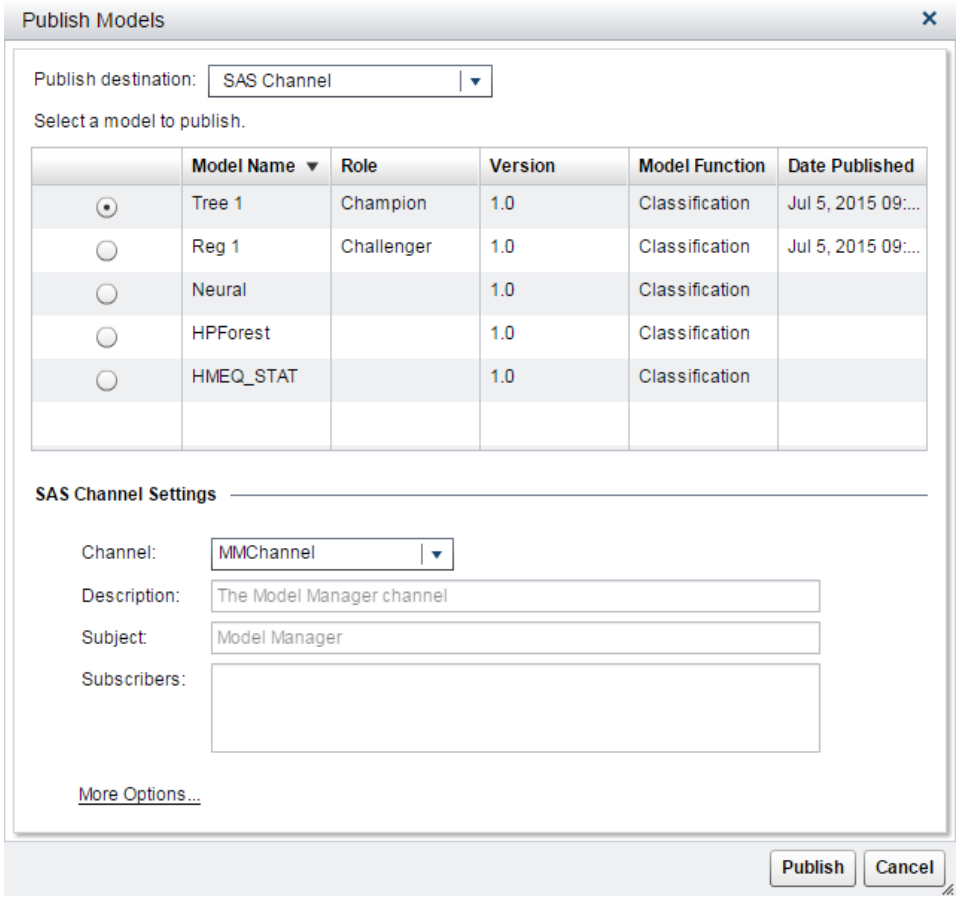
Note: Before you can deploy a model to a publications channel, a SAS administrator must configure the publication channel in SAS Management Console to publish models as archive (binary .SPK) files to a persistent store location. The archive persistent store location is specified as a physical file location, an FTP server, an HTTP server, or a path in WebDAV.

The **Report** attribute for a file element in a model template indicates whether SAS Decision Manager includes a file in the SAS package. You use the SAS Package Reader or a file archiver and compression utility, such as WinZip, to view the contents of the SPK file. SAS Decision Manager provides SAS macro programs to extract published models and deploy the models on testing and production scoring servers. The SAS package might contain additional files, depending on the number of file elements in the model template that have a Report attribute.

Note: The REF file contains the URL for a folder location in the project, such as **`http://`***MMServer:7980***`/SASContentServer/repository/default/ModelManager/MMRoot/`***organizational folder/project/version/Models/model_name***`/score.sas`**.

To publish a model to a channel:

1. Select a project and click .
2. Select **SAS Channel** from the publish destination list.
3. Select the model that you want to publish from the models list.
4. Select a publication channel from the channel drop-down list.



5. (Optional) Click **More Options** to specify a message subject, notes, and user-defined properties. Click **Save**.
6. Click **Publish**.

Publishing Models to the SAS Metadata Repository

About Publishing Models to the SAS Metadata Repository

SAS Decision Manager publishes a model by creating a MiningResults object in the SAS Metadata Repository. You can use the model information in the MiningResults object to set up a scoring environment. A scoring application can use SAS Data Integration Studio or SAS Enterprise Guide to access the metadata and run a batch job or stored process that executes the score code. SAS Real-Time Decision Manager can also read the metadata and use it in that process environment. Therefore, when you publish a project champion model, challenger model, or other models (with proper configuration), the scoring application always uses the most current champion model. The project champion and challenger models can be published from the project level and only the project champion models can be published from the portfolio level.

Note: SAS Decision Manager cannot publish R models.

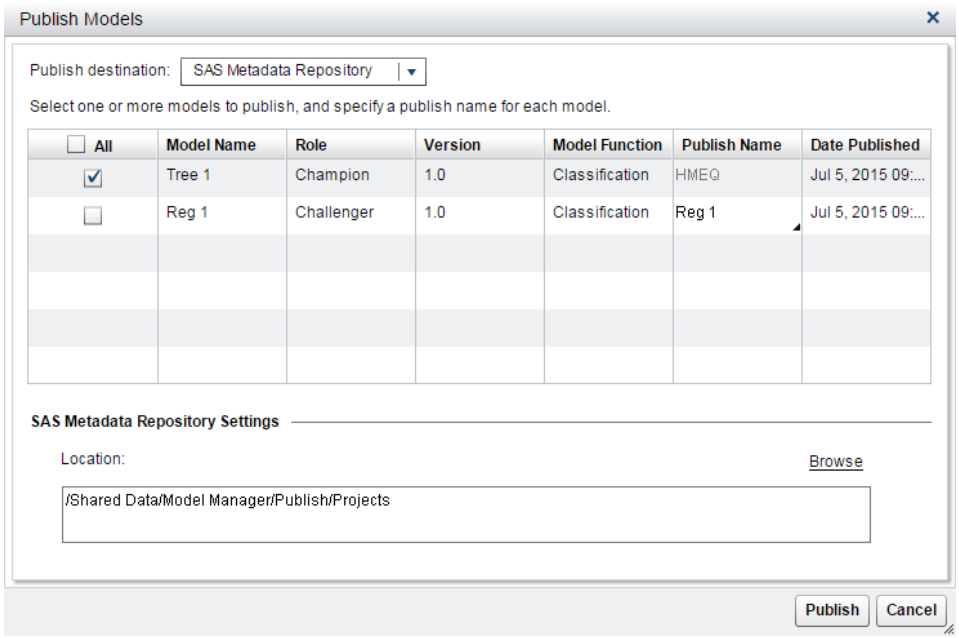
A user can publish a model to any accessible folder with Write permission, including all folders in the SAS Foundation repository and folders in custom repositories that are created in SAS Management Console to reflect the structure of your business organization.

Publish Project Champion and Challenger Models to the SAS Metadata Repository

To publish champion and challenger models from a model project to the SAS Metadata Repository:

1. From the Projects category, select a project and click .
2. Select **SAS Metadata Repository** from the publish destination list.
3. Select one or more models to publish from the models list.
4. Specify a **Publish Name** for each model.

   Note: You cannot modify the publish name for a champion model.
5. Click **Browse** and select the location to publish the model to.



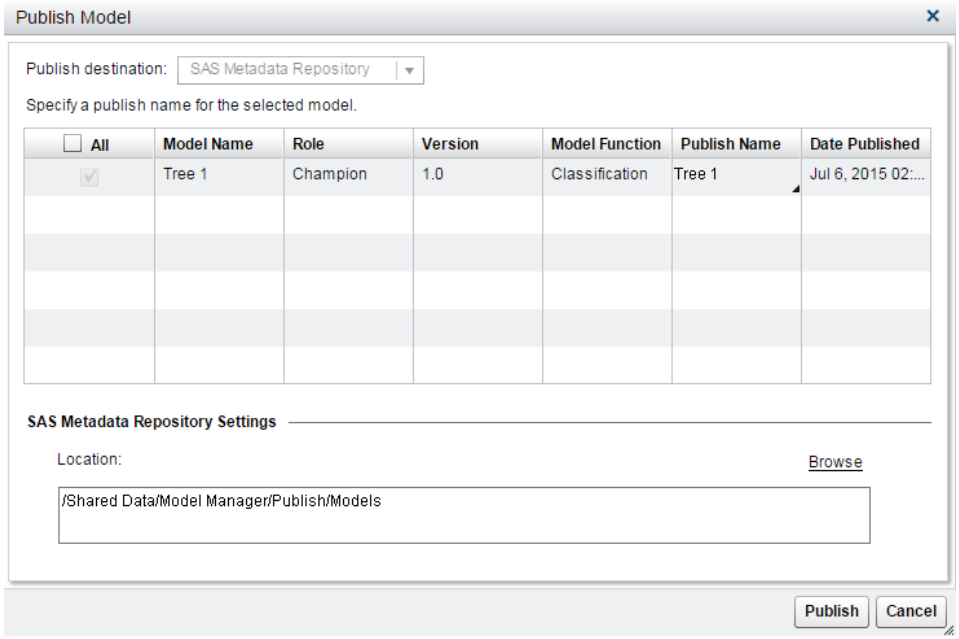
6. Click **Publish**.

Publish a Model to the SAS Metadata Repository

To publish a model to the SAS Metadata Repository:

1. On the **Models** page, select a model and click .

   Note: Alternatively, you can right-click a model and select **Publish**.
2. Specify a publish name for each model.
3. Select the location to publish the models to.



4. Click **Publish**.

Publishing Models to a Database or Hadoop

About Publishing Models to a Database or Hadoop

SAS Decision Manager enables you to publish the project champion model and challenger models that are associated with the **DATA Step** score code type to a configured database. SAS Decision Manager uses the SAS Scoring Accelerator and SAS/ACCESS interface to the database to publish models to the database or Hadoop Distributed File System (HDFS). The Scoring Accelerator takes the models from SAS Decision Manager and translates them into scoring files or functions that can be deployed inside the database. After the scoring functions are published using the SAS/ACCESS interface to the database, the functions extend the database's SQL language and can be used in SQL statements such as other database functions. After the scoring files are published, they are used by the SAS Embedded Process to run the scoring model.

If the scoring function publish method is chosen, the scoring metadata tables in the database are populated with information about the project and pointers to the scoring function. This feature enables users to review descriptions and definitions of the published model. The audit logs track the history of the model's usage and any changes that are made to the scoring project.

For more information about the SAS Scoring Accelerator, see the SAS In-Database Technology page available at http://support.sas.com.

Prerequisites for Publishing to a Database or Hadoop

The following prerequisites must be completed before users can publish a model scoring function using the scoring function publish method, or publish a model's scoring files using the SAS Embedded Process publish method:

- The user must have the proper authorization to publish approved models from SAS Decision Manager to the database or HDFS for SAS In-Database scoring.

  Note: When using the SAS Embedded Process publish method to publish models from SAS Decision Manager to a Teradata database, you must have a user database with the appropriate user permissions in order to validate the scoring results. For more information, see "Teradata Permissions for Publishing Formats and Scoring Models" in *SAS In-Database Products: Administrator's Guide*.

  Note: If your system is configured for Kerberos authentication, each user must have a valid Kerberos ticket. Also, you must complete post-installation configuration steps to enable users to publish models from the SAS Decision Manager application. For more information, see "Configure Users Authenticated by Kerberos for Publishing Models" in *SAS Decision Manager: Administrator's Guide*.

- The champion model for the project must be set.

- A predictive (classification or prediction), analytical, or segmentation model must have been selected for production scoring deployment via SAS Decision Manager.

  SAS Decision Manager can publish models to a database or Hadoop when they are associated with the **DATA step** score code type. Models that have a score code type of **Analytic store** can be published only to Hadoop and Teradata. Models that have a score code type of **SAS Program**, **PMML**, or **DS2** cannot be published to a database.

  The score code component file (score.sas) is DATA step score code and is used as input by the SAS Scoring Accelerator when publishing a model to a database. When you use the scoring function publish method, some SAS language elements and syntax are not supported when you create or modify your score code. Only the SAS language elements and syntax that are required to run critical data transformations and model scoring functions are available. If you use a statement or function that is not supported, an error occurs and your model is not published to the database. For more information, see "Considerations When Creating or Modifying DATA Step Score Code" in *SAS In-Database Products: User's Guide*.

  The SAS Scoring Accelerator requires three files when you are publishing a SAS analytic store model. The files are score.sas, score.sasast, and score.xml. The score.xml file is generated at publish time by SAS Decision Manager. For more information, see "Introduction to Analytic Store Scoring" in *SAS In-Database Products: User's Guide*.

- A database must have been configured to install scoring functions or model scoring files.

- If the model contains user-defined formats, a file that contains the user-defined formats must be attached to the version and stored in a format catalog.

- The following prerequisites are only for the scoring function publish method.

  - (Optional) A project user-defined property **DbmsTable** is defined for the champion version of the project from which to publish the scoring function.

    Note: The **DbmsTable** property must be defined if you plan to use a scoring application or SQL code to score your model.

  - The JDBC driver must be accessible from the middle-tier server when using the scoring function publish method.

  - The scoring function metadata tables are required in the target database if the **Metadata usage** option is enables in SAS Management Console.

Make User-Defined Formats Available When Publishing Models to a Database

In order to publish models with user-defined formats to a database, you must make the user-defined formats available to SAS Decision Manager.

To make the user-defined formats available for publishing:

1. Translate the user-defined formats SAS catalog (formats.sas7bcat) that was created with the model into a formats.cport file.

Here is an example:

```
filename tranfile "C:\formats.cport";
libname source "C:\myformats";

proc cport library=source file=tranfile memtype=catalog;
run;
quit;
```

2. Attach the formats.cport file to the version that contains the project champion model or challenger models. For more information, see "Attach a Portable Formats File" on page 136.
3. Send a request to the SAS administrator and ask them to either put the user-defined formats catalog (formats.sas7bcat) in the *SASConfigDirectory*\Lev1\SASApp \SASEnvironment\SASFormats directory or add the LIBNAME definition for the formats library to the *SASConfigDirectory*\Lev1\SASApp \appserver_autoexec_usermods file.

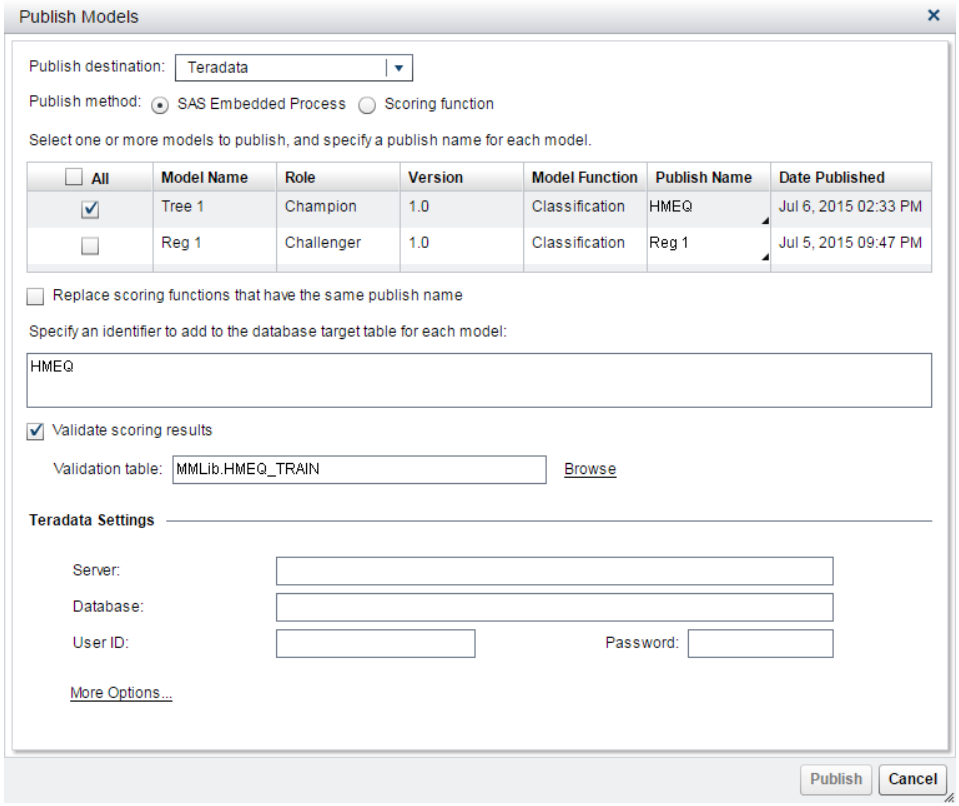
Here is an example of a LIBNAME definition:

```
libname mylib "C:\myformats";
options fmtsearch = (mylib.formats);
```

How to Publish Models to a Database or Hadoop

To publish a model to a database:

1. Select a project and click  .



2. Select a publish destination from the list. The publish destination can be either a database or a Hadoop Distributed File System.
3. Select a publish method. Specifies the method to use when publishing the scoring function or model files to the database.
4. Select one or more models to publish from the models list.
5. To specify a new publish name, click in the **Publish Name** column for a selected model. Specifies the name to use when publishing a scoring function or model files to the database. The publish name is a user-defined value that can be modified. The SAS Embedded Process publish method uses the **Publish Name** as the model name to publish the model files to the database. The scoring function publish method has a system-generated **Prefix** and the **Publish Name** that makes up the scoring function name. These are used to publish the model scoring function. The prefix portion of the scoring function name is 11 characters long and is in the format of **`Yyymmddnnn_`**:
   - **`Y`** is a literal character and is fixed for all prefixes.
   - **`yy`** is the two-digit year.
   - **`mm`** is the month and ranges from 01 to 12.
   - **`dd`** is the day and ranges from 01 to 31.
   - **`nnn`** is a counter that increments by 1 each time that a scoring function completes successfully. The value can range from 001 to 999.
   - _ is the underscore that ends the prefix.

   The **`yymmdd`** value in the prefix is the GMT timestamp that identifies the date on which you published the model. An example of a function name is **`Y081107001`**_*user_defined_value*. Here are the naming convention requirements:

Here are the naming convention requirements for the publish name:

- The user-defined value is case insensitive. The maximum length of alphanumeric characters is determined by the database type and publish method that is selected. No spaces are allowed. An underscore is the only special character that can be included in the publish name.
- The recommended maximum lengths of the publish name for the scoring function publish method are the following:
  - 19 alphanumeric characters for Teradata
  - 32 alphanumeric characters for Netezza, Greenplum, and DB2

UNIX Specifics
The publish name (user-defined) portion of the function name in an AIX environment has a maximum length of 16 alphanumeric characters for Teradata.

- The recommended maximum length of the publish name for the SAS Embedded Process publish method is 32 alphanumeric characters for all database types. The database types that are currently supported by SAS Decision Manager are Teradata, Oracle, Greenplum, and DB2.

The value of the publish name is validated against the target database, when the option **Replace scoring files that have the same publish name** is not selected for the SAS Embedded Process publish method. If the publish name is not unique, an error message is displayed.

Note: The default format of the publish name is configured by the SAS administrator.

6. (Optional) Select whether to **Replace scoring files that have the same publish name**. Specifies to replace the model scoring files that have the same publish name when you are using the SAS Embedded Process publish method. The value of the publish name is validated against the target database when this option is not selected. If the publish name is not unique, an error message is displayed.
7. Specify an identifier to add to the database target table for each model. Specifies the value of the identifier that is added to each model in the database so that the Database administrator or other users can query the database. The default value is the project name. This option is available only for the SAS Embedded Process publish method.
8. (Optional) Select whether to **Validate scoring results**. Specifies to validate the scoring results when publishing a model scoring function or model scoring files. This option creates a benchmark scoring result on the SAS Workspace Server using the DATA Step score code. The scoring input data set is used to create an equivalent database table. Scoring is performed using the new scoring function or model scoring files and database table. The scoring results are then compared. If selected, click **Browse** to navigate to the appropriate training table. The default train table that is specified in the properties of the published model is used by default.

   Note: When using the SAS Embedded Process publish method to publish models from SAS Decision Manager to a Teradata database, your user database must have the appropriate user permissions to validate the scoring results. For more information, see "Teradata Permissions for Publishing Formats and Scoring Models" in *SAS In-Database Products: Administrator's Guide*.
9. Specify the database settings.

   Note: For Hadoop and Teradata, when you are using the SAS Embedded Process publish method, the User ID and Password settings are hidden if your system is configured for Kerberos authentication. For Netezza, when you are publishing

using the Scoring function publish method, the name of the database is case sensitive.

Here are the available database settings according to the publish method and database type:

Database Settings	SAS Embedded Process	Scoring Function
Server	• Teradata • Oracle • Netezza • Greenplum • Hadoop • DB2 • SAP HANA	• Teradata • Netezza • Greenplum • DB2
Database	• Teradata • Oracle • Netezza • Greenplum • DB2	• Teradata • Netezza • Greenplum • DB2
Instance number	SAP HANA	Not applicable
User ID	• Teradata • Oracle • Netezza • Greenplum • DB2 • Hadoop • SAP HANA	• Teradata • Netezza • Greenplum • DB2
Password	• Teradata • Oracle • Netezza • Greenplum • DB2 • Hadoop • SAP HANA	• Teradata • Netezza • Greenplum • DB2
Server user ID	Not applicable	DB2
Compile database	Not applicable	Netezza
Jazlib database	Not applicable	Netezza

Database Settings	SAS Embedded Process	Scoring Function
Schema	• Oracle • Greenplum • DB2 • SAP HANA	• Greenplum • DB2
Initial wait time (in seconds)	Not applicable	DB2
FTP time out (in seconds)	Not applicable	DB2
Directory path	Hadoop	Not applicable

For a description of each database setting, see "Descriptions of Database Settings" on page 302.

10. Click **More Options** to specify other options for the database.

    Keep scoring function if validation fails (scoring function) or **Keep scoring files if validation fails** (SAS Embedded Process)
    specifies to save the scoring function or model scoring files if the validation of the scoring results fails. Saving the scoring function or model scoring files is useful for debugging if validation fails.

    Sample size
    specifies the size of the sample to use for validating the scoring function or model files. The default value is 100. The maximum number of digits that are allowed is 8.

    Display detailed log messages
    provides detailed information, which includes warnings and error messages that occur when you publish a scoring function or scoring model files.

    Use model input
    specifies to use the selected model input when publishing the scoring function or model files instead of using the project input, which is the default. This is useful when the project input variables exceed the limitations for a database.

Here are the limitations for the number of model input variables when publishing a champion model or challenger model to a database:

Database Type	SAS Embedded Process	Scoring Function
DB2	The maximum depends on the page size of the database tablespace. For a 4K page size database, the limit is 500. If you have it configured for any of the larger page sizes (8K, 16K, 32K), then the limit is 1012.	90
Greenplum	1660	100

Database Type	SAS Embedded Process	Scoring Function
Hadoop	No limit	Not applicable
Netezza	1600	64
Oracle	1000	Not applicable
SAP HANA	1000	Not applicable
Teradata	If you use Teradata version 13.1 or 14.0, the maximum is 1024. If you use the SAS Embedded Process and Teradata version 14.10, the maximum is 2048.	128

Protected mode (Teradata only)
specifies the mode of operation to use when publishing a model using the scoring function publish method. There are two modes of operation, protected and unprotected. You specify the mode by selecting or deselecting the **Protected mode** option. The default mode of operation is protected. Protected mode means that the macro code is isolated in a separate process from the Teradata database, and an error does not cause database processing to fail. You should run the Publish Scoring Function in protected mode during validation. When the model is ready for production, you can run the Publish Scoring Function in unprotected mode. You might see a significant performance advantage when you run the Publish Scoring Function in unprotected mode.

Fenced mode (DB2 and Netezza only)
specifies the mode of operation to use when publishing a model using the scoring function publish method. There are two modes of operation, fenced and unfenced. You specify the mode by selecting or deselecting the **Fenced mode** option. The default mode of operation is fenced. Fenced mode means that the macro code is isolated in a separate process from the DB2 database, and an error does not cause database processing to fail. You should run the Publish Scoring Function in fenced mode during validation. When the model is ready for production, you can run the Publish Scoring Function in unfenced mode. You might see a significant performance advantage when you run the Publish Scoring Function in unfenced mode.

11. Click **Publish**.

Descriptions of Database Settings

The following are descriptions of the database settings that are used for publishing models.

Server
specifies the name of the server where the database resides. The port can be included as part of the server name (for example, *server-name*:*port*). When you are using the scoring function publish method, the default port is used if you do not specify the port as part of the server name.

Note: The port must be included as part of the server name for SAP HANA SPS09 or higher. For SAP HANA SPS08, you must use the instance number setting instead of adding the port to the server name.

Database
specifies the name of the database.

Note: For Netezza, the name of the database is case sensitive if you are publishing models using the Scoring function publish method.

User ID
specifies the user identification that is required to access the database.

Note: For Hadoop or Teradata, this setting is hidden if your system is configured for Kerberos authentication.

Password
specifies the password that is associated with the **User ID**.

Note: For Hadoop or Teradata, this setting is hidden if your system is configured for Kerberos authentication.

Server user ID (DB2 only)
specifies the user ID for SAS SFTP. This value enables you to access the machine on which you have installed the DB2 database. If you do not specify a value for **Server user ID**, the value of **User ID** is used as the user ID for SAS SFTP.

Schema (Greenplum, Oracle, and DB2)
specifies the schema name for the database. The schema name is owned by the user that is specified in the **User ID** field. The schema must be created by your database administrator.

Initial wait time (DB2 only)
specifies the initial wait time in seconds for SAS SFTP to parse the responses and complete the SFTP –batch file process.

Default: 15 seconds

FTP time out (DB2 only)
specifies the time-out value in seconds if SAS SFTP fails to transfer the files.

Default: 120 seconds

Compile database (Netezza only)
specifies the name of the database where the SAS_COMPILEUDF function is published.

Default: SASLIB

See Also: For more information about publishing the SAS_COMPILEUDF function, see the *SAS In-Database Products: Administrator's Guide*.

Jazlib database (Netezza only)
specifies the name of the database where the SAS 9.3 Formats Library for Netezza is published.

Default: SASLIB

Instance number (SAP HANA only)
specifies the instance number.

Note: The instance number must be used for SAP HANA SPS08, instead of adding the port to the server name. For SAP HANA SPS09 or higher the port must be included as part of the server name and the instance number is ignored.

Directory path (Hadoop only)
specifies the directory path for the server.

Remove Published Models

The SAS Embedded Process publish method enables you to replace the model scoring files, but the scoring function publish method publishes the model as a separate entry in the database each time. The Remove Published Models feature enables you to remove previously published models, so that you can clean up the test or production database. If you modify the previously published models or change the champion model or challenger models after you have published models to a database or Hadoop, you can remove them to clean up the database for future publishing of models.

To remove published models:

1. Select **Actions** ⇨ **Remove Published Models**.
2. Select the publish destination and then specify the database or Hadoop settings. Click **Log On**.
3. Select the models that you want to remove from the database.
4. Click **Remove Models**. A warning message appears.
5. Click **Yes**.

View Publish History

To view the publish history of a model, select the **Models** page. To view the publish history of all models, select the **Published** tab on the **History** page. All models that have been published to a SAS Channel, to the SAS Metadata Repository, and to a database are displayed. Select a model from the list to view the full publish details.

To view the full publish details for a model:

1. Open a model and select the **Model Properties** tab.
2. Select **History** ⇨ **Published** to view the publish history.

To view the full publish details for all models:

1. Open a project and select the **History** page.
2. Select the **Published** tab to view the publish history.

Part 7

Using SAS Workflow with SAS Decision Manager

Chapter 24

Starting a Workflow and Working with Tasks

Overview of Using Workflows

SAS Decision Manager uses the Workflows and My Tasks category views to use SAS Workflow. A *workflow* is a copy of a workflow template. A workflow can be used to track the progress of objects, such as model projects and rule flows at the version level. An authorized user can use SAS Workflow Studio to define workflow templates and to make them available to SAS Decision Manager for use. Workflow templates contain the set of tasks, participants, policies, statuses, and data objects that comprise a business task. The status that you select when completing a task determines the next task in the workflow.

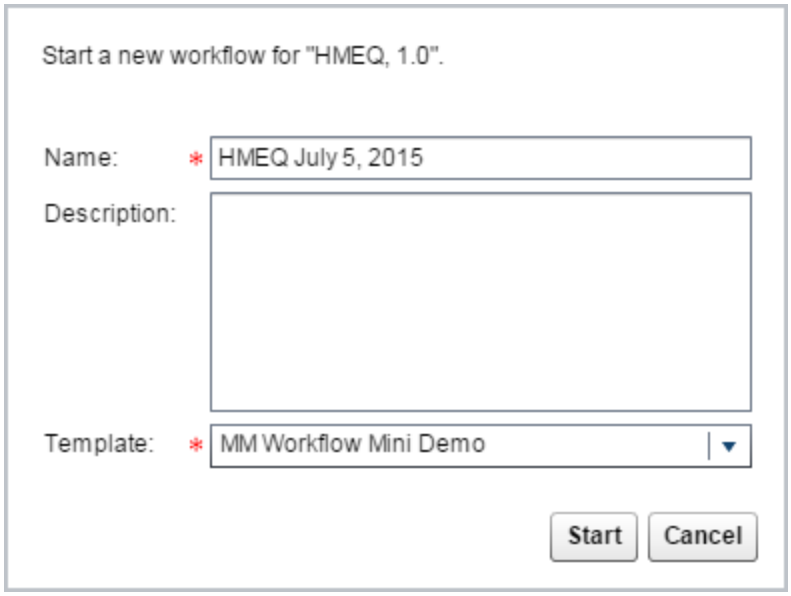
All users can access the My Tasks category view. By default, only users that are in the Decision Manager Common Administrators group can access the Workflows category view.

For more information about user permissions, see *SAS Decision Manager: Administrator's Guide*

Start a New Workflow

When you start a new workflow, it is associated with the selected version of a project or rule flow. For a specific version, only one workflow can be in progress at a time. To start another workflow for the same version, you must first complete the in-progress workflow, or terminate the in-progress workflow process. A workflow can be started only for a version of a rule flow that is in the state of Current.

1. Open a project or rule flow.
2. Click on the right-side of the object toolbar.

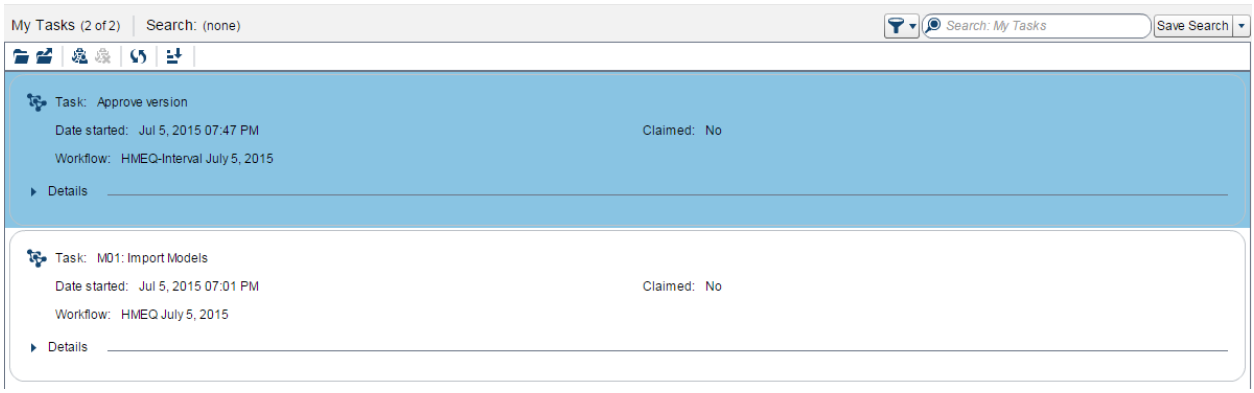


3. Enter a name for the new workflow.
4. (Optional) Enter a description for the workflow.
5. Select a template from which to create the workflow.
6. Click **Start**.

For more information, see Chapter 25, "Managing Workflows," on page 311.

Working with Workflow Tasks

The My Tasks category view displays the tasks for In Progress workflows that you have been assigned to as a potential owner or that have been claimed by you.



From the My Tasks category view, you can perform the following:

- open a task that pertains to the associated object
- claim and open a task that pertains to the associated project or rule flow
- claim a task
- release a task

- view the task details and workflow diagram

To complete a task:

1. Select a task and click in order to open the associated project or rule flow and perform the task.
2. Navigate through the project's or rule flow's pages to perform the steps for the current task.
3. Click .
4. Select an action to take for the selected task. The actions that are available are the status values for the task in the workflow.
5. Click **Done**. The workflow process continues to the next task.

Note: Only a business administrator who has access to the Workflows category can release a task that has been claimed by another participant. For more information, see "Release a Task" on page 315.

Chapter 25

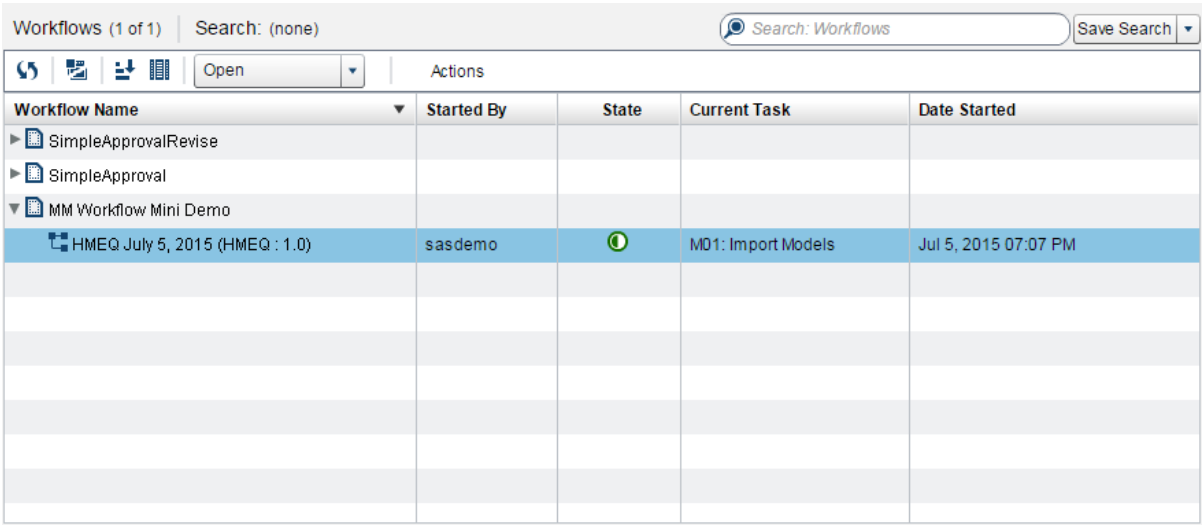
Managing Workflows

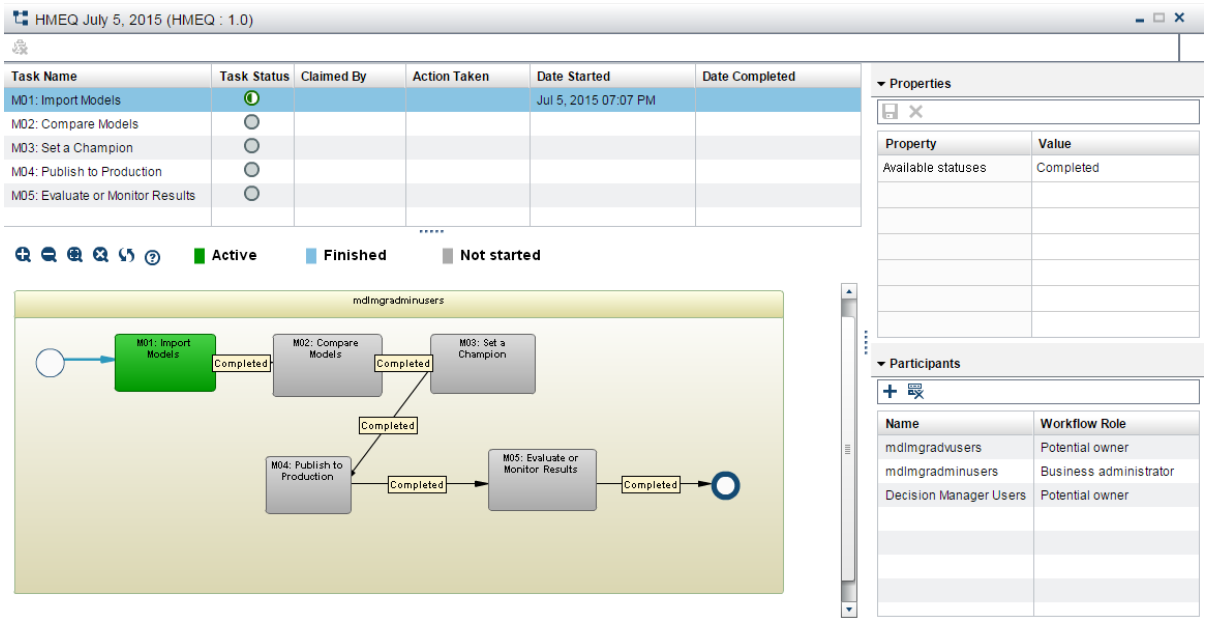
Overview of Managing Workflows

SAS Decision Manager can be used to manage workflows. You can create new workflows, view workflows, and interact with tasks that are associated with a workflow. If a user is assigned to the workflow role of business administrator, they can influence the progress of a task by actions such as assigning a task, or releasing the task that is claimed by another user, as well as specify values for properties to share information with other users. After the workflow templates are made available, an application administrator can set the object mappings using the Workflows category view. Each workflow consists of tasks.

Note: By default, only users that are in the Decision Manager Common Administrators group can access the Workflows category view.

Select **Workflows** to view a list of available workflows.





Viewing Workflows

Only a user who is able to access the Workflows category view can manage workflows. Other users can view the list of tasks from the workflow task drop-down list that is accessible from the project or rule flow toolbar. If a user is the actual owner of a task, or assigned as a potential owner of a task, they can view the workflow diagram and tasks that in the My Tasks category view. Workflows are associated with a project or rule flow at the version-level.

From the Workflows category view, you can perform the following actions:

- set mappings
- terminate a workflow process

Note: You need to migrate workflows only if you have migrated content from SAS Model Manager 12.3. For more information, see "Migrate Workflows" in *SAS Decision Manager: Administrator's Guide*.

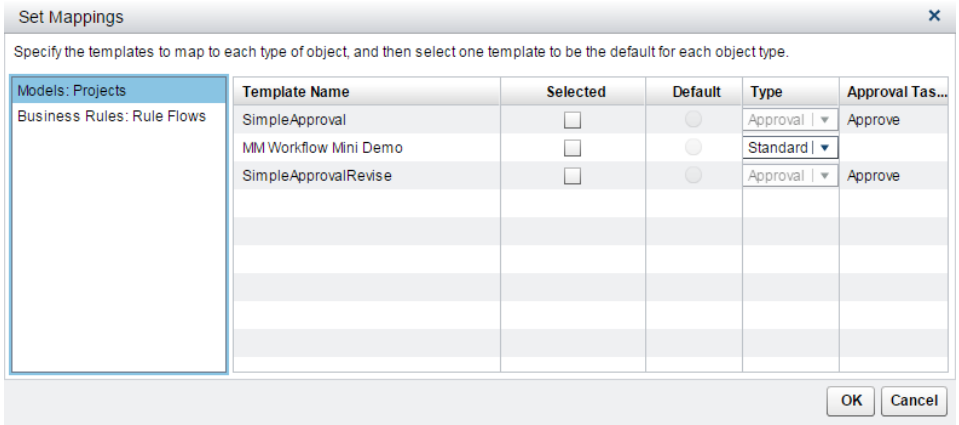
To view detailed information for a workflow, double-click a workflow name. The list of tasks, the task status, and who the task is claimed by are displayed. You can then view the properties and participants that are associated with a task by selecting a task. The workflow diagram is also displayed with the current status of the workflow and its tasks.

For more information, see "Working with Workflow Participants" on page 314.

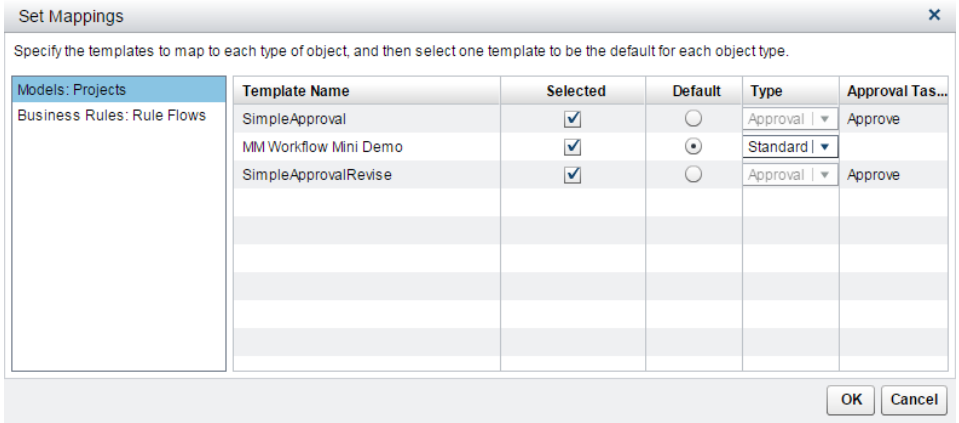
Set Mappings

There are two different types of workflow templates that can be configured for use with SAS Decision Manager. Workflow templates that contain tasks that are configured with an approval status are considered an approval workflow. Workflow templates that do not contain tasks with an approval status are considered a standard workflow. The rule flows object can be associated with only an approval type. After you define your workflow template, save, and activate it using SAS Workflow Studio. You must specify the templates to map to each type of object. This enables you to start a new workflow using one of the templates that are associated with the specific object.

1. Select **Actions** ⇨ **Set Mappings**. The Set Mappings window appears.



2. Select an object and then select one or more templates to map to the object.
3. Select a type for each template. The types of templates that are available are **Approval** and **Standard**.
4. Select the default template for the object.



5. Click **OK**.

Working with Workflow Participants

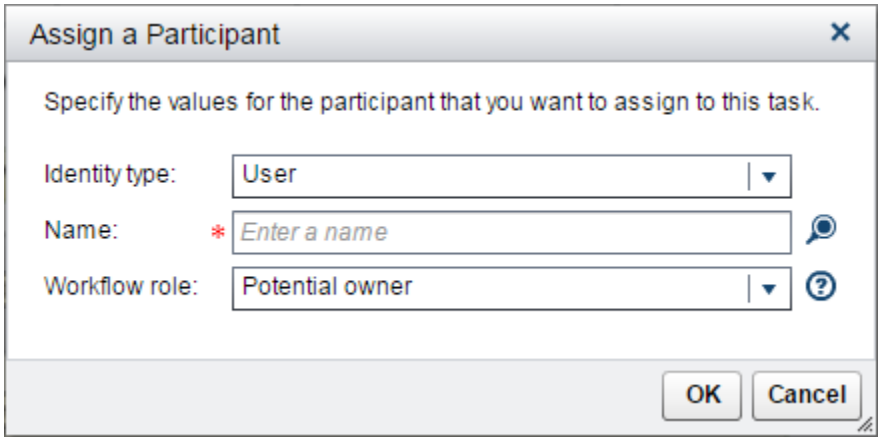
From the Workflow details view you can access the properties and participants that are associated with a task by selecting a task. If you are a user that is associated with the workflow role of business administrator, you can assign or remove participants, and release tasks that have been claimed by another user.

Assign Participants to Tasks

Default participants might have been assigned already to tasks when a workflow definition was created.

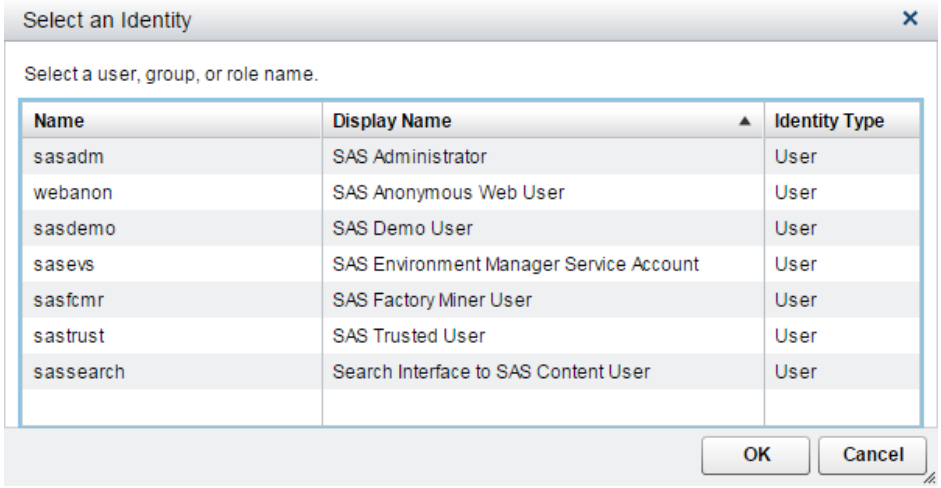
To assign an additional participant to a task:

1. From the Workflows category view, double-click a workflow. The Workflow details view is displayed.
2. Select a task, and then click + in the Participants pane. The Assign a Participant window appears.



3. Select an identity type.
4. Enter part of the user, group, or role name, and click .

Note: If you do not enter part of the name, all of the names for the selected identity type are displayed.



Select a name and click **OK**.

5. Select a workflow role for the participant.

   Here are the workflow roles that you can assign to participants for a workflow task:

   - **Business administrator**: a participant who can influence the progress of a task by actions such as assigning a task, or releasing the task claimed by another user.
   - **Potential owner**: a participant who can claim a task in a workflow process and who becomes the actual owner of a task.

6. Click **OK**. The new participant is added to the list in the Participants pane.

Remove Participants from a Task

To remove a participant from a task:

1. From the Workflows category view, double-click a workflow name.
2. Select a task, and then select a participant from the Participants pane.

   Note: You cannot remove a participant who is associated with the workflow roles of business administrator or actual owner.

3. Click . A message is displayed asking if you are sure that you want to remove the participant from the task.
4. Click **Yes**. The user is removed from the list in the Participants pane.

Release a Task

An authorized user with the capability to access the Workflows category view can release a task that has been claimed by a workflow participant. The name of the actual owner is displayed in the Participants pane.

To release a task:

1. In the Workflows category view, double-click a workflow name. The Workflow details view is displayed.

2. Select a task name, and click . The **Claimed By** value for the selected task is cleared.

Edit Task Properties

A task can contain properties. Properties that are editable display a triangular icon in the bottom right corner of the property value in the data grid.

To edit the properties for a task:

1. From the Workflows category view, open a workflow, and select a task. The properties that are associated with the task are displayed to the right in the Properties pane.
2. Click on the property value, and then enter a value or change the existing value.
3. To save the changes to the properties, click .

   If you do not want to save the changes to the properties, click .

Terminate a Workflow

When you terminate a workflow process, all tasks that have not yet been completed are changed to a state of Terminated. After you terminate a workflow process, it cannot be restarted. However, you can start a new workflow for the same version.

To terminate a workflow:

1. From the Workflows category view, select a workflow name and click .
2. Click **Yes** to terminate the selected workflow.

Part 8

Creating and Publishing Decisions

Chapter 26

Creating and Publishing Decisions

Introduction to Decision Builder

Decision Builder enables you to combine analytical models, rule flows, 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. You can automate the decision-making process. Automating decisions with Decision Builder provides a streamlined mechanism for controlling and monitoring the rules and processes used by your organization.

Accessing Decision Builder

You can access Decision Builder in two ways:

- If you are signed in to SAS Visual Analytics home page, click ≡ in the application banner and select **Decision Builder**.
- In the address bar of your web browser, enter the URL for Decision Builder and press Enter. The default URL is http://*host_name*:*port*/SASDecisionManager/DecisionBuilder. In the Sign In window, enter your user ID and password, and click **Sign In**.

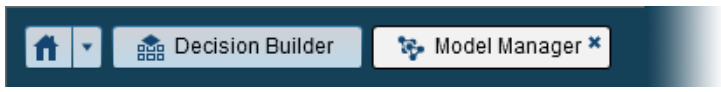
  Note: Your user ID might be case sensitive, depending on the operating system that is used to host the application server. Your password is case sensitive.

To sign out of Decision Builder, click ▾ in the banner, and select **Sign Out**.

Switching between Applications

If SAS Visual Analytics Hub is installed at your site, you can use the side menu to switch from Decision Builder to other applications. Click ≡ in the application banner and select an application. Select **Home** to open SAS Visual Analytics Hub. You can use the Settings window to customize the content of the side menu.

In other applications such as SAS Business Rules Manager or SAS Model Manager, buttons are added to the banner. You can click these buttons to switch between applications, or click ▾ to open a menu of applications. Click ⌂ to open SAS Visual Analytics Hub.



Managing Settings and Notifications in Decision Builder

Managing Application Settings

Use the Settings window to edit user preferences or to customize accessibility settings for all SAS web applications. Changing these settings does not impact other users. To access these settings, click your name in the application bar and select **Settings**.

General Settings

By default, the user locale is set to match your browser locale. The user locale specifies the geographic region whose language and conventions are used in the applications. If you do not want your user locale and browser locale to be the same, change the value of **User locale**.

If you want to display messages that you previously chose not to display, click **Reset Messages**. By default, all warning and information messages are displayed.

Accessibility Settings

Several settings in the **Accessibility** section can assist people who rely on assistive technologies.

- Select **Enable sounds** to hear audio indicators for events that occur within the user interface.
- Select **Enable visual effects** to show visual effects that indicate state changes. For example, when this setting is selected, you see a subtle movement in the user interface if you delete an item.
- Select **Invert application colors** to change the application interface colors (for example, a black-on-white display). Changing the display colors can make the user interface easier to see for users with sensitivity to bright colors. You can also use the Ctrl+` (Ctrl+back quote) keyboard shortcut to toggle the application colors.
- Select **Display tooltips when using the keyboard to navigate** to enable keyboard users to access tooltips. When this option is selected, putting keyboard focus on a control also displays the tooltip on the screen. By default, this option is not selected, so a mouse is required to see a tooltip. You can also select the location in the browser window to display the tooltip. By default, the tooltip is displayed in the bottom right corner of the browser window.
- The focus indicator is an outline that indicates which user interface component is active. You can make the focus indicator easier to see by selecting **Customize the focus indicator settings** and adjusting the color, thickness, and opacity.

Managing Notifications

Notifications are messages that are generated by events. Click to see notifications that were sent by the system. You can delete notifications, but you cannot turn them off or customize them.

Create a New Decision

1. Click . The New Decision window appears.
2. Enter a name for the decision. Names are limited to 100 characters and must be unique. The first character must be an alphabetic character. Subsequent characters can be a number or any character except forward slash (/), backslash (\), left brace ({), right brace (}), left bracket ([), right bracket (]), colon (:), and question mark (?).
3. (Optional) Enter a description. Descriptions are limited to 1024 characters.
4. Select the vocabulary for the decision, and click **Save**. Decision Builder opens the new decision.

   TIP Click the **Properties** tab to display the properties of a decision. The properties include the date on which the decision was created and the user ID that created it, in addition to the name, description, and vocabulary.
5. Add content to the decision. A decision can contain any combination of rule flows, models, and conditions. See "Editing a Decision" on page 322 for more information.
6. Click .

TIP Click at any time to return to the list of all decisions.

Open Existing Decisions

To open a single decision, select the decision.

To open multiple decisions:

1. Select the check boxes for the decisions that you want to open.
2. Click .

The opened items icon in the upper right of the window shows how many decisions are open. For example, 2 indicates that two decisions are open.

To switch between open decisions, click 2 and select the decision that you want to display.

TIP Click at any time to return to the list of all decisions.

Editing a Decision

A decision can contain rule flows, models, and conditions, which consist of IF clauses and ELSE clauses. The IF and ELSE clauses of a condition can contain additional rule flows, models, and conditions.

Control the Order of Elements in a Decision

When you add a new item to a decision, the decision editor adds the item to the decision based on whether another item is currently selected and, if so, what the selected item is.

- If no item is currently selected, the decision editor adds the new item at the beginning of the decision.
- If the currently selected item is a rule flow or a model, the decision editor adds the new item immediately below it.
- If the currently selected item is a condition, placement depends on whether the condition is collapsed or expanded.
  - If the condition is collapsed, the decision editor adds the new item immediately below it.
  - If the condition is expanded and the currently selected item is the IF or ELSE clause, the decision editor adds the new item to the content of the clause.

To move an item up or down, including into and out of a condition, select the item and click ⬆ or ⬇. You can also move the selected item using Shift + ↑ and Shift + ↓ on the keyboard.

Add a Rule Flow

1. Select an item according to where you want the new rule flow to be added. See "Control the Order of Elements in a Decision" on page 323 for more information.
2. Click **Add Rule Flow**. The Open window appears.
3. Select the rule flow and click **Open**. The Select Rule Flow Version window appears.

   Note: You cannot add complex rule flows to a decision.
4. Select the version of the rule flow that you want to add to the decision, and click **OK**.

Add a Model

1. Select an item according to where you want the new model to be added. See "Control the Order of Elements in a Decision" on page 323 for more information.
2. Click **Add Model**. The Open window appears.

   Note: Only models with a **Score Type Code** of **DS2** or **Data step** are supported in Decision Builder.
3. Select the model and click **Open**.

Add a Condition

1. Select an item according to where you want the new condition to be added. See "Control the Order of Elements in a Decision" on page 323 for more information.
2. Click **Add Condition**.
3. Enter the expression for the IF clause.

a. In the first field, select a variable for the left side of the expression.

b. Select the operator.

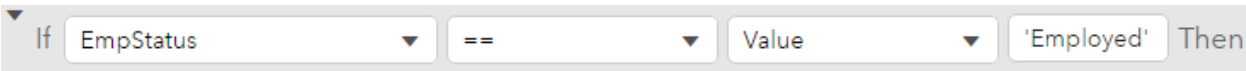
c. For the right side of the expression, select **Value** or **Variable**.

- To compare the first variable to a constant, select **Value**.
- To compare the first variable to another variable, select **Variable**.

d. In the last field, either enter the constant or select the comparison variable.

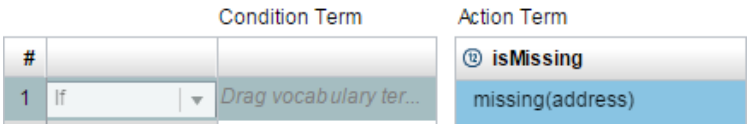
4. Add content to the IF and ELSE clauses. You can leave an IF or ELSE clause empty. A clause can contain any combination of rule flows, models, and conditions.

For example, the expression **`EmpStatus == 'Employed'`** appears in Decision Builder as shown in the following figure.



The expression **`Credit_Score < Score_Minimum`** appears in Decision Builder as shown in the following figure.

TIP You cannot use a period (.) to check for missing values. You can use the MISSING function instead. See "MISSING Function" in *SAS Functions and CALL Routines: Reference* for more information. For example, the following rule checks whether the value for the variable address is missing.



Exclude Variables from Input or Output

Decision Builder expects all input variables in a decision to be mapped to columns in an input table. All output variables are included in the output data sets by default. You can exclude variables from the mapping process and from the output data sets.

To exclude variables:

1. Click the **Variables** tab.

   TIP You can use the radio buttons and the **Filter** field to limit the display to variables of certain object types or to a specific model or rule flow.

2. (Optional) For each variable that you do not want to map to an input column, clear the **Input** check box.

3. (Optional) For each variable that you do not want included in the output, clear the **Output** check box. To exclude all variables from the output, clear the check box next to the Output column heading.

   TIP To use a variable only while a decision is executing, clear both the **Input** and **Output** check boxes.

Map Variables to Vocabulary Terms

The variables referenced in a decision must be mapped to terms in the vocabulary that the decision uses. Decision Builder automatically maps a variable to an existing term if the variable and term have the same name and the same data type. If a matching term does not exist, Decision Builder creates a matching term and maps the variable to the term. To change the default variable mappings:

1. Select the rule flow or model for which you want to map variables.
2. Click . The Properties window appears.
3. On the **Variables** tab, under the **Maps To** column, select or change the term for each variable as needed.
4. Click **Close**.

Change a Rule Flow or Model Used in a Decision

1. Select the rule flow or model that you want to change.
2. Click . The Open window appears.
3. Select the rule flow or model that you want to use in the decision, and click **Open**.

Delete an Item in a Decision

To delete a rule flow, model, or condition, click for that object.

If you delete a condition, the contents of the condition (both clauses) are retained in the same position that the condition occupied.

Note: You cannot delete empty ELSE clauses.

Close Decisions

To close a single decision, click **Close**.

To close all open decisions, click and select **Close All**.

Duplicate a Decision

1. Select the check box for the decision that you want to duplicate.
2. Click . The Duplicate Decision window appears.
3. Enter a name for the new copy, and click **OK**.

Delete Decisions

Note: You cannot delete open decisions. If a decision that you want to delete is open, close it first.

To delete decisions:

1. Select the check boxes for the decisions that you want to delete.
2. Click 🗑. The application asks you to confirm that you want to delete the selected items.
3. Click **Yes**.

Testing a Decision

Input Data for Decision Tests

Decision Builder expects the input data for the decision test to already exist and to be defined as a data source. See "About Managing Data Tables" on page 47 for information about defining data sources. Your user ID must have permission to access the data.

Create and Run a New Decision Test

1. Open the decision that you want to test.

   Note: If you make any changes to the decision, click 💾. You cannot create a test for a decision that contains unsaved changes.
2. Click **Test**. The New Test window appears.
3. Enter a name for the test. Names are limited to 100 characters and must be unique. The first character must be alphabetic. Subsequent characters can be numbers, alphabetic characters, or underscores (_).
4. (Optional) Enter a description. Descriptions are limited to 1024 characters.
5. Select the data source for the test, and click **Open**.

   The application automatically maps input variables in the decision to columns in the test data source if the variable and column have the same name and the same data type. For other variables, you must manually map the variables to columns in the input data source.
6. On the **Variables** tab, under **Table Column**, select the table column for each variable as necessary.

   Note: The application automatically maps variables only when the decision is created. If input or output variables are added later to the decision, you can reselect the input data source to automatically map the additional variables.
7. (Optional) On the **Preprocessing Code** tab, enter any SAS code, such as initialization code or setup code, that you want to run before the decision is run.

TIP You can use the &DCM_LIBS_OPTION macro variable to specify librefs for libraries that your decision needs to access when it is run. By default, SAS Decision Manager restricts the libraries to the minimum set of required libraries, lib4Rst and Work. If you specify this variable, you must include lib4Rst and Work in your specification. The minimum specification is as follows:

```
%let dcm_libs_option=lib4Rst work;
```

You can add additional librefs to this statement if needed.

8. Click to save the decision test.
9. Click **Run** to run the test. Decision Builder runs the test and, if the test completely successfully, displays the results. See "View Test Results" on page 327 for more information.
10. Click **Close** to close the test.

Open an Existing Decision Test

1. Open the decision.
2. Click and select **View Tests**. The All Tests window appears.
3. Select the test that you want to open.

Run an Existing Test

To run an existing test, open the test, and click **Run**.

If you make any changes to the test, click to save the test. You cannot run a test with unsaved changes.

Decision Builder runs the test and, if the test completely successfully, displays the results. See "View Test Results" for more information.

View Test Results

When you run a test, Decision Builder automatically displays the test results. It also adds a link on the main page for the test to the test results. To view the results that were generated the last time a test was run, open the test, and click **View the results**. Decision Builder displays

- the output of the decision

  Note: Decision test output tables are limited to 1000 rows.
- the SAS code that was generated and run when the test was submitted
- the SAS log
- information about the output table such as its name and location
- information about the test itself, including the date on which the test was last run

Delete a Decision Test

1. Open the decision.
2. Click ⋮ and select **View Tests**. The All Tests window appears.
3. Select the test that you want to delete.
4. Click 🗑.

Publish a Decision

Publishing is the process of making the decision available to be run by other applications. You can choose to publish a decision as a SAS Micro Analytic Service module or as a Decision Logic metadata object. The method that you choose depends on how other applications will run the decision. To publish a decision:

1. Open the decision.
2. Click **Publish**. The Publish Decision window appears.
3. Select the method that you want to use to publish the decision.

   Use the SAS Micro Analytic Service to process transactions
   creates a REST endpoint in the SAS Micro Analytics Service instance that is running within your current SAS deployment. Any application that can call REST services can use the module to process web service transactions. See *SAS Micro Analytic Service: Programming and Administration Guide* for more information.

   Use a Decision Logic object in batch applications
   writes a Decision Logic metadata object to the SAS Metadata Repository. Batch applications such as SAS Data Integration Studio jobs can then use the object. For information about metadata objects, see "SAS Metadata Server" in *SAS Intelligence Platform: Overview*.

4. Select whether you want to publish a new module or object, or replace a previously published one. If you publish the decision as a Decision Logic object, the **Location** field is enabled.

   Note: If you replace a previously published module, Decision Builder replaces the last module that was published for the decision. If you replace a previously published Decision Logic object, Decision Builder replaces the last object that was published for the decision in the location that you select in Step 5.

5. (Optional) Select the location in metadata where you want to publish the Decision Logic object.
6. Click **Publish**. Decision Builder notifies you whether the decision was successfully published, and if so, provides additional information about the published object or module.
7. Click **Close** to close the decision.

See "Deploy the Decision in SAS Data Integration Studio" on page 45 for an example of how you can run the published decision in a SAS Data Integration Studio job.

View the Publish History for a Decision

1. Open the decision.
2. Click and select **View Publish History**. The information for each object or module includes the date on which it was published, the method used to publish it, and the user ID that published it.

   Decisions that were published as Decision Logic objects display **Batch** as the publish method. Decisions that were published as SAS Micro Analytic Service modules display **Web service** as the publish method.
3. Click **Close** to close the **Publish History** window.

Writing to Output Tables from SAS Data Integration Studio Jobs

When you run a published decision in a SAS Data Integration Studio job, the following naming guidelines apply:

- You can write to output tables by using two-level or three-level names. Two-level names follow the format Libref.Tablename. Three-level names follow the format Libref.Schema.Tablename.
- To use three-level names, the schema name for the output library must be defined in SAS Management Console with the same name and case (uppercase or lowercase) that appears in database user interface.
- For Oracle, the output table name is case-insensitive.
- For Postgres, the output table name must be all lowercase. Also, for Postgres on UNIX, you must set the PRESERVE_COL_NAMES and PRESERVE_TAB_NAMES options for the output library in SAS Management Console to **`Yes`**.

For example, if your schema is XRTEST, then you might access the output library in Oracle with the following LIBNAME statement in the preprocessing code for your job:

```
LIBNAME olib ORACLE PATH=edmdb SCHEMA=XRTEST USER=user_ID PASSWORD=password;
```

For Postgres, the LIBNAME statement might appear as follows:

```
LIBNAME plib POSTGRES PRESERVE_COL_NAMES=YES PRESERVE_TAB_NAMES=YES
DATABASE=dcmdb SERVER="server" PORT=10482 SCHEMA=XRTEST USER=dcmdb
PASSWORD=password;
```

Viewing Lineage Information for a Decision

About Lineage Information

The lineage viewer is provided by SAS Lineage. The relationship information displayed by SAS Lineage is taken from the Relationship database that is a part of the Web Infrastructure Platform Data Server. SAS Lineage can display most types of SAS metadata. This data includes models, rule flows, and data objects, including columns, tables, external files, stored processes, and more.

SAS Lineage displays three types of diagrams:

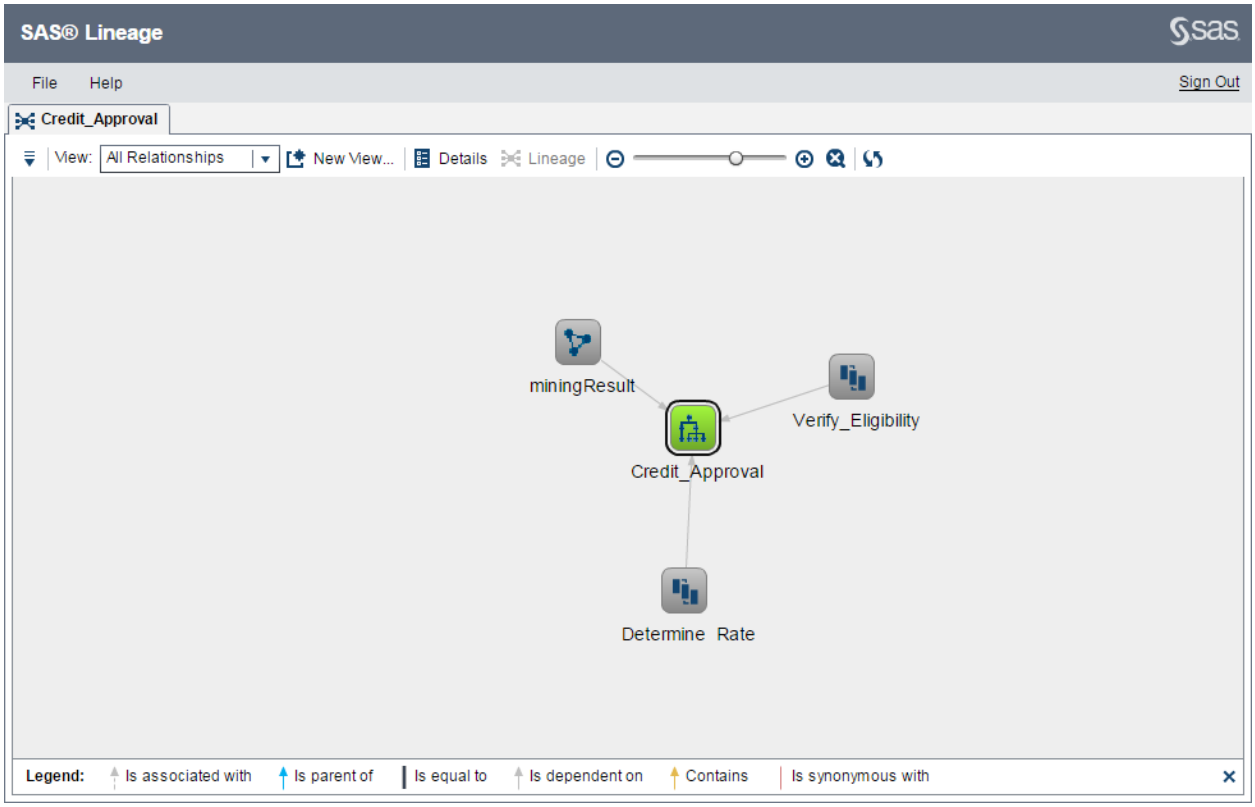
- a network diagram that displays all relationships
- a dependency diagram that displays governance information
- a dependency diagram that displays parent and child relationships

View Lineage Information for a Decision

1. Open the decision.
2. Click and select **View Lineage**. If you are not already signed in to SAS Lineage, it prompts you to sign in.
3. Enter your user ID and password, and click **SIGN IN**. SAS Lineage displays the All Relationships diagram for the decision.

For additional information about SAS Lineage, select **Help** or see *SAS Lineage: User's Guide*.

The following image shows the All Relationships diagram for a simple decision that uses one model and two rule flows.



Part 9

Appendixes

Appendix 1

Rule-Fired and Test Information Tables

Overview

When you run a rule flow by using either the %BRM_RULE_FLOW macro or the Business Rules transformation in SAS Data Integration Studio, three tables are generated. These tables contain rule-fired information and information about the execution of the rule flow.

Table	Contents	Name Generated By SAS Data Integration Studio
Test information	A single record that contains aggregate information about the execution of the rule flow.	DCM_DEPLOYMENT_EXECUTION
Rule-fired details	One row for each time that a rule evaluates to true. There might be multiple entries for the same rule, but each entry has different values for the _recordCorrelationKey and RULE-ACTION_FIRE_ID columns.	DCM_RULE_ACTION_FIRE
Rule-fired summary	A summary of how many times each rule fired.	DCM_RULE_ACTION_FIRE

When the %BRM_RULE_FLOW macro is run outside of SAS Data Integration Studio, the names of the tables are controlled by the mapping file. See "%BRM_RULE_FLOW" in *SAS Decision Manager: Administrator's Guide* for information about the macro and the mapping file.

Columns in the Test Information Table

DEPLMT_EXECUTION_ID
: the identification string of the specific instance of the rule flow that was executed. Each time a rule flow executes, a different universally unique identifier (UUID) is generated for the specific instance of the rule flow. You can use this UUID to associate the records in the test information table with the records in the rules-fired details table.

DEPLMT_SK
: an internal surrogate key for the publish information for the rule flow. The publish information includes who published the rule flow, the version number that was published, and the location to which the rule flow was published.This column is also included in the rules-fired details table. You can use this column to join the two tables.

DEPLMT_NM
: the name of the metadata object that was executed.

TRANSACTION_MODE_CD
: always set to **DIS**.

Note: This column has been deprecated.

RECORDS_PROCESSED_NO
: the number of records that were processed by the rule flow.

TEST_FLAG
: indicates whether the rule flow was run in the SAS Decision Manager test interface.

START_DTTM
: the date and time at which the rule flow started executing.

END_DTTM
: the date and time at which the rule flow finished executing.

Columns in the Rule-Fired Details Table

RULE_ACTION_FIRE_ID
: the UUID that is generated for each rule each time it is executed. Each time a rule executes, a different UUID is generated for the specific instance of the rule.

RULE_SET_SK
: the identification number of the rule set.

RULE_SET_NM
: the name of the rule set.

RULE_SK
: the identification number of the rule.

RULE_NM
: the name of the rule.

DEPLMT_SK
an internal surrogate key for the publish information for the rule flow. This column is also included in the test information table. You can use this column to join the two tables.

RULE_FLOW_SK
the identification number of the rule flow.

RULE_FLOW_NM
the name of the rule flow.

RULE_FIRE_DTTM
the date and time that the rule was run.

DEPLMT_EXECUTION_ID
the identification string of the specific instance of the rule flow that was executed. Each time a rule flow executes, a different UUID is generated for the specific instance of the rule flow. You can use this UUID to associate the records in the rules-fired details table with the records in the test information table on page 336.

ENTITY_PRIMARY_KEY
the value of the term that was specified with the &BRMPrimaryEntityKey macro variable in preprocessing code.

Note: This column has been deprecated. Use the _recordCorrelationKey column instead.

TRANSACTION_DTTM
the value of the term that was specified with the &BRMTransactionDTTM macro variable in preprocessing code.

Note: This column has been deprecated. Use the _recordCorrelationKey column instead.

_recordCorrelationKey
a UUID that enables you to associate records in the rules-fired details table (DCM_RULE_ACTION_FIRE) with records in the output results table. This column is also added to the output results table, so you can use this key to join the rules-fired details table and the output results table. Joining the tables enables you to enrich the information in the rules-fired details table with transaction times, composite keys, or other information.

Columns in the Rules-Fired Summary Table

RULE_SK
the identification number of the rule.

RULE_NM
the name of the rule.

RULE_SET_SK
the identification number of the rule set.

RULE_SET_NM
the name of the rule set.

RULE_FLOW_SK
the identification number of the rule flow.

RULE_FLOW_NM
: the name of the rule flow.

ruleFiredCount
: the number of times that the rule specified by the RULE_SK field executes for all of the input records that were processed.

Appendix 2

Model Templates

What Is a Model Template?

Models that you import into SAS Decision Manager are associated with a specific model template. A model template has properties and component files that define a type of model. SAS Decision Manager processes four types of models: analytical, classification, prediction, and segmentation. You can create your own model template if your model requires files other than those named in the SAS Decision Manager templates.

A model template is an XML file that has three sections. The **General** section names and describes the model template. The **Properties** section provides properties to name the model algorithm, the modeler, and a model label. The **Files** section contains the component files that can be used in the template for that model function type. You associate your component file with the appropriate model template component file. Your component file filenames do not need to be the same name as the filenames in the model template.

Model templates provide you with a way to define metadata about your own model. Most users do not need to write model templates because SAS Decision Manager delivers a list of model templates that handle SAS Enterprise Miner models as well as analytical, prediction, classification, and segmentation models. However, you can write your own model templates if the model templates that are provided do not satisfy your requirements. For more information, see "Creating and Managing Templates" on page 138.

Model Types

SAS Decision Manager provides model templates for analytical, classification, prediction, and segmentation models.

Model Type	Description
Analytical	The **Analytical** model template is the most generic template that is designed for models whose model function does not fall in the prediction, classification, and segmentation category.

Model Type	Description
Classification	You use the **Classification** model template if your model is a prediction model that has a categorical, ordinal, or binary target, or if your model is a LOGISTIC procedure regression model. Examples of classification models are models that might classify a loan applicant as Approved or Not Approved, or models that might assess a potential customer's risk of default as Low, Medium, or High.
Prediction	The **Prediction** model template is used for predictive models. Predictive models declare in advance the outcome of an interval target. A model that assigns a numeric credit score to an applicant is an example of a prediction model.
Segmentation	The **Segmentation** model template is used for segmentation or cluster models that are written in SAS code. Segmentation models are unsupervised models that have no target variable. A segmentation or cluster model is designed to identify and form segments, or clusters, of individuals or observations that share some affinity for an attribute of interest. The output from a segmentation model is a set of cluster IDs. R models cannot have segmentation model function.

Model Template Component Files

Here is a list of the component files that are associated with the model templates:

Filename	Analytical	Classification	Prediction	Segmentation
IGN_STATS.csv on page 342	—	✔	—	—
EMPublishScore.sas on page 342	—	✔	—	—
Scorecard_GainsTable.csv on page 342	—	✔	—	—
score.sas on page 342	✔	✔	✔	✔
modelinput.sas7bdat on page 342	✔	✔	✔	✔
modeloutput.sas7bdat on page 342	✔	✔	✔	✔
target.sas7bdat on page 343	—	✔	✔	—
inputvar.xml on page 343	✔	✔	✔	✔
outputvar.xml on page 344	✔	✔	✔	✔

Filename	Analytical	Classification	Prediction	Segmentation
targetvar.xml on page 344	—	✔	✔	—
smmpostcode.sas on page 345	✔	✔	✔	✔
trainingvariables.csv on page 345	—	✔	—	—
training.sas on page 345	✔	✔	✔	✔
training.log on page 345	✔	✔	✔	✔
training.lst on page 346	✔	✔	✔	✔
outest.sas7bdat on page 346	✔	✔	✔	—
outmodel.sas7bdat on page 346	✔	✔	✔	—
output.spk on page 346	✔	✔	✔	✔
miningResult.spk on page 346	✔	✔	✔	—
layout.xml on page 346	—	✔	✔	—
format.sas7bcat on page 346	✔	✔	✔	✔
dataprep.sas on page 346	✔	✔	✔	✔
batch.sas on page 346	✔	✔	✔	✔
pmml.xml on page 346	✔	✔	✔	—
training.r on page 346	✔	✔	✔	—
outmodel.rda on page 346	✔	✔	✔	—
score.r on page 346	✔	✔	✔	—
fitstats.xml on page 346	—	✔	✔	—
HPDMForest_VARIMPORT.csv on page 346	—	✔	✔	—

Filename	Analytical	Classification	Prediction	Segmentation
HPDMForest_ITERATION.csv on page 347	—	✔	✔	—
outmdlfile.bin on page 347	—	✔	✔	—

IGN_STATS.csv

The value of IGN_STAT.csv is the name of a file whose values are separated by commas, and whose values are bin definitions for input variables. This is a component file that is generated by SAS Enterprise Miner for a scorecard model and is not needed for SAS code models.

EMPublishScore.sas

The value of EMPublishScore.sas is the name of a SAS code file that is used to change input variables into bins and is a component of a SAS Enterprise Miner scorecard model. This file is needed to define a performance task. This file is not needed for SAS code models.

Scorecard_GainsTable.csv

This file includes the bin score definitions and is not used in reporting by SAS Decision Manager. The file's content can be viewed by users.

score.sas

The value of score.sas is the name of a filename for the SAS score code for the model.

For R models, this file transforms a scoring data set to an R data frame.

The score.sas file is DATA step score code and is used as input by the SAS Scoring Accelerator when publishing a model to a database. When you are using the scoring function publish method, some SAS language elements and syntax are not supported when you create or modify your score code. Only the SAS language elements and syntax that are required to run critical data transformations and model scoring functions are available. If you use a statement or function that is not supported, an error occurs and your model is not published to the database. For more information, see "Considerations When Creating or Modifying DATA Step Score Code" in *SAS In-Database Products: User's Guide*.

modelinput.sas7bdat

The value of modelinput.sas7bdat is the name of a sample data set that is used to create an inputvar.xml file for the model if one does not exist. When no inputvar.xml file exists for the model, SAS Decision Manager creates the inputvar.xml file using the variable name and attributes in the modelinput.sas7bdat file. Observation values are not used. Therefore, the sample data set can have no observations or it can have any number of observations. If an inputvar.xml is specified in the model template, modelinput.sas7bdat is ignored.

When you import a SAS code model, the data set that you used to test your score code can be used as the value for the modelinput.sas7bdat file.

Note: If the same variables appear in your modelinput.sas7bdat file and your modeloutput.sas7bdat file, when you import the model, SAS Decision Manager removes the duplicate variables in the outputvar.xml file.

modeloutput.sas7bdat

The value of modeloutput.sas7bdat is the name of a sample data set that is used to create an outputvar.xml file for the model if one does not exist. When no outputvar.xml file exists for the model, SAS Decision Manager creates the

outputvar.xml file using the variable name and attributes in the modeloutput.sas7bdat file. Observation values are not used. Therefore, the sample data set can have no observations or it can have any number of observations. If an outputvar.xml is specified in the model template, modeloutput.sas7bdat is ignored.

You can create a modeloutput.sas7bdat file by running the score.sas file against the modelinput.sas7bdat file.

target.sas7bdat

The value of target.sas7bdat is the name of a sample data set that is used to create a targetvar.xml file for the model if one does not exist. When no targetvar.xml file exists for the model, SAS Decision Manager creates the targetvar.xml file using the variable name and attributes in the target.sas7bdat file. Data set values are not used. Therefore, the sample data set can have no observations or it can have any number of observations. If a targetvar.xml file is specified in the model template, target.sas7bdat is ignored.

You can create a target.sas7bdat file by creating a data set that keeps only the target variables that are taken from the training data set, as in this example:

```
data mydir.target;
    set mydir.myModelTraining (obs-1);
    keep P_BAD;
run;
```

inputvar.xml

The value of inputvar.xml is the name of an XML file that defines the model input variables. When your model template includes a file for modelinput.sas7bdat, SAS Decision Manager creates the model inputvar.xml file. Otherwise, you must create the XML file.

The following XML file is a sample inputvar.xml file that has one variable, CLAGE. You can use this model to create an inputvar.xml file that contains a VARIABLE element for each model input variable.

```
<?xml version="1.0" encoding="utf-8"?>
<TABLE>
   <VARIABLE>
         <NAME>CLAGE</NAME>
         <TYPE>N</TYPE>
         <LENGTH>8</LENGTH>
         <LABEL Missing=""/>
         <FORMAT Missing=""/>
         <LEVEL>INTERVAL</LEVEL>
         <ROLE>INPUT</ROLE>
        </VARIABLE>
   </TABLE>
```

NAME
: specifies the variable name.

TYPE
: specifies the variable type. Valid values are N for numeric variables and C for character variables.

LENGTH
: specifies the length of the variable.

LABEL Missing=""
: specifies the character to use for missing values. The default character is a blank space.

FORMAT Missing=""
specifies a SAS format to format the variable.

LEVEL
specifies either NOMINAL, ORDINAL, INTERVAL, or BINARY.

ROLE
specifies INPUT for input variables.

outputvar.xml

The value of outputvar.xml is the name of an XML file that defines the model output variables. When your model template includes a file for modeloutput.sas7bdat, SAS Decision Manager creates the model outputvar.xml file. Otherwise, you must create the XML file.

The following XML file is a sample outputvar.xml file that has one variable, I_BAD. You can use this model to create an outputvar.xml file that contains a VARIABLE element for each model output variable.

```
<?xml version="1.0" encoding="utf-8"?>
<TABLE>
   <VARIABLE>
         <NAME>I_BAD</NAME>
         <TYPE>C</TYPE>
         <LENGTH>12</LENGTH>
         <LABEL>Into: BAD</LABEL>
         <FORMAT Missing=""/>
         <LEVEL>NOMINAL</LEVEL>
         <ROLE>CLASSIFICATION</ROLE>
   </VARIABLE>
 </TABLE>
```

NAME
specifies the variable name.

TYPE
specifies the variable type. Valid values are N for numeric variables and C for character variables.

LENGTH
specifies the length of the variable.

LABEL Missing=""
specifies a label for the output variable.

FORMAT Missing=""
specifies a SAS format to format the variable.

LEVEL
specifies either NOMINAL, ORDINAL, INTERVAL, or BINARY.

ROLE
specifies the type of model output. Valid values are CLASSIFICATION, PREDICT, SEGMENT, and ASSESS.

targetvar.xml

The value of targetvar.xml is the name of an XML file that defines the model target variables. When your model template includes a file for target.sas7bdat, SAS Decision Manager creates the targetvar.xml file. Otherwise, you must create the XML file.

The following XML file is a sample targetvar.xml file that has one variable, I_BAD. You can use this model to create an outputvar.xml file that contains a VARIABLE element for each model output variable.

```
<?xml version="1.0" encoding="utf-8"?>
<TABLE>
   <VARIABLE>
         <NAME>BAD</NAME>
         <TYPE>N</TYPE>
         <LENGTH>8</LENGTH>
         <LABEL>Missing=””/>
         <FORMAT Missing=""/>
         <LEVEL>BINARY</LEVEL>
         <ROLE>TARGET</ROLE>
  </VARIABLE>
 </TABLE>
```

NAME
specifies the variable name.

TYPE
specifies the variable type. Valid values are N for numeric variables and C for character variables.

LENGTH
specifies the length of the variable.

LABEL Missing=""
specifies a label for the target variable.

FORMAT Missing=""
specifies a SAS format to format the variable.

LEVEL
specifies either NOMINAL, ORDINAL, INTERVAL, or BINARY.

ROLE
specifies TARGET for the role of the variable..

smmpostcode.sas
SAS Decision Manager creates this file to document the mapping that the user specified between the model variables and the project variables.

trainingvariables.csv
This optional file contains a list of the training variables.

training.sas
This file is the optional SAS code that was used to train the model that you are importing. If at some time, SAS Decision Manager reporting utilities detect a shift in the distribution of model input data values or a drift in the model's predictive capabilities, the training.sas code can be used to retrain the model on the newer data. If it is not available at import time, the training.sas code can be added at a later point using the Add Local Files feature.

training.log
This file is the optional log file that was produced when the model that you are importing was trained. The information in the optional SAS training log can be helpful if the model must be retrained in the future.

training.lst
This file is the optional text output that is produced when the training.sas code is run. The information in the optional SAS training.lst table can be helpful if the model must be retrained in the future.

outest.sas7bdat
This data set contains output estimate parameters that are produced by a few SAS procedures, including the LOGISTIC procedure.

outmodel.sas7bdat
This data set contains output data that is produced by a few SAS procedures, including the LOGISTIC procedure and the ARBORETUM procedure. It contains complete information for later scoring by the same SAS procedure using the SCORE statement.

output.spk
This file is the SAS package file that contains the SPK collection of model component files.

miningresult.spk
This is a SAS package file that stores detailed information about SAS Enterprise Miner nodes in the flow from which the model is created and the detailed information for SAS/STAT item store models.

layout.xml
This optional file contains information about the SAS Enterprise Miner diagram topology.

format.sas7bcat
This file is the optional SAS formats catalog file that contains the user-defined formats for their training data. If the model that you are importing does not use a user-defined format, then you do not need to import a format.sas7bcat catalog file.

dataprep.sas
This file contains optional SAS code that is intended to be executed before each run of score code.

batch.sas
This file is created by SAS Enterprise Miner and is used for model retraining by SAS Decision Manager.

pmml.xml
This file contains score code in PMML format.

training.r
This is an optional R script file that is used to retrain R models in SAS Decision Manager.

outmodel.rda
SAS Decision Manager requires this file to save the output parameter estimate for R models.

score.r
This file is an R script that is used to predict new data.

fitstats.xml
This file is created by SAS Enterprise Miner and contains the basic Fit Statistics for the model.

HPDMForest_VARIMPORT.csv
This CSV file contains the variable importance data for a PROC HPFOREST model.

HPDMForest_ITERATION.csv
This CSV file contains statistics across each iteration of a PROC HPFOREST model.

OUTMDLFILE.bin
This is a binary file that contains the PROC HPFOREST model information to be used for scoring.

For information about preparing R model component files, see Appendix 5, "R Model Support," on page 367.

Model Template Properties

Template Properties

Here is a list of the general properties that define the model template.

Property Name	Description
Name	Identifies the name of the template. This property is required. The characters @ \ / * % # & $ ( ) ! ? < > ^ + ~ ` = { } [ ] | ; : ‘ " cannot be used in the name.
Description	Specifies user-defined information about the template.
Type	Specifies the type of the model. SAS Decision Manager supports the following model types: **Analytical Model** specifies the type of model that is associated with the Analytical model function. **Classification Model** specifies the type of model that is associated with the Classification model function. **Prediction Model** specifies the type of model that is associated with the Prediction model function. **Clustering Model** specifies the type of model that is associated with the Segmentation model function.
Tool	Specifies a text value that describes which tool is used to produce this type of model.
Validate	Indicates that SAS Decision Manager verifies that all of the required files are present when users try to import a model. If validation fails, the model will not be successfully imported.
Display name	Specifies a text value that is displayed as the name of the model template.
Score code type	Specifies whether the imported model score code runs by using a **DATA Step** fragment, **SAS Program** code, **PMML**, **Analytic store**, or **DS2**. *Note:* Only models with a score code type of DATA step or DS2 are supported by Decision Builder.

File List Properties

Here is a list of the File List properties that specify the files that are contained in a model.

Property Name	Definition
Name	Identifies the name of the file. This property is required.
Description	Specifies user-defined information about the file.
Required	When it is selected, indicates that the file is a required component file of the model that must be imported before using the model.
Report	When it is selected, indicates that the file is to be included in a SAS package file when a model is published to a channel.
Type	Specifies a file whose type is text or binary.
Fileref	Specifies an eight-character (or fewer) SAS file reference to refer to this file in score.sas code. The fileref is assigned by SAS Decision Manager when a SAS job is submitted.

Note: All user-defined models must have three files.

- score.sas is the model's score code.
- modelinput.sas7bdat is a SAS data set whose variables are used by the model score code. The contents of the data set is not used by SAS Decision Manager.
- modeloutput is a resulting data set when a user runs score.sas against modelinput.sas7bdat. The data set provides output variables that the model creates after a scoring test is executed. The contents of the data set is not used by SAS Decision Manager.

System and User Properties

Here is a list of the system-defined and user-defined properties for a model template. Users can set these properties when they import a model.

Property Name	Description
Name	Identifies the name of the property. This is a required field.
Description	Specifies user-defined information about the property.
Type	Specifies a property whose type is String or Date.
Edit	Indicates that the property can be modified when importing a model or after the model is imported.
Required	Indicates that the property is required.
Initial value	Specifies a text string for the initial value for the property.

Property Name	Description
Display name	Specifies a text value that is displayed as the name of the property.

Specific Properties

Here is a list of specific properties for a model that identify the fundamental model data structures and some of the critical model life cycle dates. Where applicable, project-based or version-based data structures automatically populate properties for model-based data structures.

Property Name	Description
Default scoring input table	Specifies a default SAS data set that is used as the input data table for all of scoring tests within the project. The model's **Default scoring input table** property inherits the property value from the associated version or project, if one is specified.
Default scoring output table	Specifies a default SAS data set that defines the variables to keep in the scoring results table and the scoring test output table. The model's **Default scoring output table** property inherits the property value from the associated version or project, if one is specified.
Default performance table	Specifies the default performance table for all model performance monitoring tasks within a project. A model's **Default performance table** property inherits the property value from the associated version or project, if one is specified. If you do not specify a performance table, some of the monitoring reports might not be enabled.
Default train table	The train table is optional and is used only as information. However, when a value is specified for a model's **Default train table** property.
Expiration date	Specifies a date property by which the selected model is obsolete or needs to be updated or replaced. This property is for informational purposes and is not associated with any computational action. This property is optional.
Model label	Specifies a text string that is used as a label for the selected model in model assessment charts. If no value is provided for the **Model Label** property, the text string that is specified for the **Model Name** property is used. The **Model Label** property can be useful if the Model Name property that is specified is too long for use in plots. This property is optional.

Property Name	Description
Subject	Specifies a text string that is used to provide an additional description for a model, such as a promotional or campaign code. This property is for informational purposes and is not associated with any computational action. This property is optional.
Algorithm	Specifies the computational algorithm that is used for the selected model. This property cannot be modified.
Function	Specifies the function class that was chosen when the associated project was created. The **Function** property specifies the type of output that models in the predictive model project generate.
Modeler	Specifies the Modeler ID or, when Modeler ID is missing, specifies the user ID of the individual who created the model that is stored in the SPK file for SAS Enterprise Miner models. Otherwise, the modeler can be specified during model import for local files.
Tool	Specifies whether the imported model came from SAS Enterprise Miner or from other modeling tools.
Tool version	Specifies the version number of the tool that is specified in the **Tool** property.
Score code type	Specifies whether the imported model score code is a DATA step fragment, ready-to-run SAS code, or a PMML file. Valid values are **DATA step**, **SAS Program**, **PMML**, **Analytic store**, and **DS2**. *Note:* If the model is created using PMML 4.2, the **Score Code Type** is **DATA step** and not **PMML**. For more information, see "PROC PSCORE and PMML Support" on page 365. *Note:* SAS Decision Manager cannot publish models to a database whose **Score Code Type** model property is set either to **SAS Program**, **PMML**, or **DS2**. Models that have a score code type of **Analytic store** can be published only to Hadoop and Teradata. *Note:* Only models with a score code type of DATA step or DS2 are supported by Decision Builder.
Template	Specifies the model template that was used to import the model and to create pointers to its component files and metadata.
Copied from	Specifies where the original model is if this model is copied from another model repository.

Property Name	Description
Target variable	Specifies the name of the target variable for a classification or prediction model. This property can be ignored for segmentation, cluster, and other models that do not use target variables. For example, if a model predicts when GENDER=M, then the target variable is **GENDER**.
Target event value	Specifies a value for the target event that the model attempts to predict. This property is used only when a value is specified for the **Target Variable** property. For example, if a model predicts when GENDER=M, then the target event value is **M**.

Appendix 3

Project Tables

Descriptions of Project Tables

Project Control Tables

A project control table is a data set that contains the projects, models, and segments that are used to create the structure of the projects within a portfolio. The project control table must at least contain a project variable with the name of project_name. If you want to monitor the performance of the champion models within a portfolio, then the project

control table must also contain a segment ID variable. The segment ID variable must also be in the performance tables that are used to monitor performance. If you want to include the models for each project when creating a portfolio, then the control table must also contain the model variable.

Project Input Tables

A project input table is an optional SAS data set that contains the champion model input variables and their attributes. It is a prototype table that can be used to define the project input variables and the variable attributes such as data type and length. A project can have numerous candidate models that use different predictor variables as input. Because the project input table must contain all champion model input variables, the variables in the project input table are a super set of all input variables that any candidate model in the project might use.

A project input table can have one or more observations. Data that is in a project input table is not used by SAS Decision Manager.

If you use a prototype table to define the project input variables, either create the table and register the table to the SAS Metadata Repository in the Data category view or by using SAS Management Console. Tables that are registered using the SAS Management Console must be made available to SAS Decision Manager using the Data category view.

The project input variables must be available to SAS Decision Manager either by specifying a project input table or by defining individual variables before you set a champion model. You can view input variables for a project on the **Input** tab of the project's **Variables** page, or in the Data category view.

Note: An alternative to using prototype tables to define the project input and output variables is to copy the variables from the champion or challenger model, or to modify the project variables. For more information, see "Defining Project Input and Output Variables" on page 133.

See Also

- "Defining Project Input and Output Variables" on page 133
- "Creating Project Input and Output Tables" on page 357

Project Output Tables

A project output table is an optional SAS data set or database table that defines project output variables and variable attributes such as data type and length. It is a prototype table that contains a subset of the output variables that any model in the project might create.

A project output table can have one or more observations. Data that is in a project output table is not used by SAS Decision Manager.

If you use a prototype table to define the project output variables, either create the table and register the table to the SAS Metadata Repository in the Data category view or by using SAS Management Console. Tables that are registered using the SAS Management Console must be made available to SAS Decision Manager using the Data category view. For more information, seeChapter 3, "Managing Data Tables," on page 47.

The project output variables must be available to SAS Decision Manager either by specifying a project output table or by defining individual variables before you set a

champion model. You can view output variables for a project on the **Output** tab of the project's **Variables** page or in the Data category view.

Note: An alternative to using prototype tables to define the project input and output variables is to copy the variables from the champion or challenger model, or to modify the project variables. For more information, see "Defining Project Input and Output Variables" on page 133.

See Also

- "Defining Project Input and Output Variables" on page 133
- "Creating Project Input and Output Tables" on page 357

Scoring Input Tables

A scoring input table is a SAS data set that contains the input data that is used in a scoring test.

Before you can create a scoring test, you must create a scoring input table and register it in the SAS Metadata Repository in the Data category view or by using SAS Management Console. Tables that are registered using the SAS Management Console must be made available to SAS Decision Manager using the Data category view. In SAS Decision Manager, you can view scoring input tables in the Data category view.

See Also

"Creating Scoring Input and Output Tables" on page 359

Scoring Output Tables

A scoring output table is used by a scoring test to define the variables for the scoring results table.

Depending on the mode in which a scoring test is run, the scoring output table can be a prototype table or a physical data table. A scoring test can run in test mode, which is the default mode, or it can run in production mode. In both test mode and production mode, a scoring test output table is used by the scoring test to define the structure of the scoring results table. When the scoring test runs, it creates a scoring results table. In test mode, the scoring results table is stored in the SAS Decision Manager model repository or on a local or network drive. You can view the scoring results table on the **Results** tab of the **Scoring** page for a project. The scoring output table in the SAS Metadata Repository or on a local or network drive is not updated in test mode. In production mode, the contents of the scoring output table in the SAS Metadata Repository or the local or network drive are replaced by the contents of the scoring results table. The scoring results table is not stored in the SAS Decision Manager model repository or on a local or network drive.

Before you can create a scoring test, the scoring output table must be added and accessible from the Data category view. To add the scoring output table to SAS Decision Manager, perform one the following actions:

- Add the table manually by creating the table. Then, register the table in the SAS Metadata Repository in the Data category view or by using SAS Management Console.
- Use the Create a Scoring Output Table feature that is available from the toolbar on the project's **Models** page. When you use the Create a Scoring Output Table window, SAS Decision Manager creates the table in the library that is specified in the

Library box. The table is registered in the SAS Metadata Repository and is available in the Data category view.

You can view scoring output tables in the Data category view.

See Also

"Creating Scoring Input and Output Tables" on page 359

Train Tables

A train table is used to build predictive models. Whether your predictive models are created using SAS Enterprise Miner or you created SAS code models, you used a train table to build your predictive model. SAS Decision Manager uses this same train table. The train table must be registered in the SAS Metadata Repository and accessible to SAS Decision Manager in the Data category view.

You specify a train table as a version-level property. When you define the train table at the version level, the table can be used to build all predictive models that are defined on the **Models** page for a project.

In SAS Decision Manager, train tables are used for information purposes only with one exception. SAS Decision Manager uses train tables to validate scoring results immediately after you publish a scoring function or model scoring files, and if the **Validate scoring results** box is selected when you publish scoring functions or model scoring files to a database.

Note: A train table cannot contain an input variable name that starts with an underscore.

For information about registering a train table using the Data category view, see Chapter 3, "Managing Data Tables," on page 47.

Test Tables

A test table is used to create the Dynamic Lift report and the Interval Target Variable report that can be used to identify the champion model. Test tables are typically a subset of a train table, and they are identical in table structure to the corresponding train table. Update test tables by creating a new subset of the corresponding train table.

To view test tables in SAS Decision Manager, the tables must be registered in the SAS Metadata Repository. In SAS Decision Manager, you can view test tables in the Data category view.

After a test table is added to SAS Decision Manager, you can specify the table in the **Default test table** field in the project properties.

For information about registering test tables using the Data category view, see Chapter 3, "Managing Data Tables," on page 47.

See Also

"Creating a Test Table" on page 360

Performance Tables

A performance table is a SAS data set that is used as the input table for each SAS Decision Manager performance definition. A performance definition is used to monitor a champion model's performance by comparing the observed target variable values with the predicted target variable values. A performance table is a sampling of operational

data that is taken at a single point in time. Each time you run a performance definition, you use a new performance table to take a new sampling of the operational data. For example, a champion model is deployed to a production environment for the first time in March 2013. You might want to take a new sampling of the operational data in June 2013, September 2013, and January 2014. These new tables are performance tables in the context of SAS Decision Manager.

To view a performance table in SAS Decision Manager, you must register the tables in the SAS Metadata Repository using the Data category view or by using SAS Management Console. You can view performance tables in the Data category view. After a performance table is registered, you can specify the table in the **Default performance table** field in the project properties. The default performance table value at the project level is the default value for the **Performance data source** field in the Edit Performance Definition wizard.

Note: If you run SAS Decision Manager report macros outside of SAS Decision Manager to monitor a champion model's performance, the macros cannot access the performance tables in SAS Decision Manager to create model performance monitoring reports.

See Also

"Creating a Performance Table" on page 360

See Also

"Remove a Table" on page 54

Creating Project Input and Output Tables

Create a Project Input Table

You can create a project input table either from the train table that you used to develop your model, or you can define the project variables in a DATA step. The project input table must include the input variables that are used by the champion model. Therefore, if you have several candidate models for your project, make sure that all candidate model input variables are included in the project input table. If you create the project input table from the train table, be sure to exclude the target variable from the project input table.

Here is one method that you can use to create the project input table from the train table. Use the SET statement to specify the train table and the DROP or KEEP statements to specify the variables from the train table that you want in the project input table. You can drop the target variable or keep all variables except the target variable.

This DATA step creates the project input table from the train table and drops the target variable Bad:

```
data hmeqtabl.invars;
   set hmeqtabl.training (obs=1);
   drop bad;
run;
```

This DATA step creates the project input table from the train table and keeps all variables except for the target variable Bad:

```
data hmeqtabl.invars;
```

```
   set hmeqtabl.training (obs=1);
   keep mortdue reason delinq debinc yoj value ninq job clno derog clag loan;
run;
```

You can also create the project input table using the LENGTH statement to specify the variables and their type and length. You could also specify the LABEL, FORMAT, or INFORMAT statements, or the ATTRIB statement to specify additional variable attributes. The following DATA step uses the LENGTH statement to specify the project input variables in the table:

```
data hmeqtabl.invars;
   length mortdue 8 reason $7 delinq 8
          debinc 8 yoj 8 value 8
          ninq 8 job $7 clno 8 derog 8
          clag 8 loan 8;
run;
```

If you find that you need to modify the project input variables after you have created a project input table, you can use the project's **Variables** page to modify the project variables. For more information, see "Defining Project Input and Output Variables" on page 133.

See Also

- *SAS 9.4 Statements: Reference*
- *SAS 9.4 Language Reference: Concepts*

Create a Project Output Table

You can create a project output table either from the train table that you used to develop your model, or you can define the project variables in a DATA step. The project output table includes only output variables that are created or modified by the champion model. Therefore, if you have several candidate models for your project, you must make sure that all project output variables are mapped to the champion model output variables.

To create the project output table using the training table, use the SET statement to specify the training table, and use the KEEP statement to specify the variables from the training table that you want in the project output table. The following DATA step creates the project output table Hmeqtabl.Outvars:

```
data hmeqtabl.outvars;
   set hmeqtabl.training (obs=1);
   %include "c:\temp\score.sas";
   keep score;
run;
```

The following DATA step creates the same project output table using the LENGTH statement to specify the output variable and its type and variable length:

```
data hmeqtabl.outvars;
   length score 8;
run;
```

If you find that you need to modify the project output variables after you have created a project output table, you can use the project's **Variables** page to modify the project variables. For more information, see "Defining Project Input and Output Variables" on page 133.

Creating Scoring Input and Output Tables

About Scoring Input and Output Tables

The scoring input table is a data table whose input is used by the scoring test to score a single model. The scoring input table must contain the variables and input data for the variables that the model requires. Typically, a scoring table is identical to its corresponding train table except that the target variables in the train table are not included in the scoring table.

A scoring output table contains the data that is produced when you execute a scoring test. You can provide a scoring output table or you can create a scoring output table definition in SAS Decision Manager. When a scoring test is executed, SAS Decision Manager uses the scoring output table definition to create the scoring output table. The name of the scoring output table definition is used as the name of the scoring output table.

You can create a scoring output table definition by using the Create a Scoring Output Table function on the **Models** page. In the Create a Scoring Output Table window, you select variables from a scoring input table as well as variables from the model's output. The variables in the **Input Variables** table are variables from the scoring input table if one is specified for the **Default scoring input table** property for a project or model property. Otherwise, the **Input Variables** table is empty. The **Output Variables** that appear in the window are model output variables. You use the variables from both tables to create the scoring output table.

SAS Decision Manager saves the table definition as metadata in the SAS Metadata Repository. The location of the metadata is defined by the SAS library that you specify when you create the output table definition. After SAS Decision Manager creates the table definition, the table can be selected as the output table for subsequent scoring tests.

A SAS Decision Manager scoring test can run in test mode, which is the default mode, or it can run in production mode. When the test runs, it populates a scoring output table. In test mode, the scoring output table is stored in the SAS Decision Manager model repository. You view the table under the scoring test on the project's **Scoring** page. In production mode, if the scoring output table is a table that you provided, that table is updated. If you created a scoring test output definition, the scoring output table is located in the designated SAS library that you specified when you created the table definition in the Create a Scoring Output Table window. The production scoring output table is not stored in the SAS Decision Manager repository.

Create a Scoring Input Table

This DATA step creates a scoring test input table from customer data, keeping 500 rows from the train table:

```
data hmeqtabl.scorein;
   set hmeqtabl.customer (obs=500);
   keep mortdue reason delinq debinc yoj value ninq job clno derog clage loan;
run;
```

Create a Scoring Output Table

You can create a scoring output table using the Create a Scoring Output Table window that you open from the project's **Models** page. The Create a Scoring Output Table window enables you to select the variables that you want to include in your scoring output table. If the library that you select in the Create a Scoring Output Table window is a folder in the SAS Metadata Repository, SAS Decision Manager registers the table in the repository. You can view the table in the Data category view of SAS Decision Manager. For information, see "Create Scoring Output Tables" on page 194.

You can also create a scoring output table using a DATA step to keep or drop variables from the train table.

The input variables that you might want to keep in the output data set are key variables for the table. Key variables contain unique information that distinguishes one row from another. An example would be a customer ID number.

This DATA step keeps the input variable CLNO, the client number, which is the key variable, and the output variable SCORE:

```
data hmeqtabl.scoreout;
   length clno 8 score 8;
run;
```

Creating a Test Table

The test table is used during model validation by the Dynamic Lift report. You can create a test table by taking a sampling of rows from the original train table, updated train table, or any model validation table that is set aside at model training time. This DATA step randomly selects approximately 25% of the train table to create the test table:

```
data hmeqtabl.test;
  set hmeqtabl.train;
  if ranuni(1234) < 0.25;
run;
```

See Also

"Create a Dynamic Lift Report" on page 209

Creating a Performance Table

About Performance Tables

Here are the requirements for a performance table:

- the input variables that you want reported in a Characteristic report
- if you have score code:
  - all input variables that are used by the champion model or challenger models
  - all output variables that are used by the champion model or challenger models

- if you have no score code:
  - the actual value of the dependent variable and the predicted score variable
  - all output variables that you want reported in a Stability Report

You create a performance table by taking a sampling of data from an operational data mart. Make sure that your sampling of data includes the target or response variables. The data that you sample must be prepared by using your extract, transform, and load business processes. When this step is complete, you can then use that data to create your performance table.

As part of the planning phase, you can determine how often you want to sample operational data to monitor the champion model performance. Ensure that the operational data that you sample and prepare represents the period that you want to monitor. For example, to monitor a model that determines whether a home equity loan could be bad, you might want to monitor the model every six months. To do this, you would have two performance tables a year. The first table might represent the data from January through June, and the second table might represent the data from July through December.

Here is another example. You might want to monitor the performance of a champion model that predicts the delinquency of credit card holders. In this case, you might want to monitor the champion model more frequently, possibly monthly. You would need to prepare a performance table for each month in order to monitor this champion model.

In addition to planning how often you sample the operational data, you can also plan how much data to sample and how to sample the data. Examples in this section show you two methods of sampling data and naming the performance tables. You can examine the sampling methods to determine which might be best for your organization.

Naming a Performance Table for Use with the Edit Performance Definition Wizard

The Edit Performance Definition wizard is a graphical interface to assist you in creating a performance definition to monitor the champion model performance. When you run the Edit Performance Definition wizard, you specify a performance table that has been registered to the SAS Metadata Repository. When you create a performance table, you can collect and name the performance table using a method that is most suitable for your business process.

See Also

"Overview of Performance Monitoring" on page 243

Create a Performance Table

You can use the following DATA steps as examples to create your performance tables.

This DATA step creates a performance table using 5,000 sequential observations from the operational data:

```
data hmeqtabl.perform;
   set hmeqop.JulDec (firstobs=12001 obs=17000);
run;
```

This DATA step creates a performance table from operational data for the past six months of the year. The IF statement creates a random sampling of approximately 10% of the operational data:

```
data hmeqtabl.perform;
  set hmeqop.JulDec;
  if ranuni(1234) < 0.1;
run;
```

Using Tables from a Local or Network Drive

About Using Tables from a Local or Network Drive

If you have migrated or upgraded from a previous release of SAS Decision Manager, the start-up code that enabled you to use tables from the local SAS Workspace Server or network drive is still available. In SAS Decision Manager 3.2, you can no longer define a libref to use tables on a local or network drive. Also, in SAS Decision Manager 3.2, tables must be registered in the SAS Metadata Repository and accessible in the Data category view. If the libref was defined before you migrated or upgraded, you can use the local or network tables to complete these SAS Decision Manager tasks:

- Create a project
- Create projects from a control table
- Specify project input and output variables
- Create a scoring test
- Create a model retrain definition
- Create reports
- Create a performance definition

The start-up code that was migrated can be viewed using the Edit Start-up Code window. You can access this window from the **Actions** menu on the toolbar in the Projects and Portfolios category views.

Here is an example LIBNAME statement:

```
libname SalesLib 's:\sales\2013\october';
```

Edit Start-Up Code

To edit the start-up code:

1. Select **Actions** ⇨ **Edit Start-up Code**. The Edit Start-up Code window appears.

2. Enter the SAS code.
3. Click **Run Now**.
4. Click the **Log** tab to see the SAS log.
5. Click **OK**. The SAS code is saved in the Edit Start-up Code window.

   Note: If you save the code without running it by clicking **OK**, the code automatically runs the next time the middle-tier server starts.

Delete a Libref

To delete a libref:

1. Select **Actions** ⇨ **Edit Start-up Code**
2. Type `libname` *libref* `clear`.
3. Click **Run Now**.

Appendix 4

PROC PSCORE and PMML Support

Overview

PMML is an XML markup language that was developed to exchange predictive and statistical models between modeling systems and scoring platforms. Users can import the majority of standard-compliant PMML models and score them within a SAS environment via the SAS PSCORE procedure.

PROC PSCORE Functionality

The SAS PSCORE procedure generates SAS DATA step score code that is functionally equivalent to the PMML model. The generated score code can be executed on all platforms that are supported by SAS to score the data sets. You can submit the score code in SAS Enterprise Miner via the Program Editor, SAS Enterprise Miner Project code, or within a SAS Enterprise Miner Process Flow Diagram, via the SAS Code node. However, the SAS Enterprise Miner UI environment is not necessary to run the score code.

Note: The PSCORE procedure generates both DATA step code and DS2 code. However, only DATA step model score code is generated when you are registering a PMML model into SAS Decision Manager.

Supported Versions

PROC PSCORE currently supports the use of PMML 4.2. Earlier versions of PMML are not supported for use with PROC PSCORE.

Supported PMML Models

SAS PROC PSCORE supports the following types of PMML models:

- Regression
- Trees
- Neural Networks
- Clustering models
- Scorecard
- Vector Machine
- Naïve Bayes
- Baseline models

The following models are supported on an experimental basis:

- Time Series
- General Regression

Requirements for PROC PSCORE

In order to use PROC PSCORE, you must have SAS 9 or later, a well formed PMML modeling file, and Write access to the output directory for the DATA step score file. A SAS Enterprise Miner license is not necessary to run PROC PSCORE.

PROC PSCORE Usage

```
PROC PSCORE PMML FILE = "<full-pathname-of-PMML-file>"
     DS FILE = "<full-pathname-of-output-DS-file>"
```

PROC PSCORE Example

/*Run the PSCORE procedure on a generated PMML file*/

```
PROC PSCORE PMML FILE = "C:\temp\heart_pmml1.xml"
     DS FILE = "C:\temp\ds_heart_score.sas";
  run;
```

See Also

SAS Enterprise Miner 15.1 PMML Support

Appendix 5

R Model Support

Overview of Using R Models with SAS Decision Manager

R is a freely available language and environment for statistical computing and graphics. Using the open architecture of SAS Decision Manager, you can register and import R models. SAS Decision Manager requires a model template file and model component files that are created specifically for R models.

The following SAS components are required to use R models in SAS Decision Manager:

- Ensure that the installed R language version is 2.13.0 or later.
- SAS/IML. You must license SAS/IML because the IML procedure is required to export SAS data sets to R and to submit R code.
- The RLANG system option. You must set this system option.
- The R language and SAS/IML must be installed on the same machine where the SAS Model Manager server is installed.

SAS Decision Manager supplies three R model templates that you can use, or you can create your own template as well. The R model templates that are provided by SAS Decision Manager support the analytic, classification, and prediction model functions. The segmentation model function is not supported for R models.

After the model component files are registered, you can perform all SAS Decision Manager functions except for exporting an R model to the SAS Metadata Repository.

To use R models in SAS Decision Manager, do the following tasks:

1. Ensure that the RLANG system option is set. To have the RLANG system option set when SAS starts, have your site administrator add the RLANG system option to the SAS configuration file.
2. Build an R model. For more information, see “Build an R Model” on page 368.

SAS/IML must be installed before you build an R model.

3. Ensure that you have a model template file. For more information, see "Prepare an R Model Template File" on page 369.
4. Ensure that you have the required model component files. For more information, see "Prepare R Model Component Files" on page 370.
5. Import the R model. For more information, see "Import Models from Local Files" on page 150.

Preparing R Model Files to Use with SAS/IML

Build an R Model

Use the following SAS code to create an R model and save it in the outmodel.rda model component file:

```
/* Define the libref to the SAS input data set.   */

libname libref "path-to-input-data-set";

/* Use PROC IML to export the SAS input data set to the R input data set. */

proc iml;
    run ExportDatasetToR("input-data-set" , "R-matrix-input");

/* Submit the model-fitting R code.   */

submit /R;
        attach(R-matrix-input)

        # ----------------------------------------------
        # FIT THE MODEL
        # ----------------------------------------------
        model-name<- model-fitting-function

        # ----------------------------------------------
        # SAVE THE PARAMETER ESTIMATE TO LOCAL FILE OUTMODEL.RDA
        # ----------------------------------------------
        save(model-name, file="path/outmodel.rda")
    endsubmit;
run;
quit;
```

Supply the following values:

path-to-input-data-set
is the path to the library where the input data set is stored.

input-data-set
is the name of the input data set.

R-matrix-input
is the R input data.

model-name
is the name of the model.

model-fitting-function
is the R formula that is used to fit the model.

path
is the path to where outmodel.rda is to be stored.

Here is an example of creating an R model using the HMEQ train data set as the SAS input data set:

```
libname mmsamp "!sasroot\mmcommon\sample";
proc iml;
    run ExportDatasetToR("mmsamp.hmeq_train" , "mm_inds");
    submit /R;
        attach(mm_inds)

        # ----------------------------------------------
        # FIT THE LOGISTIC MODEL
        # ----------------------------------------------
        logiten<- glm(BAD ~ VALUE + factor(REASON) + factor(JOB) + DEROG +
                   CLAGE + NINQ + CLNO , family=binomial)

        # ----------------------------------------------
        # SAVE THE PARAMETER ESTIMATE TO LOCAL FILE OUTMODEL.RDA
        # ----------------------------------------------
        save(logiten, file="c:/RtoMMfiles/outmodel.rda")
    endsubmit;
run;
quit;
```

Prepare an R Model Template File

SAS Decision Manager provides three R model templates that you can use as a model template for your R model:

- RClassification
- RPrediction
- RAnalyticalmodel

To view these model templates:

1. From the Projects category view, Click and select **Manage Templates**. The Manage Templates appears.
2. Select an R model template and click .
3. Review the model template to make sure that it contains all of the model component files and properties for your model. If it does, you can use this template to import your R model. To customize the model template, you can copy the XML content from one of the supplied template files and make modifications using a text editor. You can then create a new model template using the modified XML content and the model template to the SAS Content Server.

   To create a custom R model template, see "Model Template Component Files" on page 340 and "User-Defined Model Templates" on page 165.

Prepare R Model Component Files

R Model Component Files for Executing R Models Using SAS/IML

To submit R models from SAS Decision Manager using SAS/IML, you need several model component files:

- modelinput.sas7bdat
- modeloutput.sas7bdat
- target.sas7bdat
- inputvar.xml
- outputvar.xml
- targetvar.xml
- outmodel.rda
- score.r
- score.sas
- training.r (not required if you do not retrain your R model)
- training.sas (not required if you do not retrain your R model)

You create the modelinput.sas7bdat, modeloutput.sas7bdat, target.sas7bdat, inputvar.xml, outputvar.xml, and targetvar.xml files as you would for importing a SAS code file. For more information, see "Model Template Component Files" on page 340.

The remaining files, outmodel.rda, score.r, score.sas training.r, and training.sas require additional file preparation.

Create outmodel.rda

The outmodel.rda file contains the output parameter estimate. This file is used by SAS Decision Manager to register and score the model. You create outmodel.rda when you build an R model. See "Build an R Model" on page 368. The outmodel.rda file uses the R function save() to save the scoring results.

Here is the syntax of an outmodel.rda file:

```
save(model-name, file="path/outmodel.rda")
```

Supply the following values:

model-name
is the name of the R model.

path
is the system path to the location where outmodel.rda is stored.

Here is an example outmodel.rda file:

```
save(logiten, file="c:/temp/outmodel.rda")
```

Create score.r

The score.r script is an R script that is used to score data. You can use the following R script to create score.r:

```
attach(R-matrix-input)
```

```
# ----------------------------------------------
# LOAD THE OUTPUT PARAMETER ESTIMATE FROM FILE OUTMODEL.RDA
# ----------------------------------------------
load('&_mm_scorefilesfolder/outmodel.rda')

# ----------------------------------------------
# SCORE THE MODEL
# ----------------------------------------------
score<- predict(model-name, type="response", newdata=R-matrix-input)

# ----------------------------------------------
# MERGING PREDICTED VALUE WITH MODEL INPUT VARIABLES
# ----------------------------------------------
mm_outds <- cbind(R-matrix-input, score)
```

Supply the following values:

R-matrix-input
is the name of the input R matrix file that you specified in the ExportDatasetToR function in the IML procedure. See "Build an R Model" on page 368.

score
is the output variable. The value for *score* must match the output variable that is defined in modeloutput.sas7bdat and outputvar.xml.

model-name
is the name of the R model. The value of *model-name* must match the R save function *model-name* argument that is specified in the outmodel.rda file.

Here is an example score.r file:

```
attach(mm_inds)

# ----------------------------------------------
# LOAD THE OUTPUT PARAMETER ESTIMATE FROM FILE OUTMODEL.RDA
# ----------------------------------------------
load('&_mm_scorefilesfolder/outmodel.rda')

# ----------------------------------------------
# PREDICT
# ----------------------------------------------
score<- predict(logiten, type="response", newdata=mm_inds)

# ----------------------------------------------
# MERGE THE PREDICTED VALUE WITH MODEL INPUT VARIABLES
# ----------------------------------------------
mm_outds <- cbind(mm_inds, score)
```

Create score.sas

The score.sas program defines the score test information in a data set and calls the %mmbatch macro. When you submit the %mmbatch macro, the task mm_r_model_train_main completes the following tasks:

- transforms a scoring data set to an R data frame
- generates and submits R code for scoring
- transforms the scored output to a SAS data set for reporting in SAS Decision Manager

Here is the score.sas program:

```
filename tmp catalog "sashelp.modelmgr.mm_include.source";
%include tmp;
filename tmp;

data work.mm_score_task_information;
   length role $ 8;
   length name $ 80;
   length value $ 200;

   role = "input";
   name = "importedData";
   value = "&_mm_inputds";
   output;

   role = "input";
   name = "modelID";
   value = "&_mm_modelID";
   output;

   role = "output";
   name = "exportedData";
   value = "&_mm_outputds";
   output;

   role = "input";
   name = "dataRole";
   value = "output-variable-name";
   output;

   role = "input";
   name = "p_Target";
   value = "output-variable-name";
   output;
run;

/* mm_r_model_score_main is a SAS Decision Manager process flow that is used to run */
/* R model scripts using PROC IML.                                                   */

%mmbatch(task=mm_r_model_score_main, taskprops= mm_score_task_information);
```

Supply the following value:

output-variable-name
is the output variable that is defined in modeloutput.sas7bdat or modeloutput.xml.

To print verbose SAS logs, add the following lines before the RUN statement in the previous DATA step:

```
role = "input";
   name = "_mm_trace";
   value = "ON";
   output;
```

Create training.r

The training.r script is an R script that is used to build a train model. Use the following script for the training.r file. In the R save function, the path in the file= argument must be &_MM_TrainResultFolder.

You can use the following script to create training.r:

```
attach(R-matrix-input)

# ----------------------------------------------
# FIT THE LOGISTIC MODEL
# ----------------------------------------------
model-name<- model-fitting-function

# ----------------------------------------------
# SAVE THE OUTPUT PARAMETER ESTIMATE TO LOCAL FILE OUTMODEL.RDA
# ----------------------------------------------
save(model-name, file="&_MM_TrainResultFolder/outmodel.rda")
```

Supply the following values:

R-matrix-input
is the name of the R matrix that is specified in the ExportMatrixToR function that is used to build a model using the IML procedure.

model-name
is the name of the R model.

model-fitting-function
is an R model fitting function, such as lm() or glm().

Here is an example training.r R script to build the HMEQ R train model:

```
attach(mm_inds)

# ----------------------------------------------
# FIT THE LOGISTIC MODEL
# ----------------------------------------------
logiten<- glm(BAD ~ VALUE + factor(REASON) + factor(JOB) + DEROG + CLAGE +
              NINQ + CLNO , family=binomial)

# ----------------------------------------------
# SAVE THE OUTPUT PARAMETER ESTIMATE TO LOCAL FILE OUTMODEL.RDA
# ----------------------------------------------
save(logiten, file="&_MM_TrainResultFolder/outmodel.rda")
```

Create training.sas

If you do not need to retrain your R model in SAS Decision Manager, you do not need this file.

The training.sas program defines the train task information in a data set and calls the %mmbatch macro. When you submit the %mmbatch macro, the task mm_r_model_train_main completes the following tasks:

- transforms a training data set to an R data frame
- generates and submits R code for training
- registers the training output parameter estimate file in SAS Decision Manager

Here is the training.sas file:

```
filename tmp catalog "sashelp.modelmgr.mm_include.source";
%include tmp;
filename tmp;

data work.mm_train_task_information;
   length role $ 8;
   length name $ 80;
   length value $ 200;

   role = "input";
   name = "trainData";
   value = "&_mm_inputds";
   output;

   role = "input";
   name = "modelID";
   value = "&_mm_modelID";
   output;
run;

/* mm_r_model_train_main is a SAS Decision Manager process flow that is used to run */
/* R model scripts using PROC IML.                                                   */

%mmbatch(task=mm_r_model_train_main, taskprops= mm_train_task_information);
```

To print verbose SAS logs, add the following lines before the RUN statement in the previous DATA step:

```
role = "input";
   name = "_mm_trace";
   value = "ON";
   output;
```

Appendix 6

Statistical Measures Used in Basel II Reports

Overview of Statistical Measures Used for Basel III Reports

SAS Decision Manager Basel III reports use several statistical measures to validate the stability, performance, and calibration for the two key types of Basel III risk models: the Probability of Default (PD) model and the Loss Given Default (LGD) model.

The statistical measures for model validation are grouped into three categories:

Category	Description
Model Stability	Tracks the change in distribution of the modeling data and scoring data.
Model Performance	• Measures the ability of a model to discriminate between customers with accounts that have defaulted, and customers with accounts that have not defaulted. The score difference between non-default and default accounts helps determine the required cutoff score. The cutoff score helps predict whether a credit exposure is a default account. • Measures the relationship between the actual default probability and the predicted default probability. This helps you understand the performance of a model over a time period.
Model Calibration	Checks the accuracy of the PD and LGD models by comparing the correct quantification of the risk components with the available standards.

The sections that follow describe the measures, statistics, and tests that are used to create the PD and LGD reports.

Model Stability Measure

The following table describes the model stability measure that is used to create the PD report and the LGD reports.

Measure	Description	PD Report	LGD Report
System Stability Index (SSI)	SSI monitors the score distribution over a time period.	Yes	Yes

Model Performance Measures and Statistics

The following table describes the model performance measures that are used to create the PD and LGD reports.

Measure	Description	PD Report	LGD Report
Accuracy	Accuracy is the proportion of the total number of predictions that were correct.	Yes	No
Accuracy Ratio (AR)	AR is the summary index of Cumulative Accuracy Profile (CAP) and is also known as Gini coefficient. It shows the performance of the model that is being evaluated by depicting the percentage of defaulted accounts that are captured by the model across different scores.	Yes	Yes
Area Under Curve (AUC)	AUC can be interpreted as the average ability of the rating model to accurately classify non-default accounts and default accounts. It represents the discrimination between the two populations. A higher area denotes higher discrimination. When AUC is 0.5, it means that non-default accounts and default accounts are randomly classified, and when AUC is 1, it means that the scoring model accurately classifies non-default accounts and default accounts. Thus, the AUC ranges between 0.5 and 1.	Yes	No
Bayesian Error Rate (BER)	BER is the proportion of the whole sample that is misclassified when the rating system is in optimal use. For a perfect rating model, the BER has a value of zero. A model's BER depends on the probability of default. The lower the BER, and the lower the classification error, the better the model.	Yes	No
D Statistic	The D Statistic is the mean difference of scores between default accounts and non-default accounts, weighted by the relative distribution of those scores.	Yes	No
Error Rate	The Error Rate is the proportion of the total number of incorrect predictions.	Yes	No

Measure	Description	PD Report	LGD Report
Information Statistic (I)	The Information Statistic value is a weighted sum of the difference between conditional default and conditional non-default rates. The higher the value, the more likely a model can predict a default account.	Yes	No
Kendall’s Tau-b	Kendall's tau-b is a nonparametric measure of association based on the number of concordances and discordances in paired observations. Kendall's tau values range between -1 and +1, with a positive correlation indicating that the ranks of both variables increase together. A negative association indicates that as the rank of one variable increases, the rank of the other variable decreases.	Yes	No
Kullback-Leibler Statistic (KL)	KL is a non-symmetric measure of the difference between the distributions of default accounts and non-default accounts. This score has similar properties to the information value.	Yes	No
Kolmogorov-Smirnov Statistic (KS)	KS is the maximum distance between two population distributions. This statistic helps discriminate default accounts from non-default accounts. It is also used to determine the best cutoff in application scoring. The best cutoff maximizes KS, which becomes the best differentiator between the two populations. The KS value can range between 0 and 1, where 1 implies that the model is perfectly accurate in predicting default accounts or separating the two populations. A higher KS denotes a better model.	Yes	No
1–PH Statistic (1–PH)	1-PH is the percentage of cumulative non-default accounts for the cumulative 50% of the default accounts.	Yes	No
Mean Square Error (MSE), Mean Absolute Deviation (MAD), and Mean Absolute Percent Error (MAPE)	MSE, MAD, and MAPE are generated for LGD reports. These statistics measure the differences between the actual LGD and predicted LGD.	No	Yes

Measure	Description	PD Report	LGD Report
Pietra Index	The Pietra Index is a summary index of Receiver Operating Characteristic (ROC) statistics because the Pietra Index is defined as the maximum area of a triangle that can be inscribed between the ROC curve and the diagonal of the unit square. The Pietra Index can take values between 0 and 0.353. As a rating model's performance improves, the value is closer to 0.353. This expression is interpreted as the maximum difference between the cumulative frequency distributions of default accounts and non-default accounts.	Yes	No
Precision	Precision is the proportion of the actual default accounts among the predicted default accounts.	Yes	No
Sensitivity	Sensitivity is the ability to correctly classify default accounts that have actually defaulted.	Yes	No
Somers' D (p-value)	Somers' D is a nonparametric measure of association that is based on the number of concordances and discordances in paired observations. It is an asymmetric modification of Kendall's tau. Somers' D differs from Kendall's tau in that it uses a correction only for pairs that are tied on the independent variable. Values range between -1 and +1. A positive association indicates that the ranks for both variables increase together. A negative association indicates that as the rank of one variable increases, the rank of the other variable decreases.	Yes	No
Specificity	Specificity is the ability to correctly classify non-default accounts that have not defaulted.	Yes	No
Validation Score	The Validation Score is the average scaled value of seven distance measures, anchored to a scale of 1 to 13, lowest to highest. The seven measures are the mean difference (D), the percentage of cumulative non-default accounts for the cumulative 50% of the default accounts (1-PH), the maximum deviation (KS), the Gini coefficient (G), the Information Statistic (I), the Area Under the Curve (AUC), or Receiver Operating Characteristic (ROC) statistic, and the Kullback-Leibler statistic (KL).	Yes	No

Model Calibration Measures and Tests

The following table describes the model calibration measures and tests that are used to create the PD and LGD reports:

Measure	Description	PD Report	LGD Report
Binomial Test	The Binomial Test evaluates whether the PD of a pool is correctly estimated. It does not take into account correlated defaults, and it generally yields an overestimate of the significance of deviations in the realized default rate from the forecast rate. The Modified Binomial Test now addresses the overestimate. This test takes into account the correlated defaults. The default correlation coefficient in SAS Decision Manager is 0.04. By using past banking evaluations, you can use these rho values[1]: rho=0.04 Qualifying revolving retail rho=0.15 Residential mortgage rho=0.16 Other retail rho=0.24 Corporations, sovereign, and banks If the number of default accounts per pool exceeds either the low limit (binomial test at 0.95 confidence) or high limit (binomial test at 0.99 confidence), the test suggests that the model is poorly calibrated. To change the default rho value, contact your application administrator. The value is a report option in SAS Management Console.	Yes	No
Brier Skill Score (BSS)	BSS measures the accuracy of probability assessments at the account level. It measures the average squared deviation between predicted probabilities for a set of events and their outcomes. Therefore, a lower score represents a higher accuracy.	Yes	No

1 Robert Rauhmeier, "PD-Validation: Experience from Banking Practice," in *The Basel II Risk Parameters*, ed. Bernd Engelmann and Robert Rauhmeier (Berlin: Springer, 2011). https://doi.org/10.1007/3-540-33087-9_14

Measure	Description	PD Report	LGD Report
Confidence Interval	The Confidence Interval indicates the confidence interval band of the actual PD or LGD for a pool. The Probability of Default (PD) report provides the PD that is estimated from the model and the actual PD with its confidence interval limits. If the PD that is estimated from the model is within the confidence interval limits of the actual PD, then the model outcomes are consistent with the actual outcomes. For the Loss Given Default (LGD) report, confidence intervals are based on the pool-level average of the estimated LGD, plus or minus the pool-level standard deviation, and multiplied by the 1-(alpha/2) quantile of the standard normal distribution.	Yes	Yes
Correlation Analysis	The model validation report for LGD provides a correlation analysis of the estimated LGD with the actual LGD. This correlation analysis is an important measure for a model's usefulness. The Pearson correlation coefficients are provided at the pool and overall levels for each time period are examined.	No	Yes
Hosmer-Lemeshow Test (p-value)	The Hosmer-Lemeshow test is a statistical test for goodness-of-fit for classification models. The test assesses whether the observed event rates match the expected event rates in pools. Models for which expected and observed event rates in pools are similar are well calibrated. The p-value of this test is a measure of the accuracy of the estimated default probabilities. The closer the p-value is to zero, the poorer the calibration of the model.	Yes	No
Mean Absolute Deviation (MAD)	MAD is the distance between the account level estimated and the actual loss LGD, averaged at the pool level.	No	Yes
Mean Absolute Percent Error (MAPE)	MAPE is the absolute value of the account-level difference between the estimated and actual LGD, divided by the estimated LGD, and averaged at the pool level.	No	Yes
Mean Squared Error (MSE)	MSE is the squared distance between the account level estimated and actual LGD, averaged at the pool level.	No	Yes

Measure	Description	PD Report	LGD Report
Normal Test	The Normal Test compares the normalized difference of predicted and actual default rates per pool with two limits estimated over multiple observation periods. This test measures the pool stability over time. If a majority of the pools lie in the rejection region, to the right of the limits, then the pooling strategy should be revisited.	Yes	No
Observed versus Estimated Index	The observed versus estimated index is a measure of closeness of the observed and estimated default rates. It measures the model's ability to predict default rates. The closer the index is to zero, the better the model performs in predicting default rates.	Yes	No
Traffic Lights Test	The Traffic Lights Test evaluates whether the PD of a pool is underestimated, but unlike the binomial test, it does not assume that cross-pool performance is statistically independent. If the number of default accounts per pool exceeds either the low limit (Traffic Lights Test at 0.95 confidence) or high limit (Traffic Lights Test at 0.99 confidence), the test suggests the model is poorly calibrated.	Yes	No

Glossary

analytical model
a statistical model that is designed to perform a specific task or to predict the probability of a specific event.

baseline
the initial performance prediction against which the output data from later tasks is compared.

business entity
an object in your business domain. An entity has one or more terms, which are attributes of an entity. For example, the business entity could be the customer, and the associated terms could be the name, account number, account type, and so on.

business rule (rule)
a statement of business logic that specifies conditions to be evaluated and actions to be taken if those conditions are satisfied.

business rule flow package
a business rule flow that has been saved to an XML file.

business rule service
a business rule flow that has been implemented as a web service.

candidate model
a predictive model that evaluates a model's predictive power as compared with the champion model's predictive power.

challenger model
a model that is compared and assessed against a champion model for the purpose of replacing the champion model in a production scoring environment.

champion model
the best predictive model that is chosen from a pool of candidate models in a data mining environment.

channel
See content channel.

classification model
a predictive model that has a categorical, ordinal, or binary target.

clustering model
a model in which data sets are divided into mutually exclusive groups in such a way that the observations for each group are as close as possible to one another, and different groups are as far as possible from one another.

content channel (channel)
a virtual communication path for distributing information. In SAS, a channel is identified with a particular topic. Using the features of the Publishing Framework, authorized users or applications can publish digital content to the channel, and authorized users and applications can subscribe to the channel in order to receive the content.

data object
an object that holds the business data that is required to execute workflow tasks.

data source (source)
a table, view, or file from which you will extract information. Sources can be in any format that SAS can access, on any supported hardware platform. The metadata for a source is typically an input to a job.

file reference
See fileref.

fileref (file reference)
a name that is temporarily assigned to an external file or to an aggregate storage location such as a directory or a folder. The fileref identifies the file or the storage location to SAS. *See also* libref.

Gini coefficient
a benchmark statistic that is a measure of the inequality of distribution, and that can be used to summarize the predictive accuracy of a model.

identity
See metadata identity.

input variable
a variable that is used in a data mining process to predict the value of one or more target variables.

key
See lookup key.

Kolmogorov-Smirnov chart
a chart that shows the measurement of the maximum vertical separation, or deviation between the cumulative distributions of events and non-events.

library reference
See libref.

libref (library reference)
a SAS name that is associated with the location of a SAS library. For example, in the name MYLIB.MYFILE, MYLIB is the libref, and MYFILE is a file in the SAS library.

lookup key (key)
a value that uniquely identifies a specific record and its order among other records in a database or table.

lookup table
a table that contains lookup keys and their corresponding values.

lookup value
the value that is associated with a lookup key in a lookup table.

metadata
descriptive data about data that is stored and managed in a database, in order to facilitate access to captured and archived data for further use.

metadata identity (identity)
a metadata object that represents an individual user or a group of users in a SAS metadata environment. Each individual and group that accesses secured resources on a SAS Metadata Server should have a unique metadata identity within that server.

model function
the type of statistical model, such as classification, prediction, or segmentation.

model scoring (scoring)
the process of applying a model to new data in order to compute outputs.

package
See SAS package.

participant
a user, group, or role that is assigned to a task. These users, groups, and roles are defined in SAS metadata and are mapped to standard roles for the workflow.

PMML
See Predictive Modeling Markup Language.

policy
a workflow element that associates event-driven logic with a task or subflow. Policies are usually triggered automatically by an event such as a status change or a timer event.

prediction model
a model that predicts the outcome of an interval target.

Predictive Modeling Markup Language (PMML)
an XML based standard for representing data mining results for scoring purposes. It enables the sharing and deployment of data mining results between applications and across data management systems.

project
a collection of models, SAS programs, data tables, scoring tests, performance data, and reporting documents.

rule
See business rule.

rule flow
a logical collection of multiple rule sets that define multiple conditions and actions. Rule flows can be tested and deployed as SAS programs and services that process input data, which contain conditions, in order to create output data, which contain actions.

rule set
a logical group of business rules.

SAS code model
a SAS program or a DATA step fragment that computes output values from input values. An example of a SAS code model is the LOGISTIC procedure.

SAS Content Server
a server that stores digital content (such as documents, reports, and images) that is created and used by SAS client applications. To interact with the server, clients use WebDAV-based protocols for access, versioning, collaboration, security, and searching.

SAS Metadata Repository
a container for metadata that is managed by the SAS Metadata Server. *See also* SAS Metadata Server.

SAS Metadata Server
a multi-user server that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories.

SAS package (package)
a container for data that has been generated or collected for delivery to consumers by the SAS Publishing Framework. Packages can contain SAS files, binary files, HTML files, URLs, text files, viewer files, and metadata.

scoring
See model scoring.

scoring function
a user-defined function that is created by the SAS Scoring Accelerator from a scoring model and that is deployed inside the database.

scoring test
a workflow that executes a model's score code.

segmentation model
a model that identifies and forms segments, or clusters, of individual observations that are associated with an attribute of interest.

source
See data source.

swimlane
a workflow diagram element that enables you to group tasks that are assigned to the same participant.

task
See workflow task.

task status
the outcome of a task in a workflow. The status of a task (for example, Started, Canceled, Approved) is typically used to trigger the next task.

term
an attribute of a business entity. Terms might or might not have a list of valid values. For example, a customer entity might have terms such as account type or age. Valid values for the account type term might include "commercial" or "personal."

universally unique identifier (UUID)
a number that is used to uniquely identify information in distributed systems without significant central coordination. There are 32 hexadecimal characters in a UUID, and these are divided into five groups with hyphens between them as follows: 8-4-4-4-12. Altogether the 16-byte (128-bit) canonical UUID has 36 characters (32 alphanumeric characters and 4 hyphens). For example: 123e4567-e89b-12d3-a456-426655440000

UUID
See universally unique identifier.

vocabulary
the set of business entities that define your business domain.

WebDAV server
an HTTP server that supports the collaborative authoring of documents that are located on the server. The server supports the locking of documents, so that multiple authors cannot make changes to a document at the same time. It also associates metadata with documents in order to facilitate searching. The SAS business intelligence applications use this type of server primarily as a report repository. Common WebDAV servers include the Apache HTTP Server (with its WebDAV modules enabled), Xythos Software's WebFile Server, and Microsoft Corporation's Internet Information Server (IIS).

workflow
a series of tasks, together with the participants and the logic that is required to execute the tasks. A workflow includes policies, status values, and data objects.

workflow definition
a workflow template that has been uploaded to the server and activated. Workflow definitions are used by the SAS Workflow Engine to create new workflow instances.

workflow instance
a workflow that is running in the SAS Workflow Engine. After a workflow template is uploaded to the server and activated, client applications can use the template to create and run a new copy of the workflow definition. Each new copy is a workflow instance.

workflow task (task)
a workflow element that associates executable logic with an event such as a status change or timer event.

workflow template
a model of a workflow that has been saved to an XML file.