THE POWER TO KNOW.

SAS® Model Manager 2.2
Tutorials

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2009. *SAS® Model Manager 2.2: Tutorials*. Cary, NC: SAS Institute Inc.

SAS® Model Manager 2.2: Tutorials

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, June 2009

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

Contents

About These Tutorials

Audience

SAS Model Manager is designed for the following users:

- Those who are responsible for developing analytical models.
- Those who are responsible for modeling project management.
- Those who are responsible for model validation and performance testing.
- Scoring officers.
- Analysts.
- SAS administrators and SAS Model Manager Administrators.

Conventions Used in This Document

The following typographical conventions are used for all text in this document except for syntax:

bold
identifies an item in the SAS Model Manager window, a menu item, or a computer pathname.

`bold monospace`
identifies text that you enter in a SAS Model Manager window.

italics
identifies a book title or a value that is supplied by the user.

`monospace`
identifies SAS code.

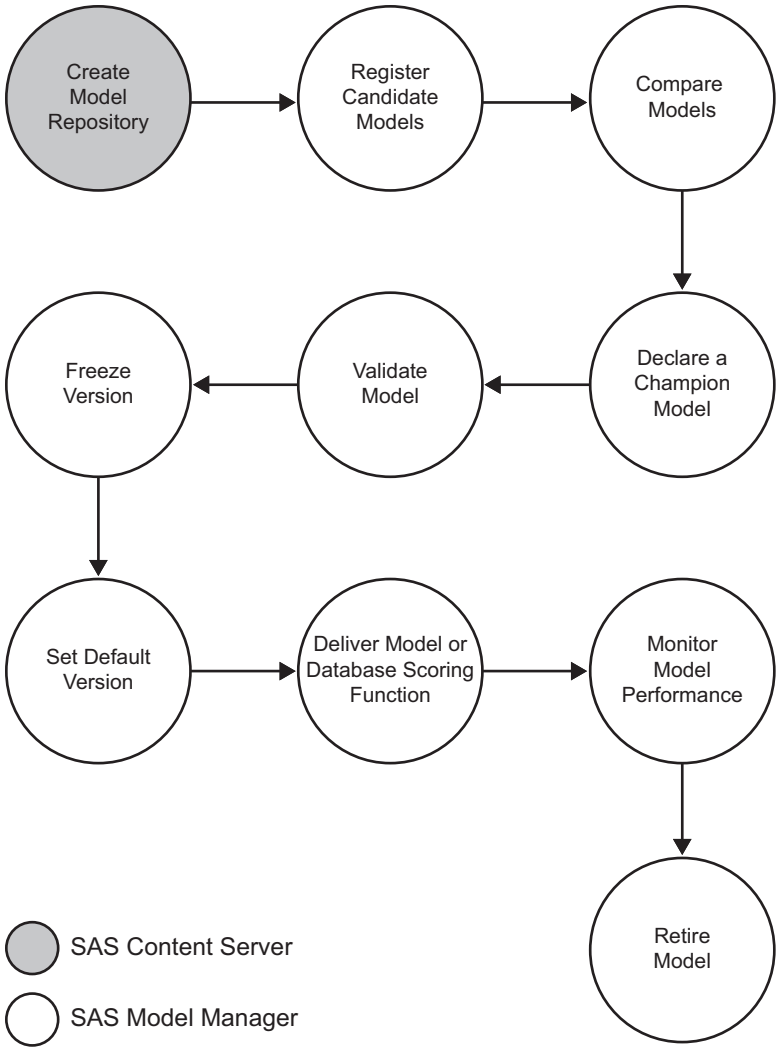
Chapter 1

SAS Model Manager Tutorials

About SAS Model Manager

SAS Model Manager is a flexible model repository and model management environment for predictive and analytical models. A centralized repository and procedural templates make it easier to manage models and metadata across organizational areas and throughout a model's life cycle. Accountability metrics and validation of analytical steps, from creation to deployment in real-time or batch scoring systems, continue until a model is retired. Storing the models in a secure, centralized repository enables you to easily manage life cycle milestones, such as development, test, production, and retirement. The following figure illustrates the model management process that you use in SAS Model Manager:

Figure 1.1 *An Example of the Model Management Process*



The goal of a modeling project is to identify a champion model that a scoring application uses to predict an outcome. SAS Model 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. You can perform scoring tests for champion and challenger model assessment as well as publish and share the model life cycle and performance data over established reporting channels. You can also run comparative performance benchmarks for the models in your production environment.

About SAS Model Manager Tutorials

The tutorials for SAS Model Manager cover basic and advanced tasks that are related to model management within an enterprise computing environment. Tutorial folders are created by extracting files from the tutorial ZIP file. You use these data files to become familiar with the following basic tasks that are involved in model management:

- define user groups for assigning and approving life cycle tasks
- define and create the components of the model hierarchy

- import models
- run model reports
- monitor event logging
- register models and select a champion model
- update life cycle milestones
- run model scoring code in SAS Model Manager

The Tutorial Files

The ZIP file SMM22Tutorial.zip contains the tutorial's data sets, models, and score code, and is available from the SAS Model Manager 2.2 documentation location. Before you begin any of the tutorials, extract the tutorial files to your local computer. Follow the steps for using WinZip to extract the files. If you are using a different extraction program, follow that program's instructions for extracting the files.

1. Create a folder on your local computer to store the tutorial files. The instructions refer to this folder as **<drive>**.
2. From the SAS Model Manager 2.2 documentation location, save SMM22Tutorial.zip to **<drive>**.
3. Open Windows Explorer to **<drive>**. Right-click **SMM22Tutorial.zip** and select **Open with WinZip**.
4. From the WinZip window, click the Extract button. The Extract dialog box appears.
5. In the **Extract to** box, select **<drive>** and click **Extract**.

   You can find the files for each tutorial in the respective tutorial folder (for example, **<drive>**\Tutorial2 or **<drive>**\Tutorial3.)

Chapter 2

Tutorial 1: Create a Life Cycle Template

Overview of Life Cycle Templates and Roles

A SAS Model Manager project consists of one or more versions. A version is a time-based container for SAS Model Manager projects. For example, you might have versions for both 2009 and 2010. The 2009 version contains the champion home equity model and the 2010 version contains the resources for developing a new home equity model.

Each version has a life cycle that is associated with it to track the progress of selecting a champion model and monitoring the model's performance. The life cycle contains milestones such as development, test, and production. Associated with each milestone are tasks that you perform to complete a milestone. When you create a version, you select a life cycle template that you want to use for the version life cycle. The life cycle template for your version must be available to SAS Model Manager before you create a version.

You use the SAS Model Manager Template Editor to create a life cycle template with milestones and tasks that are specific to your modeling project. The life cycle template is saved as an XML file. An alternative method of creating life cycle templates is to use a text editor to add the appropriate XML elements and values for those elements.

After you create the template, the life cycle templates must be saved to the user template directory on the middle-tier server where the SAS Analytics Platform is installed. Any user who has Write access to the user template directory can save the template. The SAS Analytics Platform server must then be restarted.

In SAS Model Manager, you can view life cycle templates from the Life Cycle perspective. Any user-defined template in the Life Cycle perspective can be used as a life cycle when

you create a version. The life cycle templates that SAS supplies are to be used only as examples.

Any users or groups who need to update the version life cycle status must be assigned the appropriate life cycle roles using the SAS Management Console User Manager plug-in:

- Model Manager: Life Cycle Participant Usage (participant)
- Model Manager: Life Cycle Assignee Usage (assignee)
- Model Manager: Life Cycle Approval Usage (approver)

A best practice is to assign these roles only to groups and not to users. Assigning roles to groups provides flexibility when you need to add or remove users who are responsible for life cycle tasks. However, because you can assign these roles to users as well as groups, this tutorial states that both users and groups can be assigned to tasks.

When you open the SAS Model Manager Template Editor, users or groups that are assigned to the participant role appear in the **Participants** list. Only those users and groups in the **Participants** list can be considered to be assignees or approvers. When the template is selected as the life cycle for a version, only those users or groups can update the life cycle folder.

This tutorial creates a Model Manager Tutorial Users group by using SAS Management Console. After you create the Model Manager Tutorial Users group, you create a life cycle template that can be used for the SAS Model Manager tutorials.

Create Groups for Use with the SAS Model Manager Tutorial

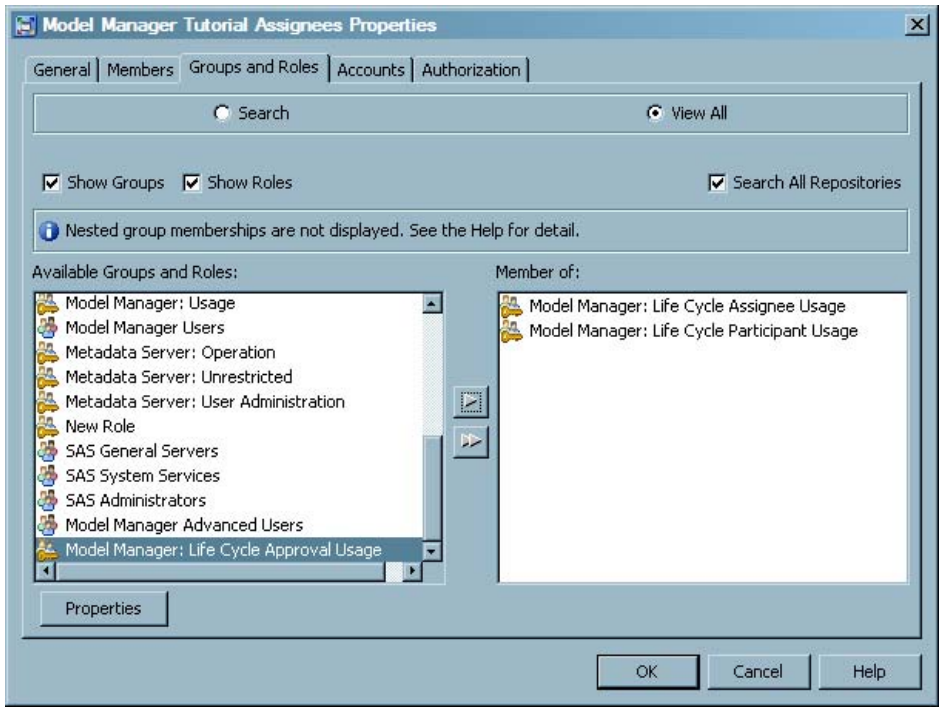
Create a SAS Model Manager Assignee Group

In this exercise, a SAS administrator creates a group in SAS Management Console for SAS Model Manager assignees. Any member of this group will be able to update the status of a life cycle task if that group is specified as a value for the task **Assignee** property.

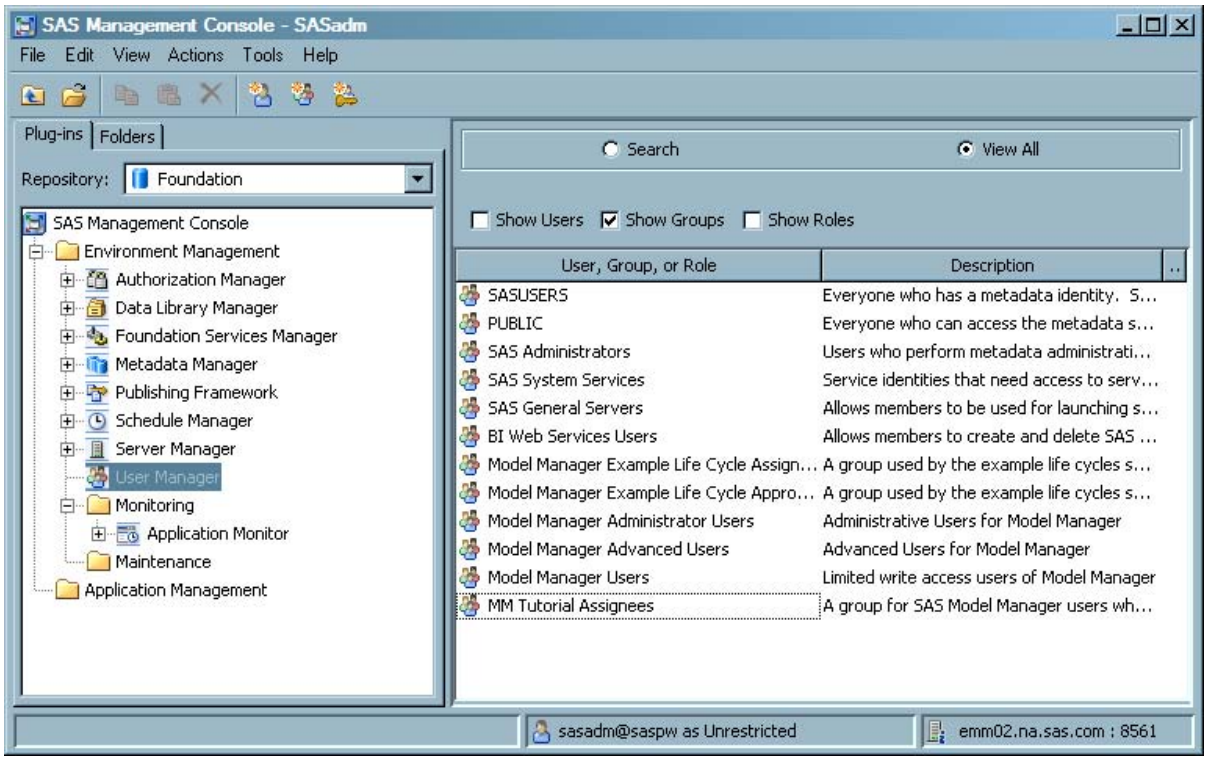
1. Start and log on to SAS Management Console as a SAS Administrator who has a role that enables you to update the metadata server user administration.
2. On the **Plug-ins** tab, right-click **User Manager** and select **New** ⇨ **Group**. The New Group Properties window opens to the **General** tab.
3. In the **Name** field, type **`MM Tutorial Assignees`**.
4. In the **Display Name** field, type **`MM Tutorial Assignees`**.
5. In the **Description** field, type **`A group for SAS Model Manager users who can be assigned to complete tasks.`**
6. Click the **Members** tab.
7. From the **Available Identities** list, select SAS Model Manager advanced users and SAS Model Manager administrators to add to this group. For each user who needs to be assigned to the group, select the user and click  to move the user to the **Current Members** list.
8. Click the **Groups and Roles** tab. Ensure that the **Show Roles** box is checked. Select the following roles and click  to move the roles to the **Member of** list:

- **Model Manager: Life Cycle Participant Usage**
- **Model Manager: Life Cycle Assignee Usage**

Here is an example of the **Groups and Roles** tab.



9. Click **OK**. Here is an example of the group **MM Tutorial Assignees** listed as a group in SAS Management Console.

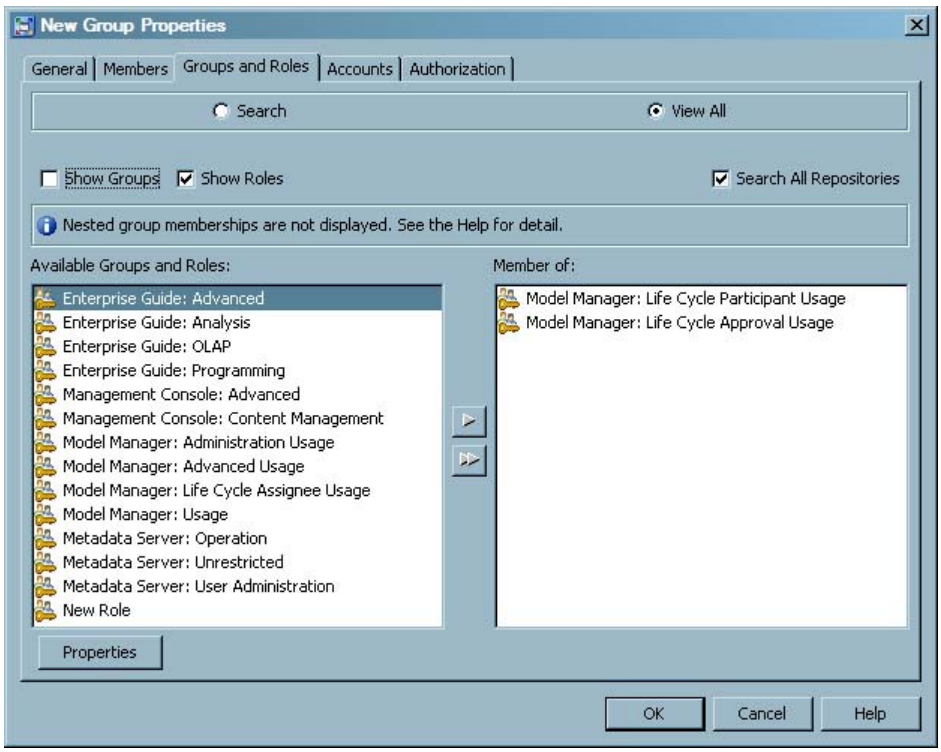


Create a SAS Model Manager Approver Group

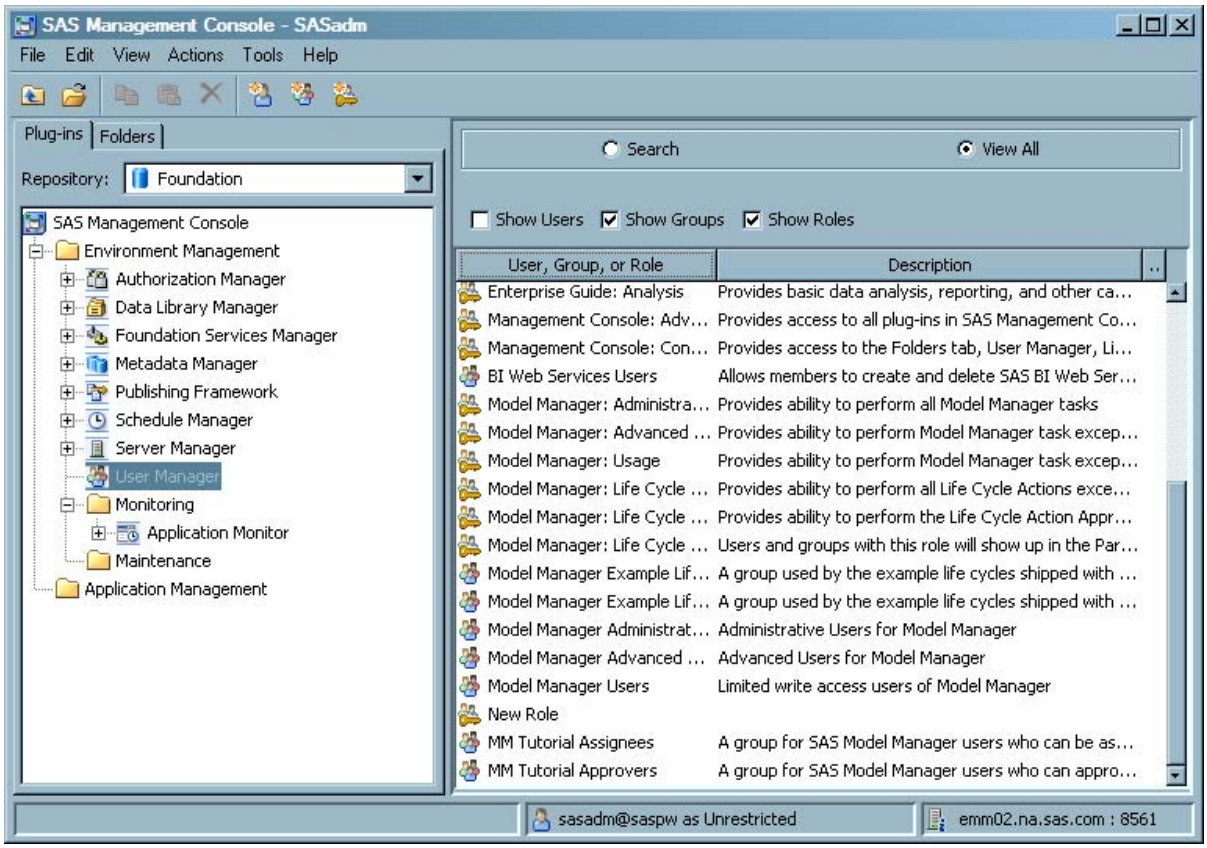
In this exercise, a SAS administrator creates a group in SAS Management Console for SAS Model Manager approvers. Any member of this group will be able to update the **Approved** status of a life cycle task if that group is specified as a value for the task **Approver** property.

1. Start and log on to SAS Management Console as a SAS Administrator who has a role that enables you to update the metadata server user administration.
2. On the **Plug-ins** tab, right-click **User Manager** and select **New** ⇨ **Group**. The New Group Properties window opens to the **General** tab.
3. In the **Name** field, type **`MM Tutorial Approvers`**.
4. In the **Display Name** field, type **`MM Tutorial Approvers`**.
5. In the **Description** field, type **`A group for SAS Model Manager users who can approve that a task is complete.`**
6. Click the **Members** tab.
7. From the **Available Identities** list, select SAS Model Manager advanced users and SAS Model Manager administrators to add to this group. For each user who needs to be assigned to the group, select the user and click  to move the user to the **Current Members** list.
8. Click the **Groups and Roles** tab. Ensure that the **Show Roles** box is checked. Select the following roles and click  to move the roles to the **Member of** list:
   - **Model Manager: Life Cycle Participant Usage**
   - **Model Manager: Life Cycle Approval Usage**

   Here is an example of the **Groups and Roles** tab.



9. Click **OK**. Here is an example of the group **MM Tutorial Approvers** listed as a group in SAS Management Console.



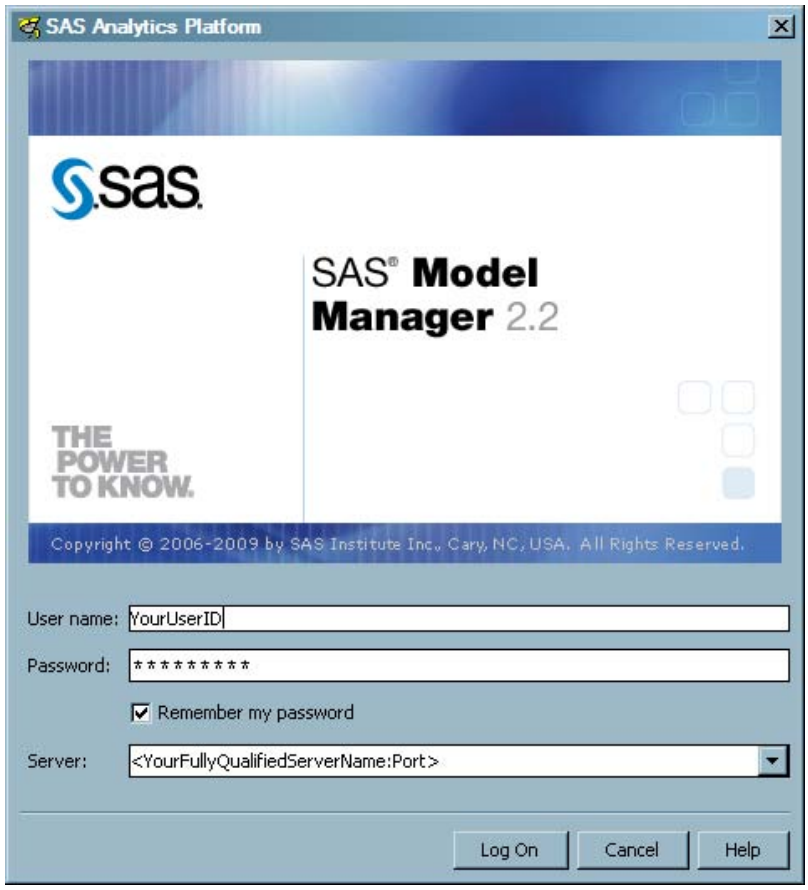
Create a Life Cycle Template

In this exercise, you use the SAS Model Manager Template Editor to create a user-defined life cycle template.

Start SAS Model Manager

To run SAS Model Manager, follow these steps:

1. On your client machine, start the SAS Model Manager client.
2. Log on to SAS Model Manager as a member of the **Model Manager Advanced Users** group.



Create a New Life Cycle Template

This task creates and names a new life cycle template.

1. Open the SAS Model Manager Template Editor (Template Editor). Select **File** ⇨ **Manage Templates** in the SAS Model Manager window.
2. In the Template Editor, open a new life cycle template. Select **File** ⇨ **New Life Cycle Template**.
3. Modify the template properties. Specify the following options:

   Name
   enter **`Tutorial Life Cycle`**. The name of the template appears in the Life Cycle perspective. This option is required.

   Description
   enter **`A life cycle for the tutorial.`**

   Version
   use the default version, **1**.

   TIP A best practice is to increment the template version by 1 each time you modify the template.

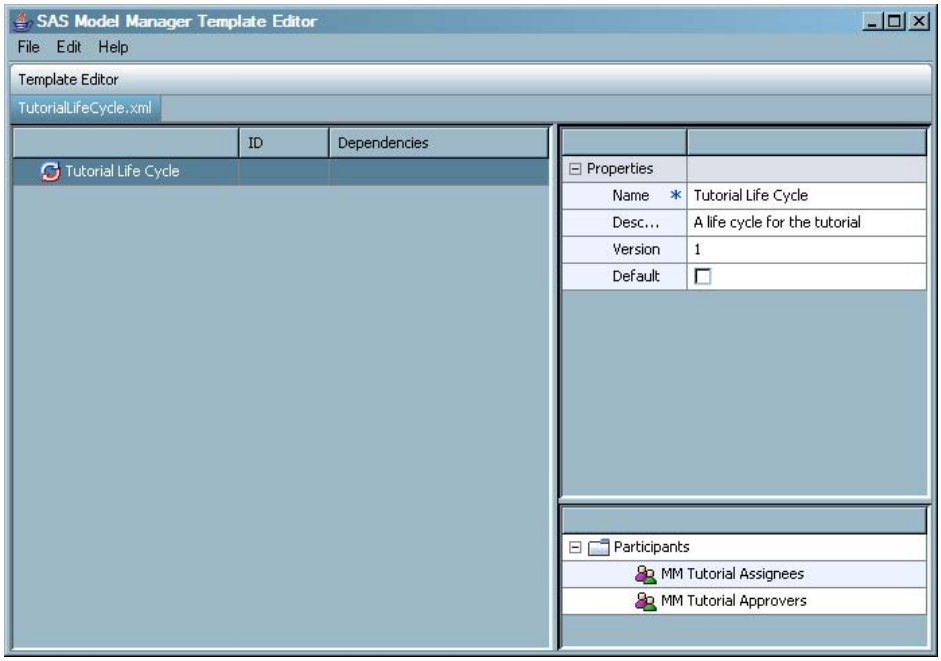
4. Save the template to your local computer by selecting **File** ⇨ **Save As**. In the Save dialog box, select the location on your local computer. Enter **`TutorialLifeCycle.xml`** as the filename and click **Save**. Click **OK** in the information dialog box to save the template as an XML file.

5. Using a text editor, open the life cycle template XML file that you just saved. Remove all participants except the MM Tutorial Assignees group and the MM Tutorial Approvers group. The participants are enclosed in <Participants> </Participants> tags. Here are the final participants in the XML file:

```
<Participants>
   <Participant id="A5W6VW1T.A3000000W"
                name="MM Tutorial Assignees">
   </Participant>
   <Participant id="A5W6VW1T.A3000000X"
                name="MM Tutorial Approvers">
   </Participant>
</Participants>
```

   Save the file.

6. In the SAS Model Manager Template Editor, select **File** ⇨ **Open**. In the Open dialog box, select the template and click **Open**. The **Participants** list displays only **MM Tutorial Assignees** and **MM Tutorial Approvers**.



Add Milestones to a User-defined Template

This exercise adds three milestones to the life cycle template: Develop, Test, and Production.

1. Add the Develop milestone. Right-click **Tutorial Life Cycle** and select **New Milestone**. In the New Milestone window, complete the **Name** and **Type** fields and click **OK**.

   Name
   enter `Develop`. This field is required.

   Type
   use the default milestone type, **Develop**.

   After you click **OK**, the **Develop** milestone has an ID of 1.

2. Add the **Test** milestone. Right-click **Tutorial Life Cycle** and select **New Milestone**. In the New Milestone window, complete the **Name** and **Type** fields and click **OK**.

   Name
   enter **`Test`**.

   Type
   click the **Type** box and select **Test**.

   After you click **OK**, the **Test** milestone has an ID of 2.

3. Add the **Production** milestone. Right-click **Tutorial Life Cycle** and select **New Milestone**. In the New Milestone window, complete the **Name** and **Type** fields and click **OK**.

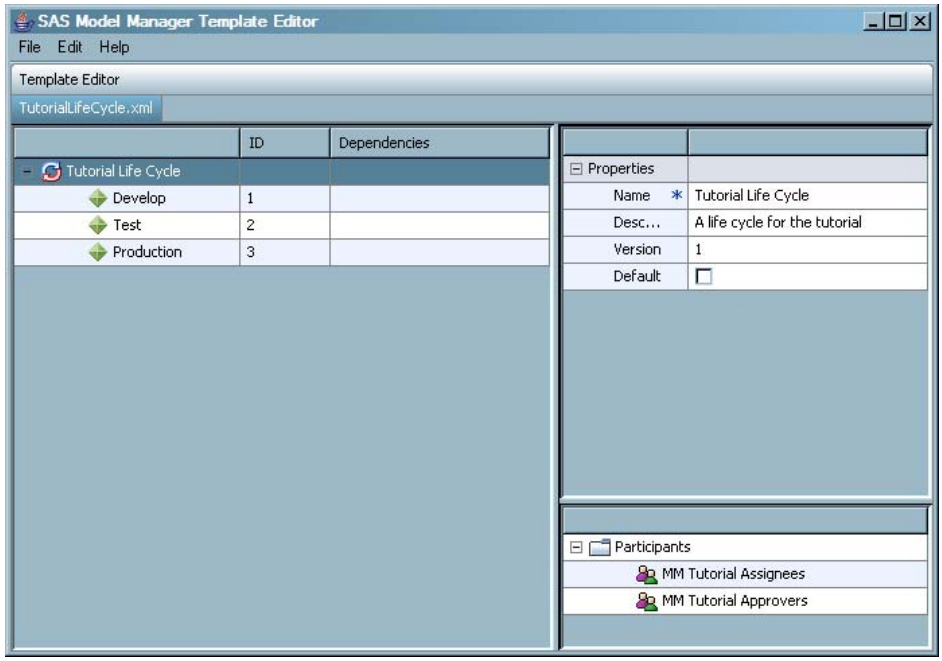
   Name
   enter **`Production`**.

   Type
   click the **Type** box and select **Production**.

   After you click **OK**, the **Production** milestone has an ID of 3.

4. Select **File** ⇨ **Save** to save the template. Click **OK** when the Warning dialog box appears.

Here is the template at the end of this exercise:



Add Tasks to the Life Cycle Template Milestones

This exercise adds tasks to each milestone.

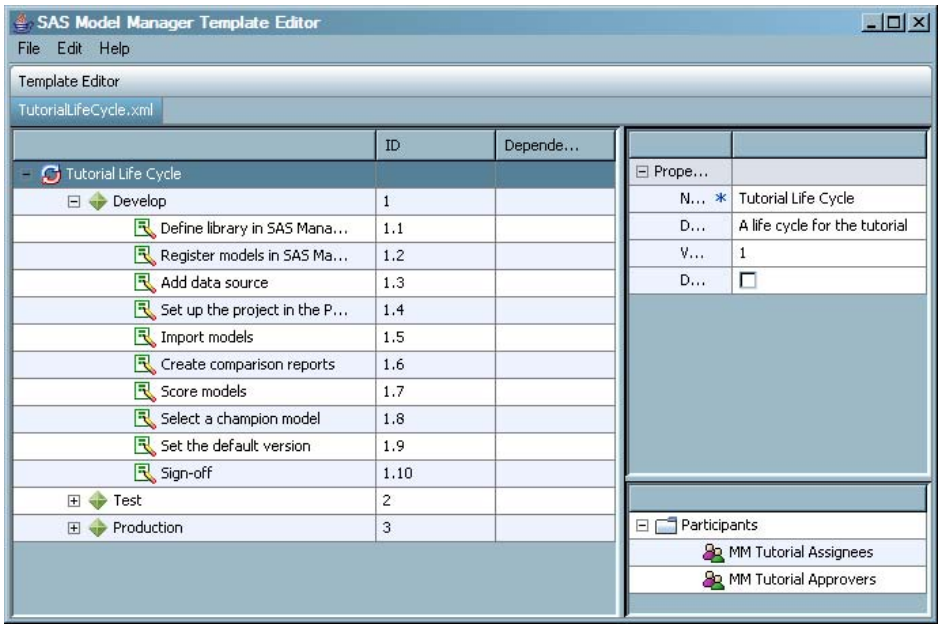
1. Add tasks to the **Develop** milestone.

   For each task, right-click the **Develop** milestone and select **New Task**. In the New Task window, complete the **Name** field and **Type** field using the following table. Click **OK**. The task names are descriptive. Therefore, a description is not necessary.

Task Name	Task Type
`Define library in SAS Management Console`	**UserDefined**
`Register models`	**UserDefined**
`Add data source`	**UserDefined**
`Set up the project in the Project Tree`	**UserDefined**
`Import models`	**UserDefined**
`Create comparison reports`	**UserDefined**
`Score models`	**UserDefined**
`Select a champion model`	**SetChampion**
`Set the default version`	**UserDefined**
`Sign-off`	**Signoff**

Expand the Develop milestone. Each task has an ID in the form **`milestone.task`**. The first number in the ID is the milestone ID. The second number in the ID identifies the specific task.

Here is the template after adding the tasks for the **Develop** milestone:



2. Add tasks to the **Test** milestone.

   For each task, right-click the **Test** milestone and select **New Task**. In the New Task window, complete the **Name** field and **Type** field using the following table. Click **OK**. The task names are descriptive. Therefore, a description is not necessary.

Task Name	Task Type
`Validate score input data`	UserDefined
`Validate score output data`	UserDefined
`Test scoring`	UserDefined
`Sign-off`	Signoff

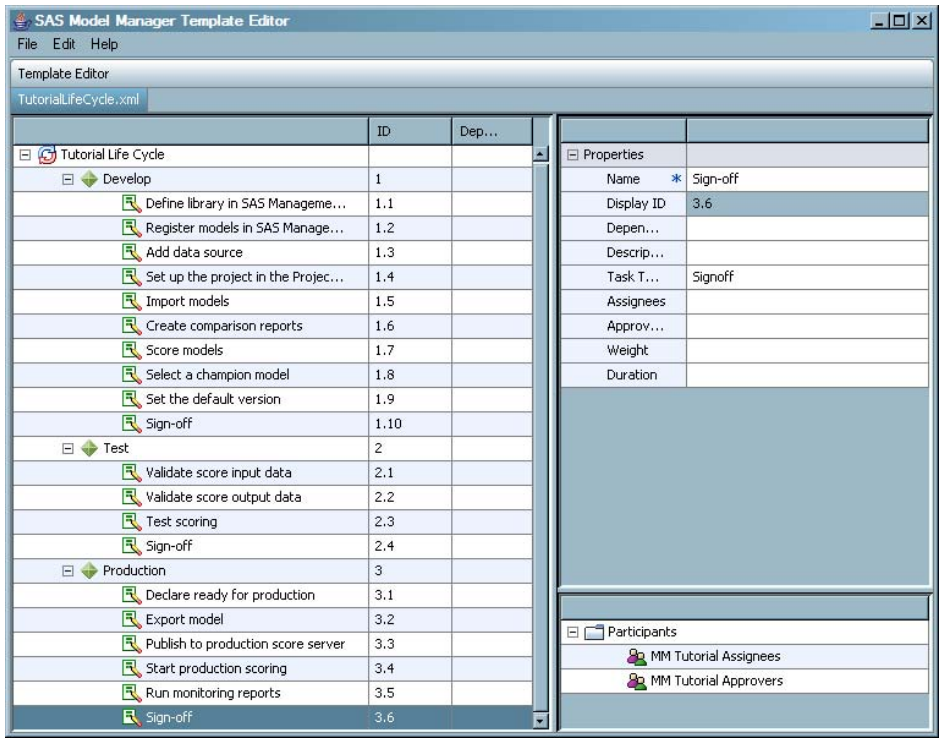
3. Add tasks to the **Production** milestone.

   For each task listed below, right-click the **Production** milestone and select **New Task**. In the New Task window, complete the **Name** field and **Type** field using the following table. Click **OK**. The task names are descriptive. Therefore, a description is not necessary.

Task Name	Task Type
`Declare ready for production`	**DeclareProduction**
`Export model`	**UserDefined**
`Publish to production score server`	**UserDefined**
`Start production scoring`	**UserDefined**
`Run monitoring reports`	**UserDefined**
`Sign-off`	**Signoff**

4. Select **File** ⇨ **Save** to save the template. Click **OK** when the Warning dialog box appears.

Here is the template after all milestone tasks have been defined:



Add Task Dependencies

About Dependencies

Your model development process might require an order to complete some tasks. You can assign dependencies to a task that indicates that one or more tasks must be completed before another one can be marked complete. You specify dependencies in the **Dependencies** property for a task.

This exercise assigns dependencies to milestone tasks.

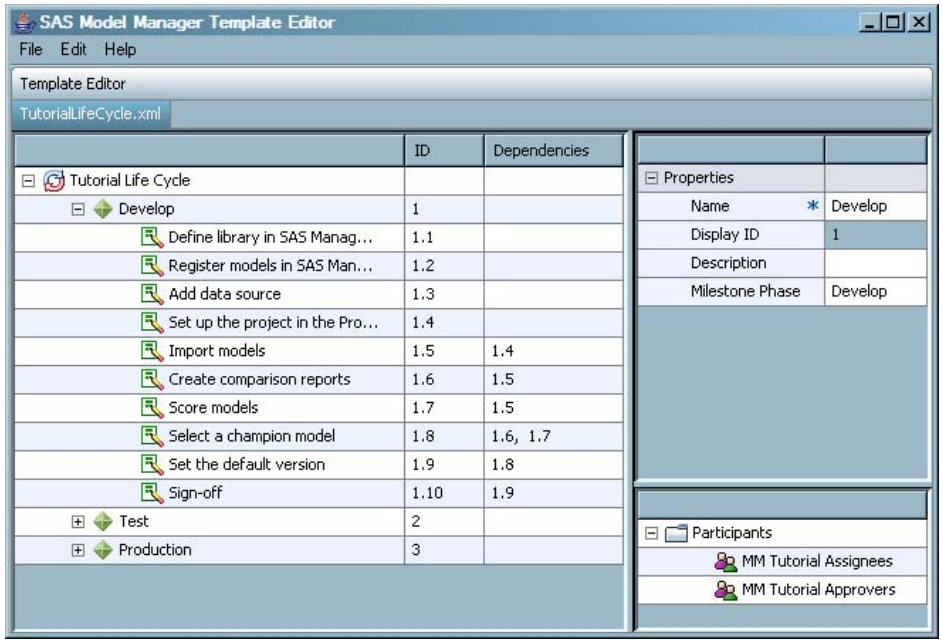
Add Dependencies for the Develop Milestone Tasks

To assign the **Develop** milestone task dependencies, follow these steps:

1. Select the **Import models** task. Click the **Dependencies** property value field and then click the ellipsis button. Select the box for **Set up the project in the Project Tree**. Click **OK**. The **Import models** task now has a dependency on task 1.4, **Set up project in the Project Tree**.
2. Select the **Create comparison reports** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Import models**. Click **OK**. The **Create comparison reports** task now has a dependency on task 1.5, **Import models**.
3. Select the **Score models** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Import models**. Click **OK**. The **Score models** task now has a dependency on task 1.5, **Import models**.
4. Select the **Select a champion model** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Create comparison reports** and **Score models**. Click **OK**. The **Select a champion model** task now has a dependency on task 1.6, **Create comparison reports** and task 1.7, **Score models**.

5. Select the **Set the default version** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Select a champion model**. Click **OK**. The **Set the default version** task now has a dependency on task 1.8, **Select a champion model**.
6. Select the **Sign-off** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Set the default version**. Click **OK**. The **Sign-off** task now has a dependency on task 1.9, **Set the default version**.
7. Click any field to refresh the template editor.

   Here is the template after the **Develop** milestone task dependencies have been assigned:



Add Dependencies for the Test Milestone Tasks

To assign the **Test** milestone task dependencies, follow these steps:

1. Select the **Test scoring** task. Click the **Dependencies** property value field and then click the ellipsis button. Select the box for **Validate score input data** and **Validate score output data**. Click **OK**. The **Test scoring** task now has a dependency on task 2.1, **Validate score input data**, and task 2.2, **Validate score output data**.
2. Select the **Sign-off** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Test scoring**. Click **OK**. The **Sign-off** task now has a dependency on task 2.3, **Test scoring**.
3. Click any field to refresh the template editor.

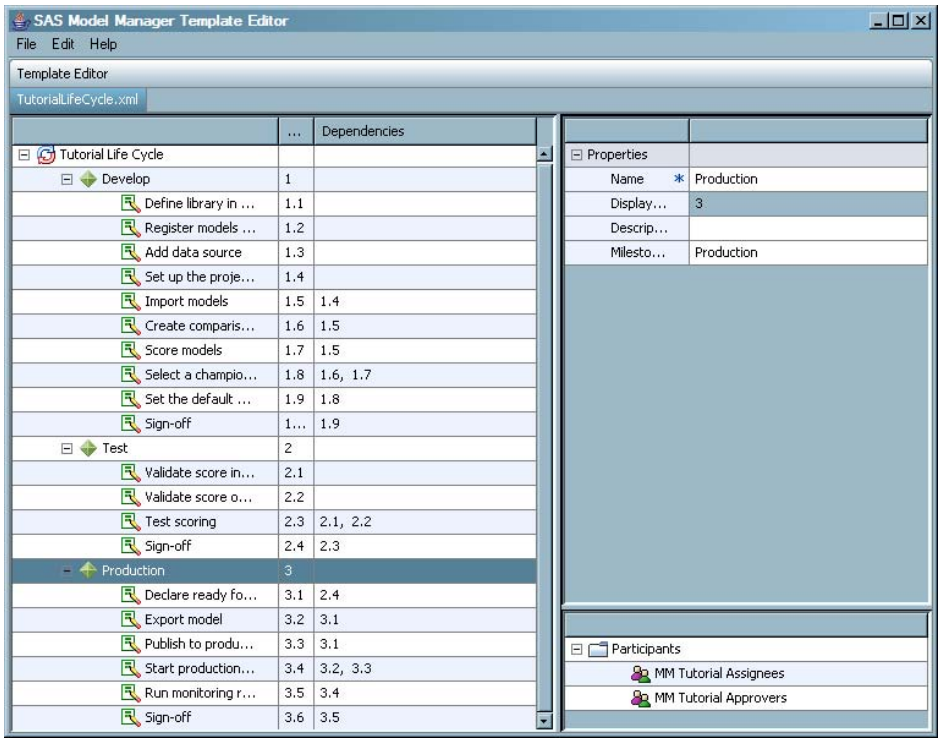
Add Dependencies for the Production Milestone Tasks

To assign the **Production** milestone task dependencies, follow these steps:

1. Select the **Declare ready for production** task. Click the **Dependencies** property value field and then click the ellipsis button. Select the box for **Sign-off** for **ID 2.4** . Click **OK**.
2. Select the **Export model** task. Click the **Dependencies** property value field and then click the ellipsis button. Select the box for **Declare ready for production**. Click **OK**. The **Export model** task now has a dependency on task 3.1, **Declare ready for production**.

3. Select the **Publish to production score server** task. Click the **Dependencies** property value field and then click the ellipsis button. Select the box for **Declare ready for production**. Click **OK**. The **Publish to production score server** task now has a dependency on task 3.1, **Declare ready for production**.
4. Select the **Start production scoring** task. Click the **Dependencies** property value field and then click the ellipsis button. Select the box for **Export model** and **Publish to production score server**. Click **OK**. The **Start production scoring** task now has a dependency on task 3.2, **Export model**, and task 3.3, **Publish to production score server**.
5. Select the **Run monitoring reports** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Start production scoring**. Click **OK**. The **Run monitoring reports** task now has a dependency on task 3.4, **Start production scoring**.
6. Select the **Sign-off** task. Click the **Dependencies** property and then click the ellipsis button. Select the box for **Run monitoring reports**. Click **OK**. The **Sign-off** task now has a dependency on task 3.5, **Run monitoring reports**.
7. Click any field to refresh the template editor.

   Here is the template after all dependencies have been assigned:



8. To save the template to your local computer, select **File** ⇨ **Save**. Type **`TutorialLifeCycle.xml`** and click **Save**.

Complete Task Properties

Complete the Develop Task Properties

In this exercise you complete the **Develop** task properties.

Select each task and enter property values using the values in the following table.

To assign property values for the **Assignees** and **Approvers** properties, click the ellipsis button to open the Select Participants window. Check the box for the assignee or approver and click **OK**.

Task	Assignees Property	Approvers Property	Weight Property	Duration Property
Define library in SAS Management Console	MM Tutorial Assignees	MM Tutorial Approvers	10	1
Register models in SAS Management Console	MM Tutorial Assignees	MM Tutorial Approvers	10	1
Add data source	MM Tutorial Assignees	MM Tutorial Approvers	10	1
Set up the project in the Project Tree	MM Tutorial Assignees	MM Tutorial Approvers	10	1
Import models	MM Tutorial Assignees	MM Tutorial Approvers	10	1
Create comparison reports	MM Tutorial Assignees	MM Tutorial Approvers	10	2
Score models	MM Tutorial Assignees	MM Tutorial Approvers	10	2
Select a champion model	MM Tutorial Assignees	MM Tutorial Approvers	20	4
Set the default version	MM Tutorial Assignees	MM Tutorial Approvers	10	1
Sign-off	MM Tutorial Assignees	MM Tutorial Approvers	10	1

Save the template.

Complete the Test Task Properties

In this exercise you complete the **Test** task properties.

Select each task and enter property values using the values in the following table.

To assign property values for the **Assignees** and **Approvers** properties, click the ellipsis button to open the Select Participants window. Check the box for the assignee or approver and click **OK**.

Task	Assignees Property	Approvers Property	Weight Property	Duration Property
Validate score input table	MM Tutorial Assignees	MM Tutorial Approvers	25	1
Validate score output table	MM Tutorial Assignees	MM Tutorial Approvers	25	1
Test scoring	MM Tutorial Assignees	MM Tutorial Approvers	40	5
Sign-off	MM Tutorial Assignees	MM Tutorial Approvers	10	1

Save the template.

Complete the Production Task Properties

In this exercise you complete the **Production** task properties.

Select each task and enter property values using the values in the following table.

To assign property values for the **Assignees** and **Approvers** properties, click the ellipsis button to open the Select Participants window. Check the box for the assignee or approver and click **OK**.

Task	Assignees Property	Approvers Property	Weight Property	Duration Property
Declare ready for production	MM Tutorial Assignees	MM Tutorial Approvers	10	1
Export model	MM Tutorial Assignees	MM Tutorial Approvers	20	1
Publish to production scoring server	MM Tutorial Assignees	MM Tutorial Approvers	20	1
Start production scoring	MM Tutorial Assignees	MM Tutorial Approvers	20	2
Run monitoring reports	MM Tutorial Assignees	MM Tutorial Approvers	20	1
Sign-off	MM Tutorial Assignees	MM Tutorial Approvers	10	1

Save the template.

Deploy the Life Cycle Template

In this exercise, you deploy the new life cycle template.

1. Contact a user who has Write permission to the middle-tier server directory ***sasconfigdir*\Lev#\AnalyticsPlatform\apps\ModelManager\ext**.

   sasconfigdir is the path prefix to the SAS configuration directory (for example, **c:\SAS\Config**.)

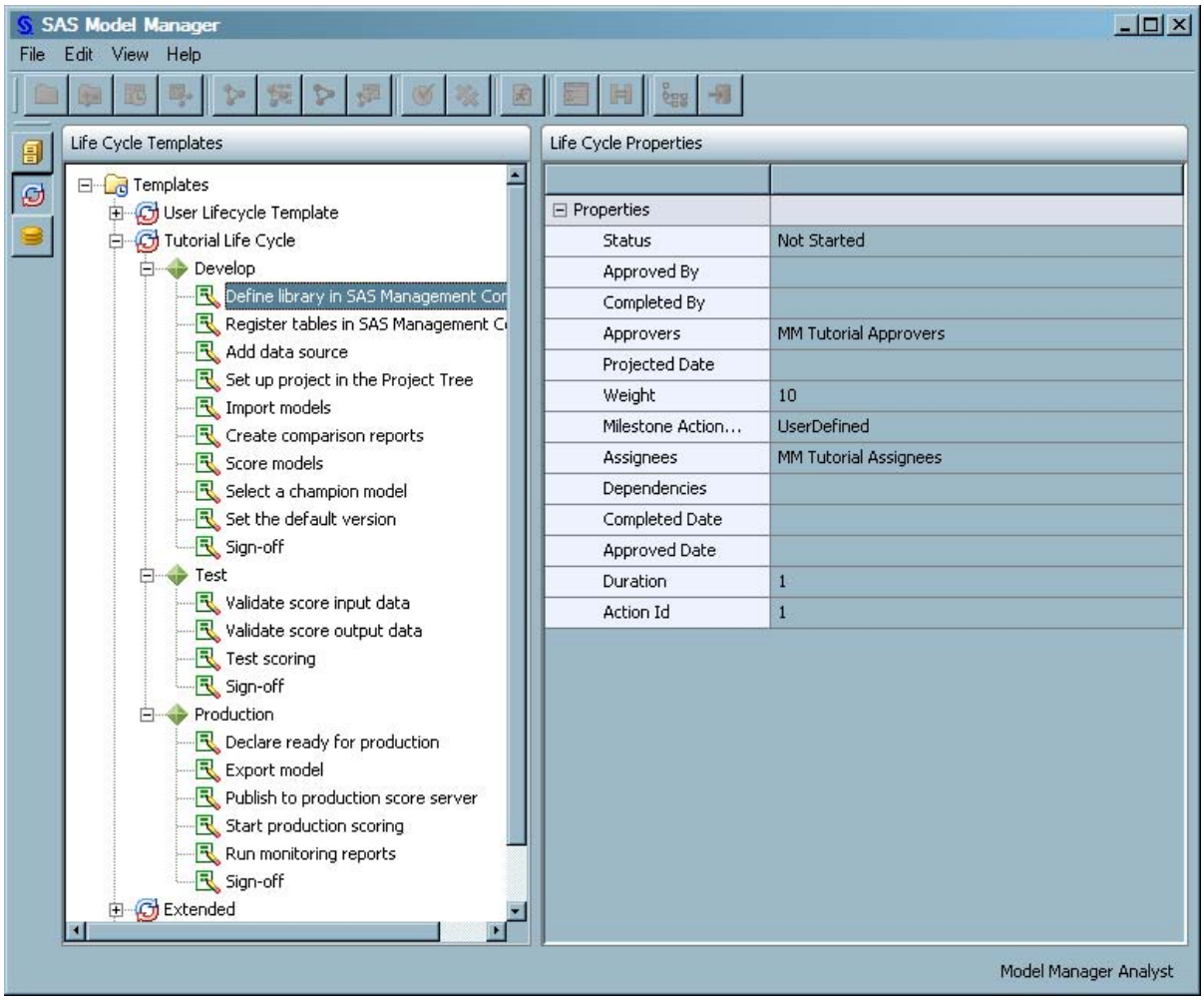
   Lev# is a level number that is set during installation.

   Ask this user to copy the life cycle template TutorialLifeCycle.xml from the local computer to the directory listed above on the middle-tier server.

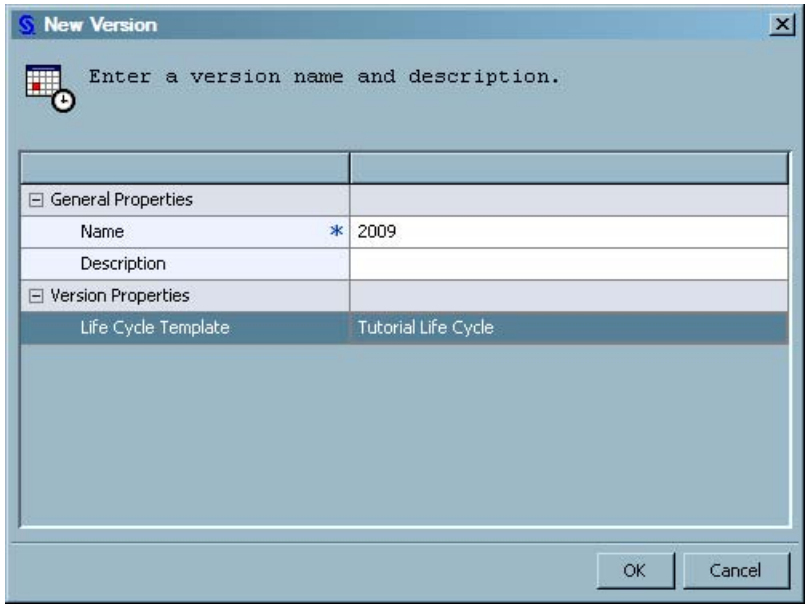
2. Ask your administrator to restart the SAS Analytics Platform.

3. Restart SAS Model Manager. Click the Life Cycle perspective and expand **Tutorial Life Cycle**.

   Here is **Tutorial Life Cycle** in the Life Cycle perspective:



Tutorial Life Cycle can now be specified as a life cycle template for a version, as shown in the New Version window:


New Version
Enter a version name and description.
General Properties
Name
2009
Description
Version Properties
Life Cycle Template
Tutorial Life Cycle
OK
Cancel

Chapter 3

Tutorial 2: Performing Basic SAS Model Manager Tasks

Overview of SAS Model Manager Basics

After an administrator has defined your user ID in SAS Management Console and you have a life cycle template, you are ready to work in SAS Model Manager. This tutorial guides you through a simple modeling project process.

To enable you to track the progress of the modeling project, each task has instructions to update the version life cycle status.

Prepare Tutorial Data Sets and Models

In this exercise you create a file system folder to store the SAS data tables that are used by SAS Model Manager and the SAS Workspace Server.

The Required Tutorial Files

The SAS data sets and models that are required for this tutorial are on your local computer after you extract them from the ZIP file SMM22Tutorial.zip. If you have not extracted the tutorial files, extract them using the instructions in "The Tutorial Files" on page 3.

This tutorial requires the following files and folders in the **<*drive*>\Tutorial2\Samples** folder:

- delinquencyProjectOutput.sas7bdat
- delinquencyProjectInput.sas7bdat
- delinquencyScoringInput.sas7bdat
- delinquencyScoringOutput.sas7bdat
- delinquencyTest.sas7bdat
- delinquencyTrain.sas7bdat
- The **model1** folder contains these model files:
  - modelin1.sas.7bdat
  - modelout1.sas.7bdat
  - om.sas.7bdat
  - result1.sas.7bdat
  - score1.sas.7bdat
  - target1.sas.7bdat
- The **model2** folder contains these model files:
  - modelin2.sas.7bdat
  - modelout2.sas.7bdat
  - ot.sas.7bdat
  - result2.sas.7bdat

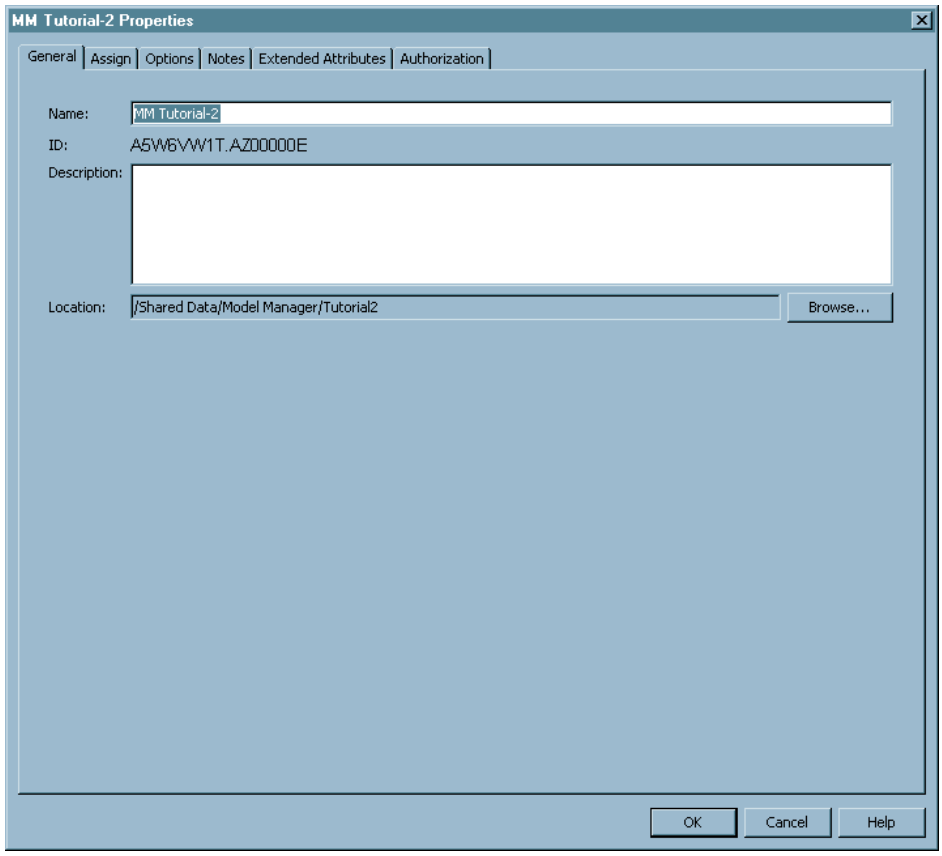
  - score2.sas.7bdat
  - target2.sas.7bdat
- The **model3** folder contains these model files:
  - modelin3.sas.7bdat
  - modelout3.sas.7bdat
  - result3.sas.7bdat
  - score3.sas.7bdat
  - target3.sas.7bdat

Define a Data Library in SAS Management Console

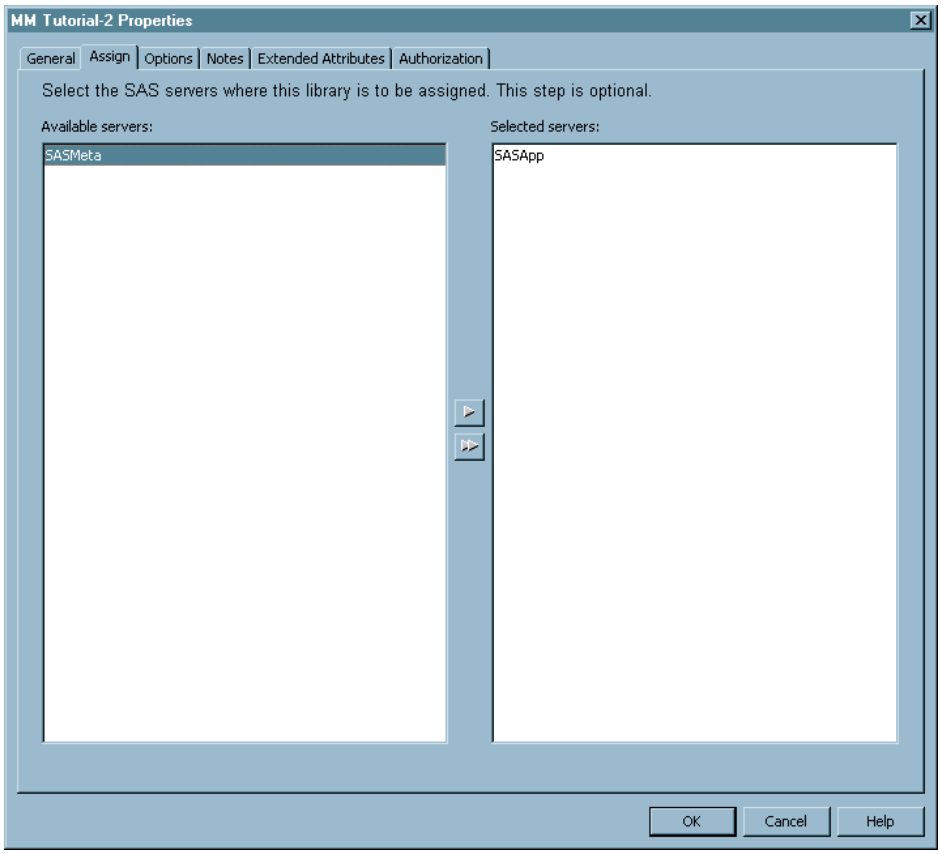
Use SAS Management Console to define a data library and register the tables. Follow these steps:

1. Start and log on to SAS Management Console as a SAS Administrator with the role **Metadata Server: Operation or Metadata Server: Unrestricted**.
2. Open the New Library Wizard to a define the data library. Click **Environment Management** ⇨ **Data Library Manager** on the **Plug-ins** tab. Right-click **Libraries** and select **New Library** from the pop-up menu.
3. In the New Library Wizard, create a SAS library.
   a. Select **SAS BASE Library** and click **Next**.
   b. Specify **`MM Tutorial-2`** in the **Name** field and click **Browse**. In the Select a Location dialog box, double-click **Model Manager** and then click the **New folder** icon. Create the folder **`Tutorial2`** and make **Tutorial2** the active folder. Click **OK**. Click **Next**.
   c. Select **SASApp** from the **Available servers** box and click  to move the server name to the **Selected servers** list. Click **Next**.
   d. Specify **`smm2tor2`** for the libref and click **New**.
   e. In the **Paths** box of the New Path Specification dialog box, enter the tutorial folder that you previously created, **`<drive>\Tutorial2\Samples`**. Click **OK**.
   f. Click **Next** and **Finish**. Verify that the library MM Tutorial-2 is a library in the **Libraries** folder.
4. Register the data tables in the metadata repository.
   a. Right-click **MM Tutorial-2** under the **Libraries** folder, select **Register Tables** from the pop-up menu, and click **Next**.
   b. If prompted, specify a user ID and password that has access to the metadata server and click **OK**.
   c. Click **Select All Tables**, click **Next**, and click **Finish**.
5. Verify that table metadata was created and close SAS Management Console. Right-click **MM Tutorial-2** and select **Properties**.

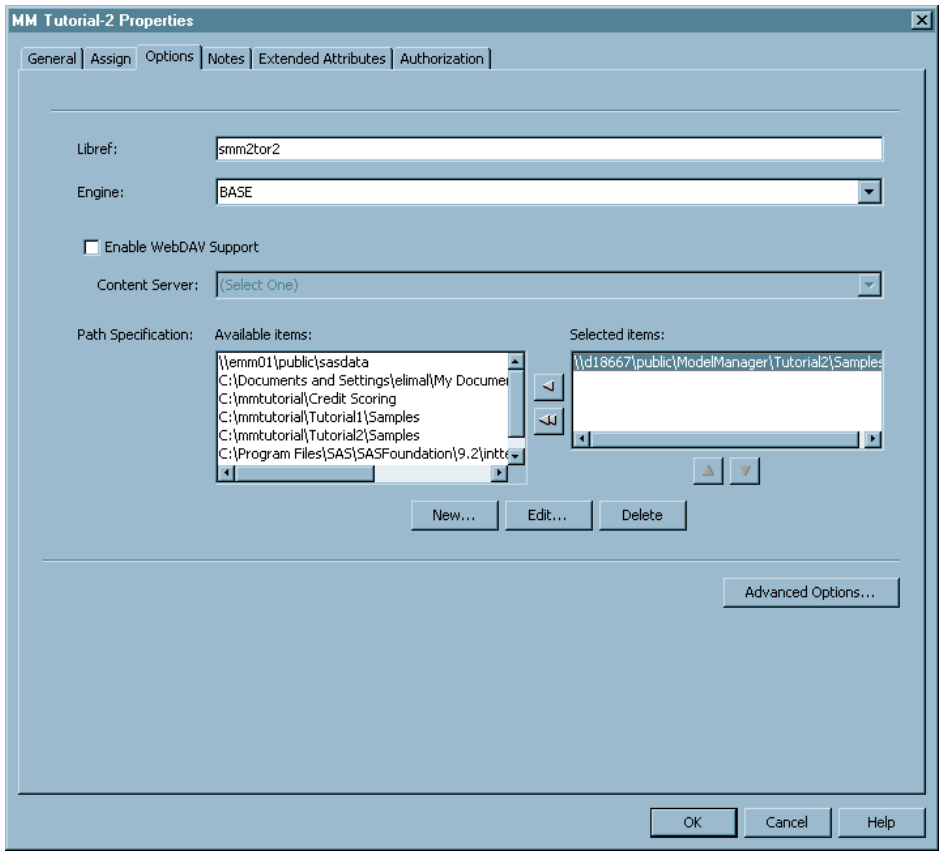
   Verify the tutorial name and location.


MM Tutorial-2 Properties
General
Assign
Options
Notes
Extended Attributes
Authorization
Name:
MM Tutorial-2
ID:
A5W6VW1T.AZ00000E
Description:
Location:
/Shared Data/Model Manager/Tutorial2
Browse...
OK
Cancel
Help

Click the **Assign** tab. Verify that the appropriate server is in the **Selected servers** list.



Select the **Options** tab. Verify the libref, the engine, and the path specification.



Verify Your User ID as a Member of Model Manager User Groups

This exercise ensures that your user ID is a member of the **MM Tutorial Assignees** group and the **Model Manager Advanced Users** group.

1. Open SAS Management Console and log on to the SAS Metadata Server.
2. On the **Plug-ins** tab, select **User Manager**.
3. In the right pane, double-click the **MM Tutorial Assignees** group and click the **Members** tab.
4. Review the **Current Members** list, and locate your user ID or a group that your user ID is a member of. If your user ID or group is not a member of the **MM Tutorial Assignees** group, ask your administrator to add you to this group. Close the properties window.
5. Find and double-click your user ID in the right pane of SAS Management Console.
6. Click the **Groups and Roles** tab. Review the **Member of** pane and locate the group **Model Manager Advanced Users**. If your user ID is not a member of this group, ask your administrator to add you to this group. Close the properties window.
7. Close SAS Management Console.

See Also

"Create Groups for Use with the SAS Model Manager Tutorial" on page 6

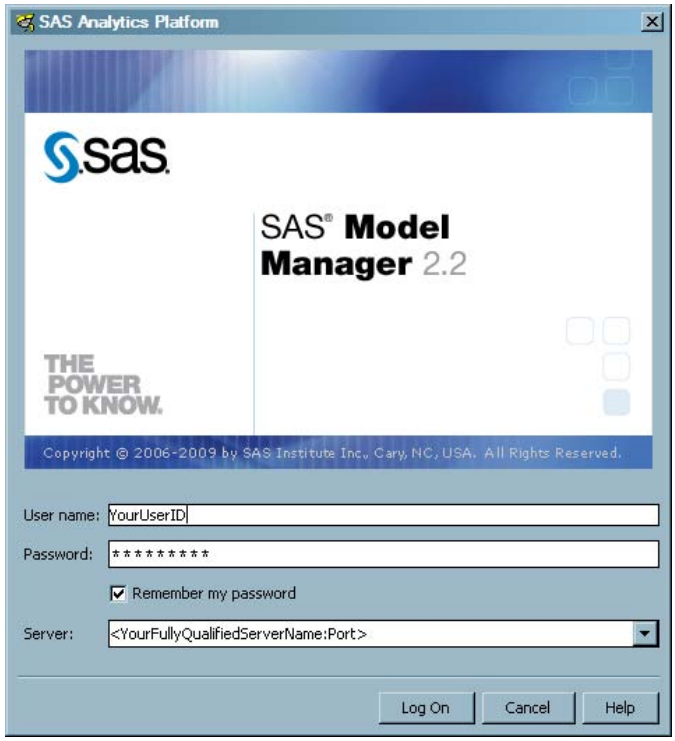
Add Data Sources

In this exercise you use SAS Model Manager to access the data tables that the models or model tasks use.

Start SAS Model Manager

To run SAS Model Manager, follow these steps:

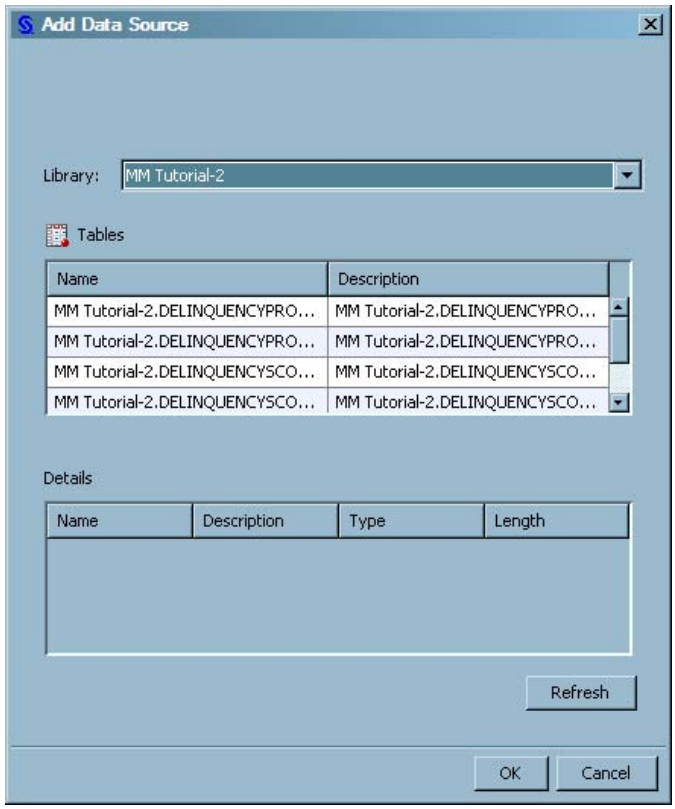
1. On your client machine, start the SAS Model Manager client.
2. Log on to SAS Model Manager. Ensure that your user ID is set up to use the tutorials.



Select Data Sources

Before you begin any project, the tables that you registered in the SAS Metadata Repository must be added as SAS Model Manager data sources using the Data Sources perspective. To add data tables in the Data Source perspective, follow these steps:

1. Click the **Data Sources** perspective button  .
2. Right-click **Project Input Tables** and select **Add Data Source**. The Add Data Source window opens.
3. In the **Library** box, select the library **MM Tutorial-2**. If you do not see the **MM Tutorial-2** library in the list, click **Refresh** and click the **Library** box again.
4. In the **Tables** box, select **MM Tutorial-2.DELINQUENCYPROJECINPUT** and click **OK**.



5. Repeat the above steps to add the remaining tables to the corresponding data source folder:

Folder name	Data Table name
Project Output Tables	**MM Tutorial-2.DELINQUENCYPROJECTOUTPUT**
Test Tables	**MM Tutorial-2.DELINQUENCYTEST**
Scoring Task Input Tables	**MM Tutorial-2.DELINQUENCYSCORINGINPUT**
Scoring Task Output Tables	**MM Tutorial-2.DELINQUENCYSCORINGOUTPUT**
Training Tables	**MM Tutorial-2.DELINQUENCYTRAIN**

Organize the Model Hierarchy

In this exercise you learn to use the Project Tree to create a modeling project.

Create a Folder

To provide an organizational folder to manage your modeling projects, follow these steps:

1. Click the Project perspective button  and right-click **MMRoot** in the Project Tree. Select **New** ⇨ **New Folder**. The New Folder dialog box appears.
2. Specify values for the following folder properties and click **OK**.

   Name
   enter `Tutorial2` for the folder name.

   Description
   enter an optional folder description.

The new folder appears in the Project Tree.

Create a New Project

To create a project and define the model function, follow these steps:

1. Right-click **Tutorial2** and select ⇨ **New** ⇨ **New Project**. The New Project dialog box appears.
2. Specify the following project properties and click **OK**:

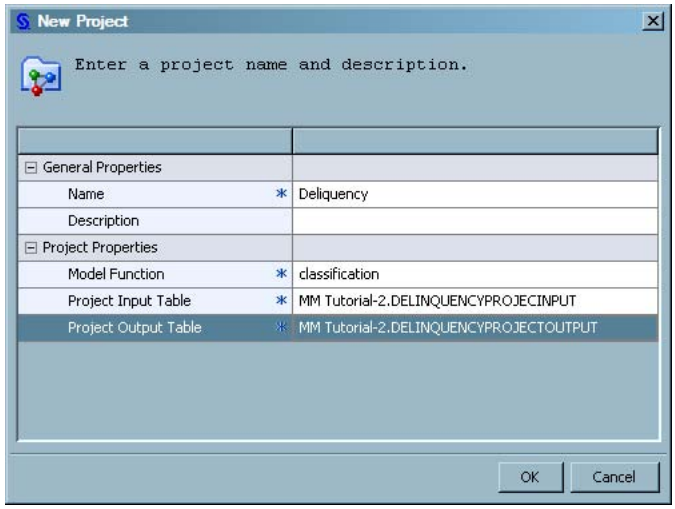
   Name
   enter `Delinquency` for the project name.

   Description
   enter an optional description.

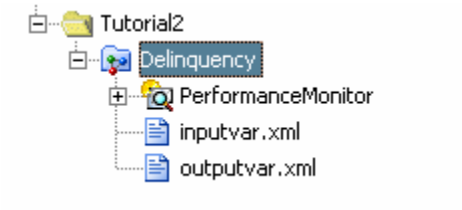
   Model Function
   select **classification**.

   Project Input Table
   select **MM Tutorial-2.DELINQUENCYPROJECINPUT**

   Project Output Table
   select **MM Tutorial-2.DELINQUENCYPROJECTOUTPUT**



3. Examine the **Tutorial2** folder to verify that it contains the **Delinquency** project.



Define the Project Properties

To define the properties that SAS Model Manager uses to create reports and score models, follow these steps:

1. Select the **Delinquency** project in the **Tutorial2** folder and expand **Specific Properties** in the right pane.
2. Specify the default data tables and model variables for the project, then right-click **Delinquency**, and select **Refresh Item**.

   Default Test Table
   select **MM Tutorial-2.DELINQUENCYTEST**.

   Default Scoring Task Input Table
   select **MM Tutorial-2.DELINQUENCYSCORINGINPUT**.

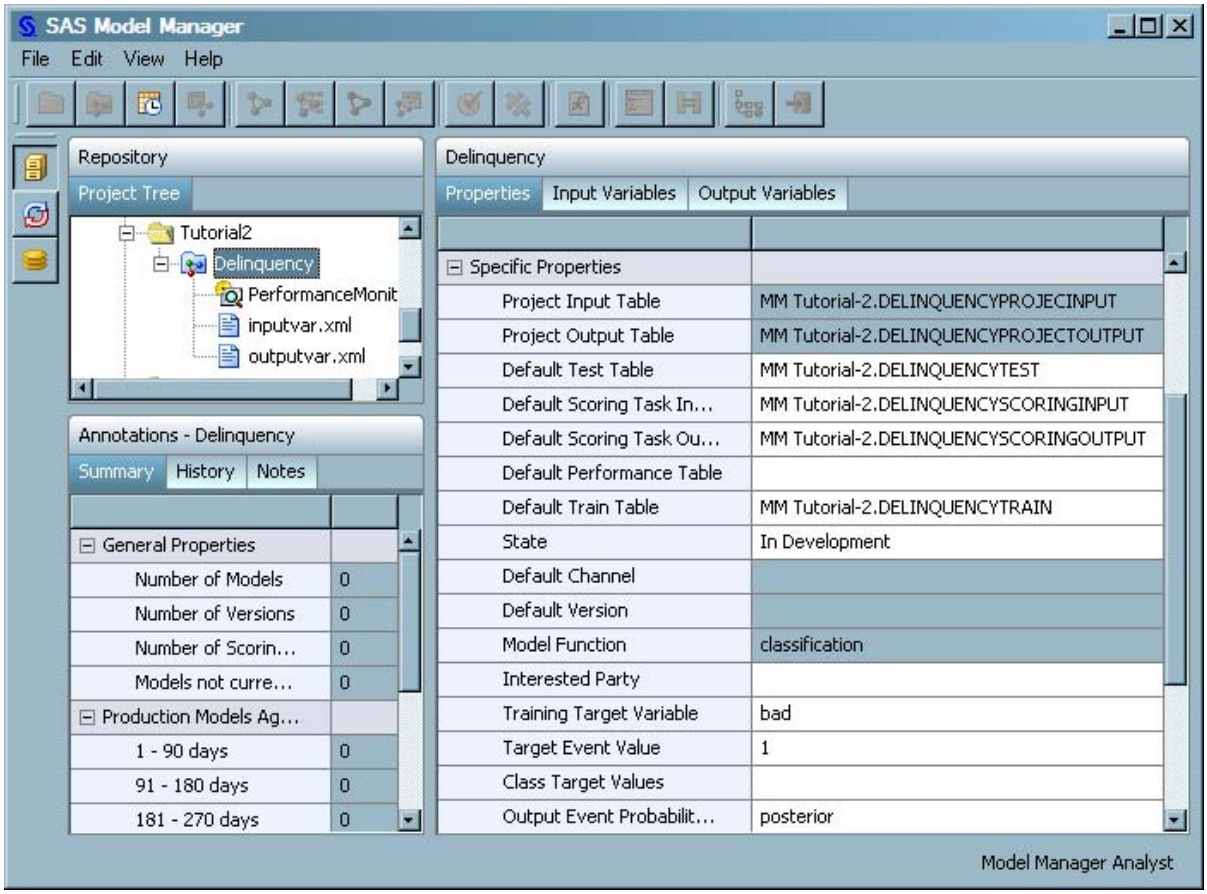
   Default Scoring Task Output Table
   select **MM Tutorial-2.DELINQUENCYSCORINGOUTPUT**.

   Default Train Table
   select **MM Tutorial-2.DELINQUENCYTRAIN**.

   Training Target Variable
   enter `bad`.

   Target Event Value
   enter `1`.

   Output Event Probability Variable
   select **posterior**.



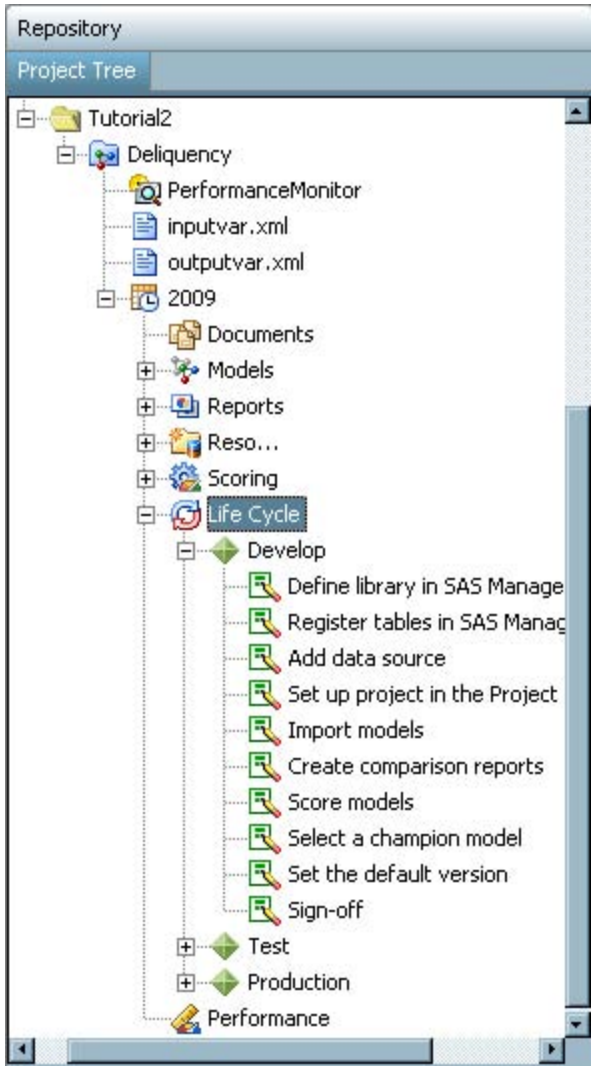
Create a Version

Create a version for the project. The version folder contains life cycle information, auxiliary version documents, candidate model files, model comparison reports, resource files, scoring tasks, and model performance reports. To create a new version, follow these steps:

1. Right-click the **Delinquency** project and select **New** ⇨ **New Version**. The New Version dialog box appears.
2. Specify the following version properties and click **OK**.

   Name
   enter `2009` for the version name.

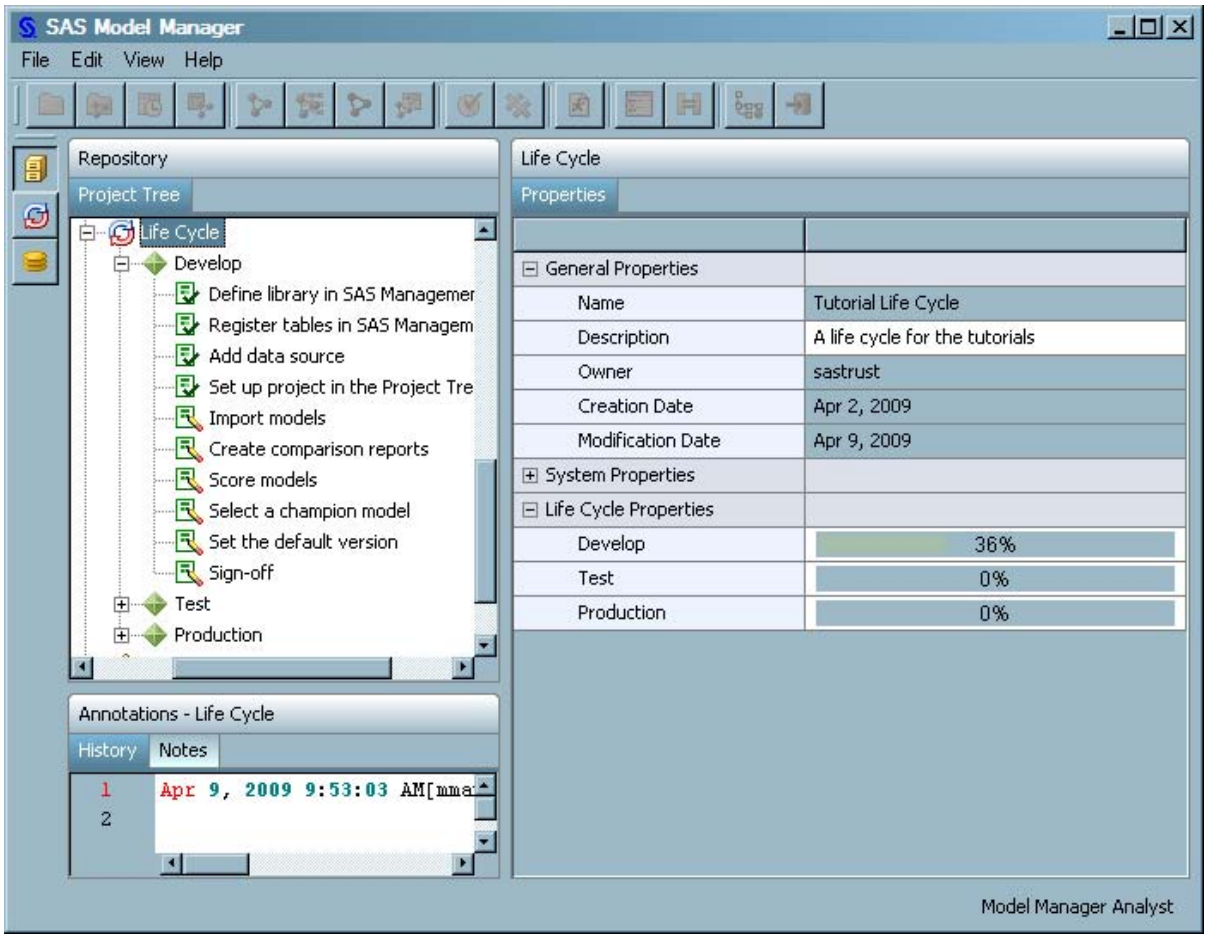
   Life Cycle Template
   select the user-defined template **Tutorial Life Cycle**.
3. Examine the **Delinquency** project to verify that it contains one version called **2009**. Select **Life Cycle**. Verify that the **Name** property is **Tutorial Life Cycle**.


Repository
Project Tree
Tutorial2
Deliquency
PerformanceMonitor
inputvar.xml
outputvar.xml
2009
Documents
Models
Reports
Reso...
Scoring
Life Cycle
Develop
Define library in SAS Manage
Register tables in SAS Manag
Add data source
Set up project in the Project
Import models
Create comparison reports
Score models
Select a champion model
Set the default version
Sign-off
Test
Production
Performance

Update the Life Cycle

To complete the milestone task of adding data sources and setting up the project in the Project Tree, follow these steps:

1. In the **Delinquency** project, expand **2009** ⇨ **Life Cycle** ⇨ **Development**.
2. Select the **Define library in SAS Management Console** task and examine the task properties. The **To Be Complete By** property, assigned in the life cycle template, determines which users or user groups from the Participants list are responsible for this milestone task. Because you are a member of the **MM Tutorial Assignees** group, you are authorized to update the task status.
3. Click the **Status** box and select **Completed**.
4. Select the **Register Models in SAS Management Console** task and examine the task properties. Click the **Status** box and select **Completed**.
5. Select the **Add data source** task. Click the **Status** box and select **Completed**.
6. Select the **Set up project in the Project Tree** task. Click the **Status** box and select **Completed**.
7. Select any object in the Project Tree to refresh the life cycle properties. Select all of the tasks whose status you updated and examine the properties. Verify that the value of the **Completed Date** property is today and that the value of the **Completed By** property is your user ID.
8. Select the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Develop** property displays a bar chart that shows the percentage of completed tasks for this milestone.



Import Models

In this exercise you import models into SAS Model Manager, set model properties, and map the model variables. The imported models are SAS code models. The SAS code for the first model is the LOGISTICS procedure, whereas the SAS code for the second and third models consists of DATA step fragments. To import SAS code file, at least three component files are required: the model score code, the model input file, and the model output file. For prediction or classification models, you also must prepare model target files.

Import SAS Code Models

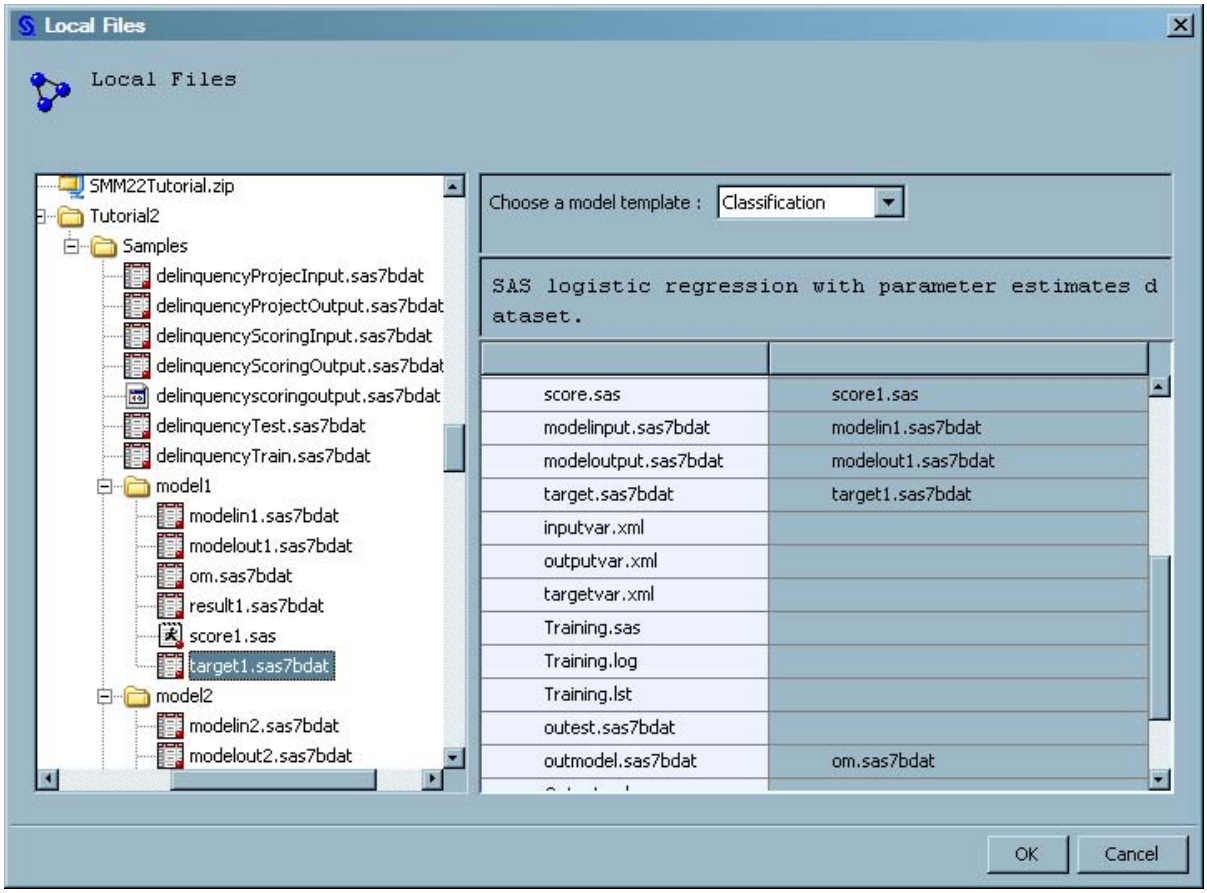
To import SAS code models, follow these steps:

1. Expand the **2009** version in the **Delinquency** project and right-click **Models** folder. Then select **Import From** ⇨ **Local Files**. The Local Files window opens.
2. Import Model 1.
   a. In the left pane, expand the **Desktop** folder and select `<drive>:\Tutorial2\Samples\model1`.
   b. Type `Model 1` in the **Name** box. For each filename in the Object column, click the filename and drag it to the corresponding option box. This action maps the

tutorial model component filenames to the SAS Model Manager model component filenames.

Object	Option
modelin1.sas7bdat	**modelinput.sas7bdat**
modelout1.sas7bdat	**modeloutput.sas7bdat**
om.sas7bdat	**outmodel.sas7bdat**
score1.sas	**score.sas**
target1.sas7bdat	**target.sas7bdat**

Here is the Local Files window after the files have been mapped.



c. Click **OK**.

3. Import Model 2.

a. Open the Local Files window. In the left pane, expand the **Desktop** folder and select **`<drive>:\Tutorial2\Samples\model2`**.

b. Type **`Model 2`** in the **Name** box. For each filename in the Object column, click the filename and drag it to the corresponding option box. This action maps the tutorial model component filenames to the SAS Model Manager model component filenames.

Object	Option
modelin2.sas7bdat	**modelinput.sas7bdat**
modelout2.sas7bdat	**modeloutput.sas7bdat**
ot.sas7bdat	**outmodel.sas7bdat**
score2.sas	**score.sas**
target2.sas7bdat	**target.sas7bdat**

c. Click **OK**.

4. Import Model 3.

   a. Open the Local Files window. In the left pane, expand the **Desktop** folder and select **`<drive>:\Tutorial2\Samples\model3`**.

   b. Type **`Model 3`** in the **Name** box. For each filename in the Object column, click the filename and drag it to the corresponding option box. This action maps the tutorial model component filenames to the SAS Model Manager model component filenames.

Object	Option
modelin3.sas7bdat	**modelinput.sas7bdat**
modelout3.sas7bdat	**modeloutput.sas7bdat**
score3.sas	**score.sas**
target3.sas7bdat	**target.sas7bdat**

   c. Click **OK**.

5. Examine the **Models** folder to verify that it contains the three models. Right-click the folder and select **Expand All Items** to examine the model files.

Set Model Properties

Set the properties for the model. SAS Model Manager requires that the **Score Code Type** be set to **Data Step** if the score code is a DATA step fragment, or be set to **SAS Program** if the score code is a SAS procedure. Follow these steps.

1. Select **Model 1**. Click the **Description** field and enter **`first model for tutorial 2`**.
2. Select **Model 2**. Click the **Score Code Type** box and select **DATA Step**.
3. Select **Model 3**. Click the **Score Code Type** box and select **DATA Step**.

Map Model Variables to Project Variables

When the names of the model output variable are not identical to the names of the project output variables, you must map the variables. To map model output variables to project output variables, follow these steps:

1. Map model variables for the first model. Select **Model 1** in the **Models** folder and then select the **Model Mapping** tab in the right pane. Ensure that the following model variables are mapped to their respective project variables. To map a model variable to a project variable, click in the box in the **Model Variables** column and select a variable.

Project Variables	Model Variables
prediction	**I_bad**
posterior	**P_1**

2. Map model variables for the second model. Select **Model 2** in the **Models** folder and then select the **Model Mapping** tab in the right pane. Ensure that the following model variables are mapped to their respective project variables. To map a model variable to a project variable, click the box in the **Model Variables** column and select a variable.

Project Variables	Model Variables
prediction	**prediction**
posterior	**prob2**

3. Map model variables for the third model. Select **Model 3** in the **Models** folder and then select the **Model Mapping** tab in the right pane. Ensure that the following model variables are mapped to their respective project variables. To map a model variable to a project variable, click the box in the **Model Variables** column and select a variable.

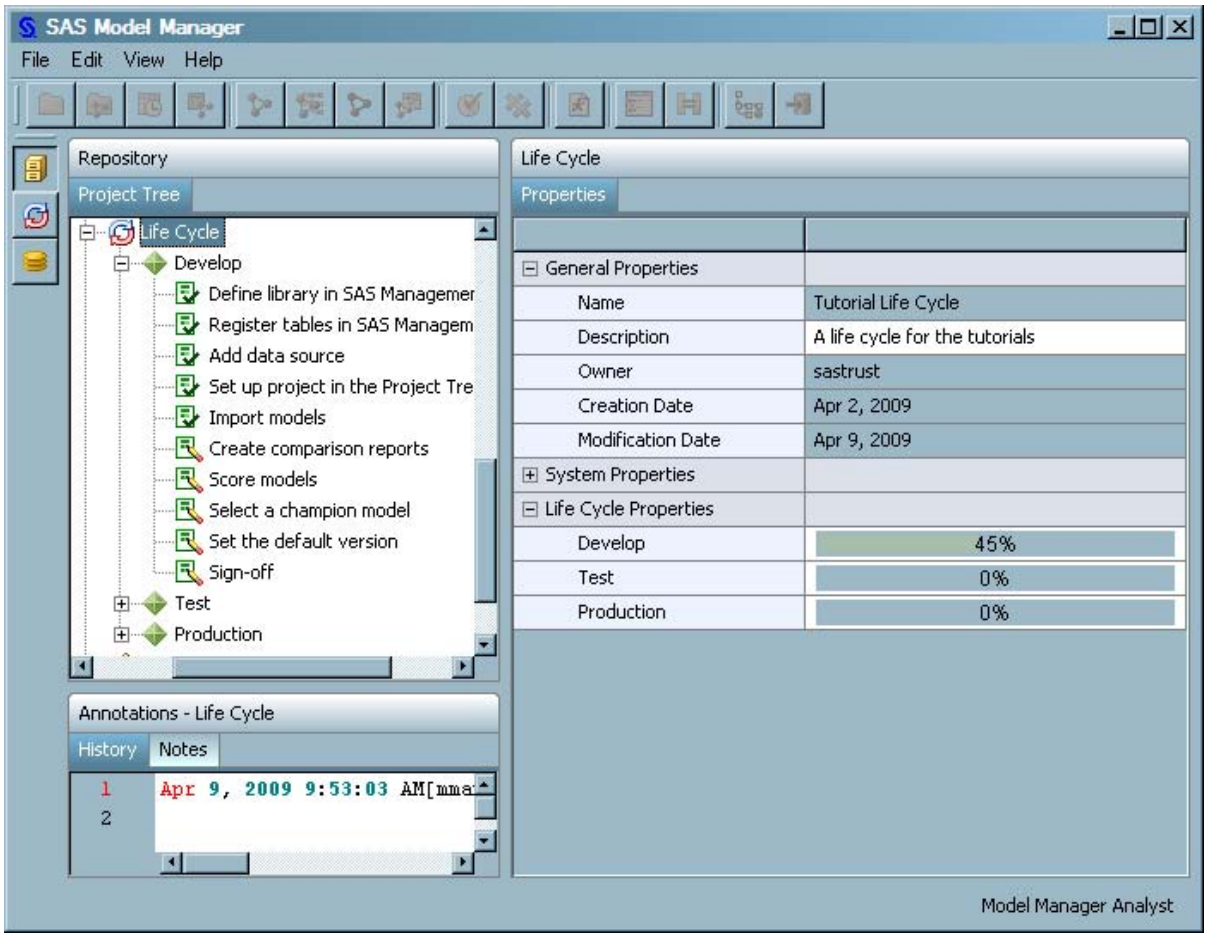
Project Variables	Model Variables
prediction	**prediction**
posterior	**p_bad1**

Update the Life Cycle

To update the Development milestone, follow these steps:

1. In the **Delinquency** project, expand **2009** ⇨ **Life Cycle** ⇨ **Develop**.
2. Select the **Import models** task. Click the **Status** box and select **Completed**.
3. Select the **Develop** milestone to refresh the property values. Select **Import models**. The **Completed Date** and **Completed By** fields have been updated with today's date and your user ID.

4. Click the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Develop** property displays a bar chart that shows the percentage of completed tasks for this milestone.



Create Model Comparison Reports

In this exercise you create several model comparison reports that are used in the selection and approval of a champion model. After you create the reports, you view the reports in the **Reports** folder. The reports enable you to evaluate candidate models in a version or across versions by assessing the structure, performance, and resilience of your models.

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 this report, follow these steps:

1. Expand the **2009** version in the **Delinquency** project and right-click the **Reports** folder. Then select **Reports** ⇨ **New Report Wizard**. The New Report Wizard opens.

2. In the New Report Wizard, use the specified values for these fields and click **OK**:

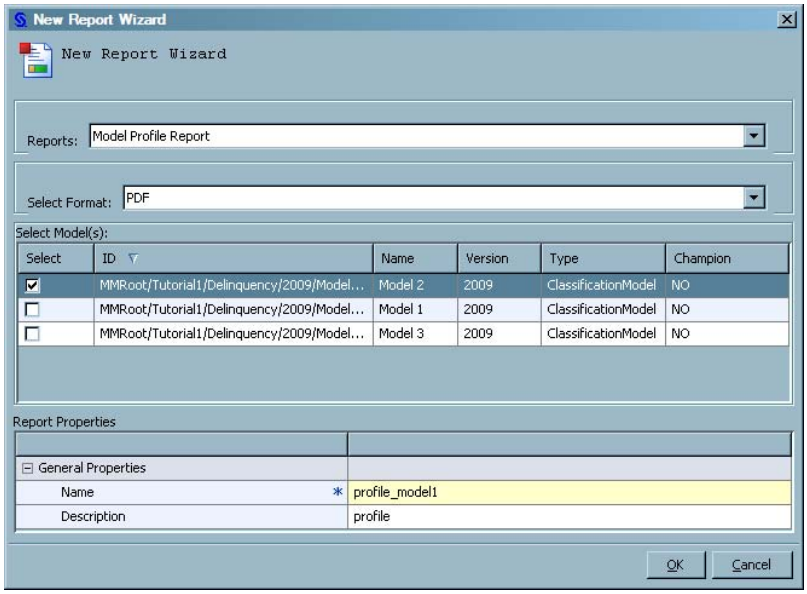
   Reports
   select **Model Profile Report**.

Select Format
select **PDF**. **PDF** is the default value. It might already be the value for **Select Format**.

Select Model(s)
select the box for **Model 1**.

Report Properties
replace the default report name with the report name **profile_model1** in the **Name** field .

Here is the New Report Wizard before you click **OK**.



3. When the **Information** dialog box confirms that the report was created successfully, click **OK**.

Create a Delta Report

The Delta report compares the profile data for two models and notes the differences. To create this report, follow these steps:

1. Expand the **2009** version in the **Delinquency** project and right-click the **Reports** folder. Then select **Reports** ⇨ **New Report Wizard**. The New Report Wizard opens.
2. In the New Report Wizard, use the specified values for these New Report Wizard fields and click **OK**:

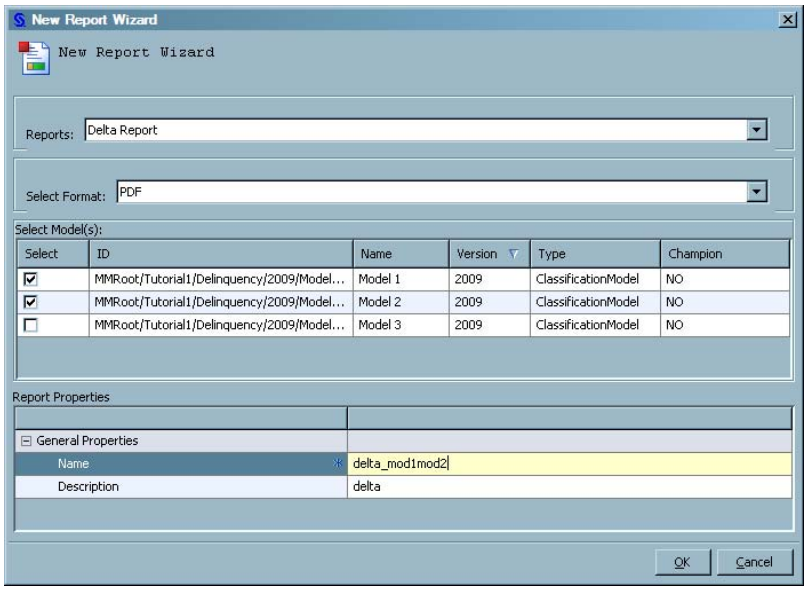
   Reports
   select **Delta Report**.

   Select Format
   select **HTML**.

   Select Model(s)
   select the boxes for **Model 1** and **Model 2**.

   Report Properties
   replace the default report name with the report name **delta_mod1mod2** in the **Name** field.

   Here is the New Report Wizard before you click **OK**.



3. When the **Information** dialog box confirms that the report was created successfully, click **OK**.

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 performance. To create this report, follow these steps:

1. Expand the **2009** version in the **Delinquency** project and right-click the **Reports** folder. Then select **Reports** ⇨ **New Report Wizard**. The New Report Wizard opens.

2. In the New Report Wizard, specify the following options and click **OK**:

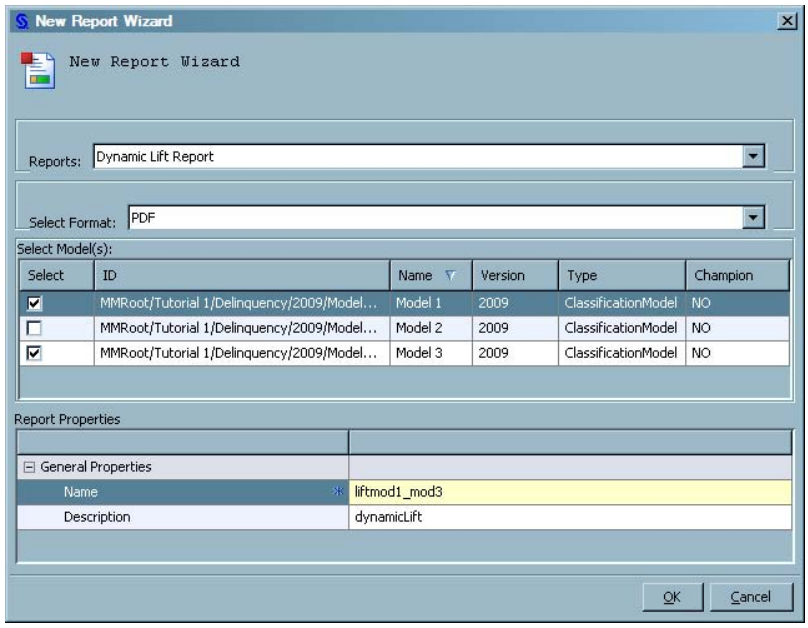
   Reports
   select **Dynamic Lift Report**.

   Select Format
   select **PDF**.

   Select Model(s)
   select the boxes for **Model 1** and **Model 3**.

   Report Properties
   replace the default report name with the report name **lift_mod1mod3** in the **Name** field.

   Here is the New Report Wizard before you click **OK**.



3. When the **Information** dialog box confirms that the report was created successfully, click **OK**.

View a Model Comparison Report

To view a model comparison report, follow these steps:

1. Expand the version folder **2009** and the **Reports** folder.
2. Right-click the report name and select **Reports** ⇨ **View Report**.

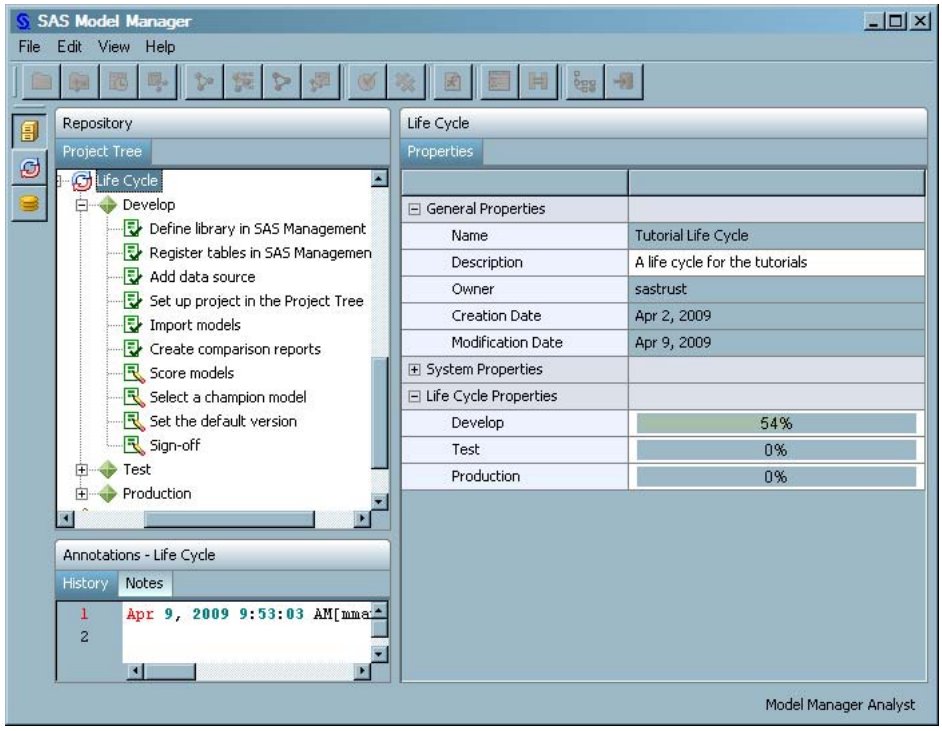
   Note: If user credentials are required, then specify a user ID and password that have permission to access the SAS Content Server.
3. Use the PDF viewer to distribute or print a copy of the report. In Adobe Reader, select **File** ⇨ **Attach to Email**, enter the e-mail address, and click **Send**.
4. Close the PDF Viewer.

For a detailed description of the model comparison reports, see the *SAS Model Manager: User's Guide*.

Update the Life Cycle

To update the Development milestone, follow these steps:

1. In the **Delinquency** project, expand **2009** ⇨ **Life Cycle** ⇨ **Develop**.
2. Select the **Create comparison reports** task. Select the **Status** box and select **Completed**.
3. Select the **Develop** milestone to refresh the property values. Select **Create comparison reports**. The **Completed Date** and **Completed By** fields have been updated with today's date and your user ID.
4. Click the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Develop** property displays a bar chart that shows the percentage of completed tasks for this milestone.



Using the Annotation View

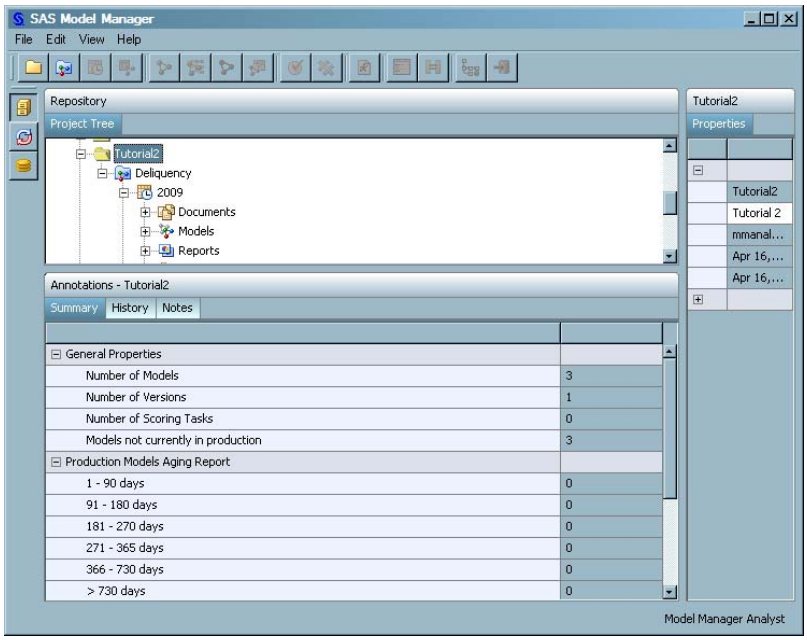
In this exercise, you use the **Annotations** view to examine the time-stamped event log, add text information to a model component, and examine the synopsis of the project and model reports. The Annotations view is the lower left pane of the SAS Model Manager window.

View History

To view the event log for the different components of a project, follow these steps:

1. Select the **Delinquency** project.
2. In the **Annotations** view, select the **History** tab. This tab displays a time-stamped entry each time you create, modify, import, publish, export, or delete a component. SAS Model Manager records the following information:
   - The date and time that the action occurred
   - The user ID that performed the action
   - The action that was performed

Here is an example of the **History** in the **Annotations** view:

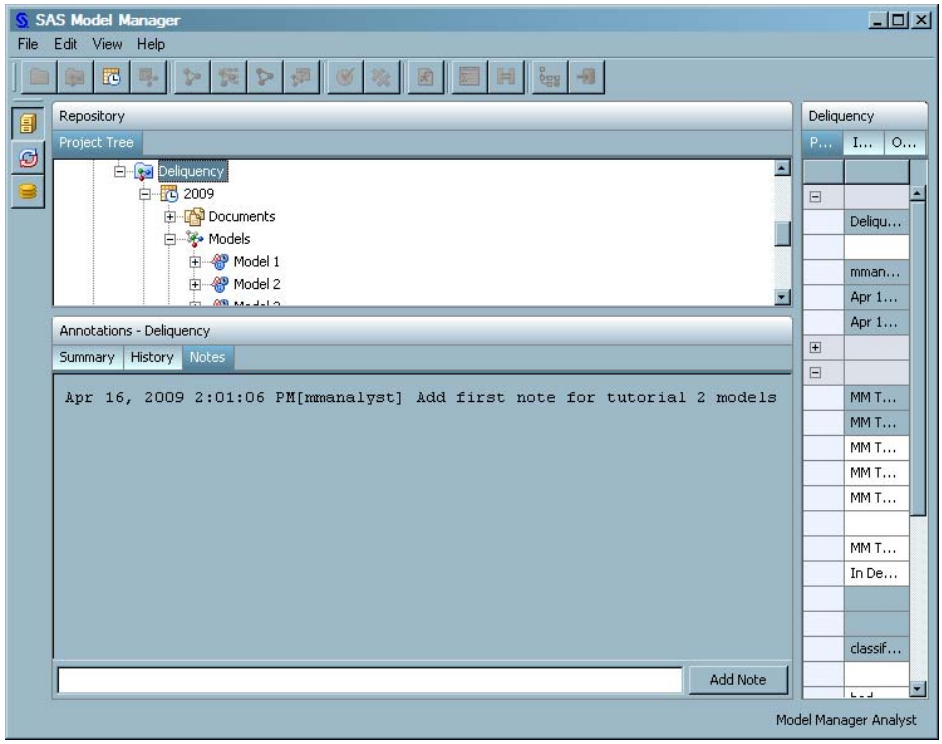


View Notes

To create persistent annotations that are associated with the different components of a project, follow these steps:

1. Expand the **2009** version in the **Delinquency** project and select the **Models** folder.
2. In the **Annotations** view, select the **Notes** tab.
3. In the **Add Notes** field, enter `Add first note for tutorial 2 models` and click **Add Notes**.

   Here is the Delinquency project note in the **Annotations** view:



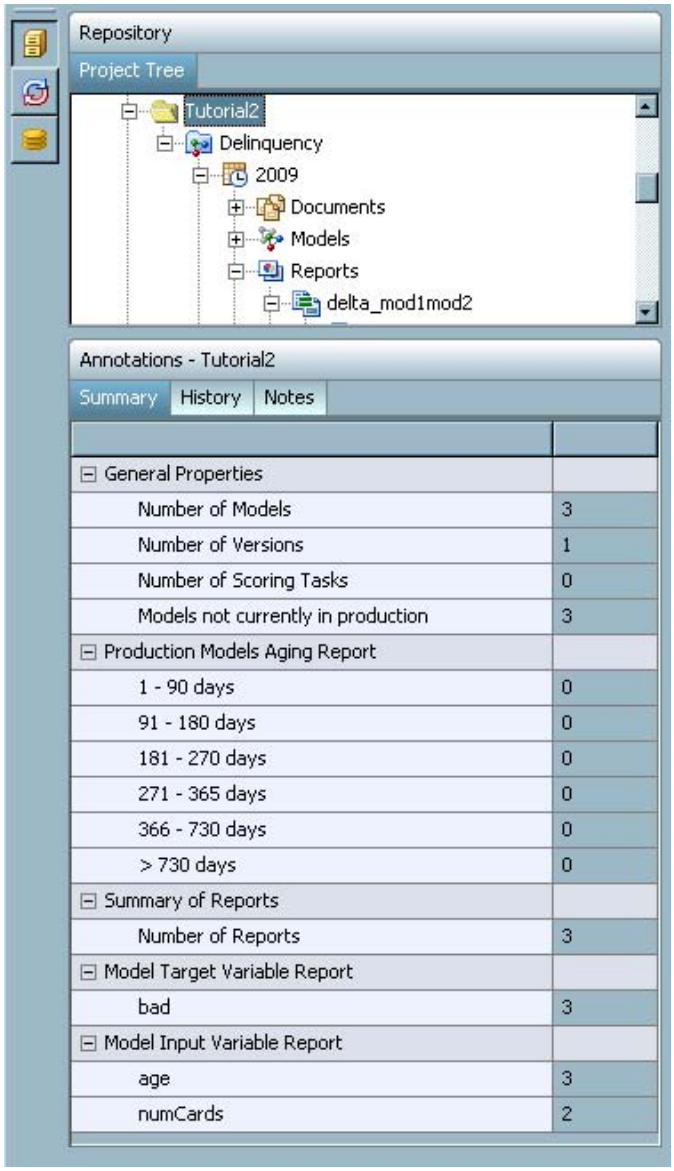
View the Summary Information

The Summary contains information about the components that are contained in the selected folder.

To view summary information, follow these steps:

1. In the Project Tree, click the **Tutorial2** folder.
2. In the **Annotations** view, click the **Summary** tab.
3. Examine the information on this tab. SAS Model Manager includes general property information about the components that are part of this folder, aging information, number of reports, target information, and input information about the project.

Here is the **Summary** information for **Tutorial2**:



General Properties	
Number of Models	3
Number of Versions	1
Number of Scoring Tasks	0
Models not currently in production	3
Production Models Aging Report	
1 - 90 days	0
91 - 180 days	0
181 - 270 days	0
271 - 365 days	0
366 - 730 days	0
> 730 days	0
Summary of Reports	
Number of Reports	3
Model Target Variable Report	
bad	3
Model Input Variable Report	
age	3
numCards	2

The information on the **Summary** tab dynamically reflects the contents of the selected node and its subnodes in the Project Tree. If you select **MMRoot**, you see summary information for all nodes in the Project Tree. If you select a project, the summary information reflects the project and all nodes within that project.

Scoring Models

In this exercise you create a scoring task that is used to run the score code of a model and produce scoring results. You use the results to determine the scoring accuracy and to analyze the model performance. The scoring task uses data from a scoring task input table, and then generates the results in a scoring task output table.

Create a Scoring Task

To create a new scoring task, follow these steps:

1. Expand the **2009** version, right-click the **Scoring** folder and select **New** ⇨ **New Scoring Task**. The **New Scoring Task** window opens.
2. Specify the following options and click **OK**:

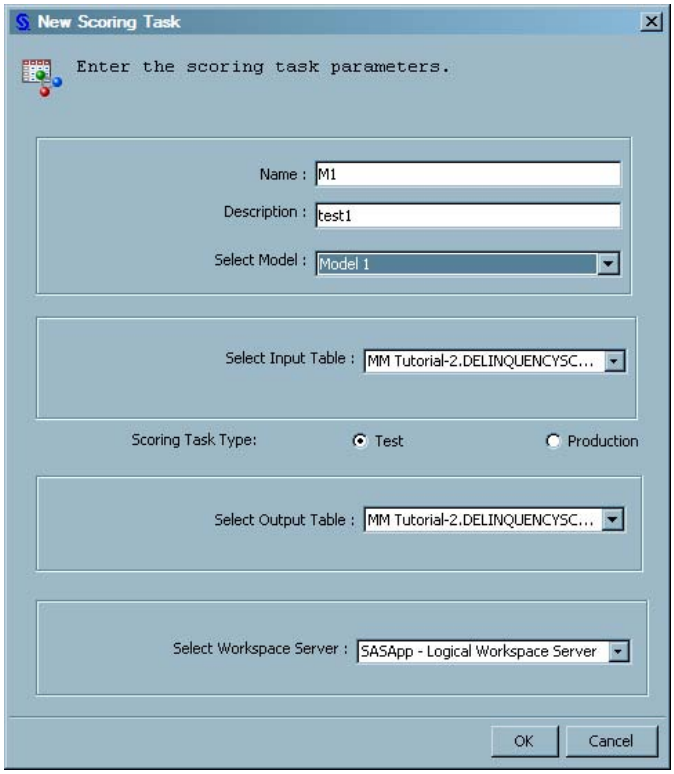
   Name
   enter `M1` for the project name.

   Description
   enter `test1`.

   Select Model
   select **Model 1**. This model controls the available values for the input and output tables.

   Scoring Task Type
   select **Test**.

   TIP A best practice is to start all scoring tasks with **Test** selected. The results of a scoring task when it is run as type **Test** are not overwritten. You can change the type to **Production** after you are satisfied with the scoring task results and when the model is ready for production.

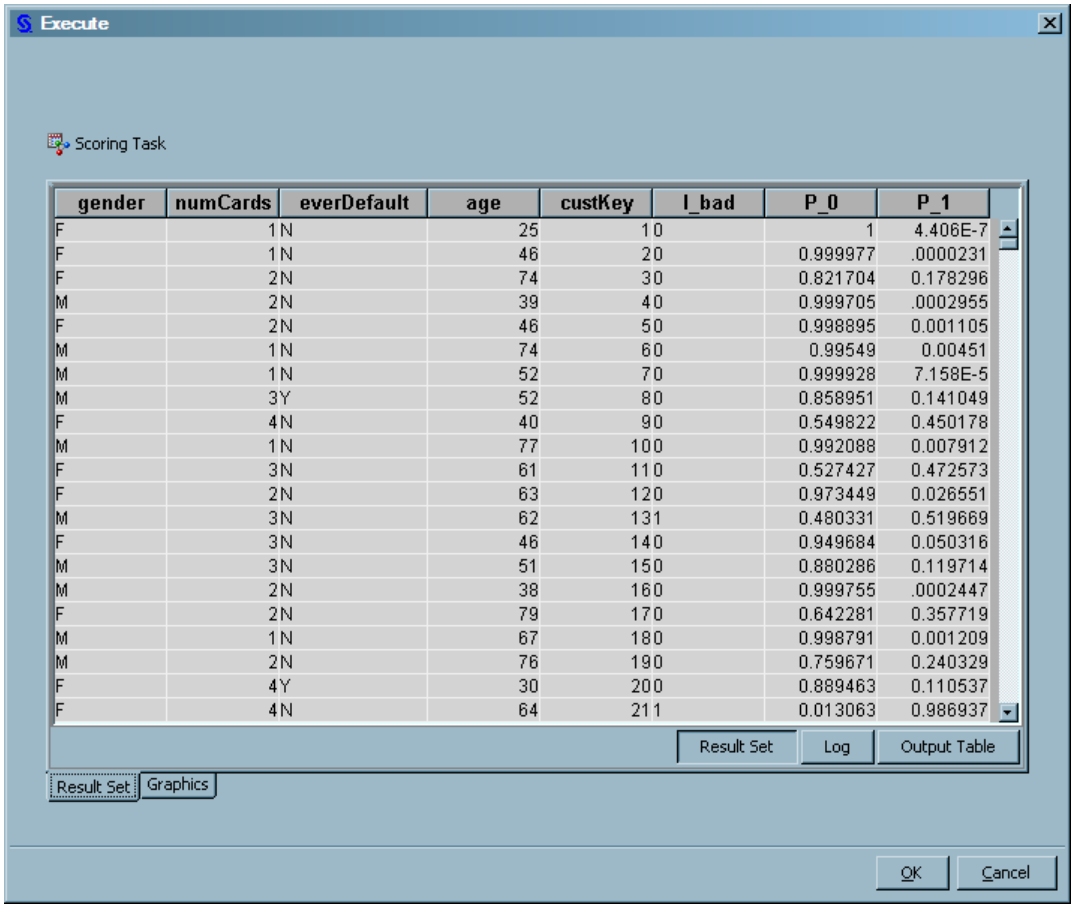


3. Select the **M1** scoring task to examine its properties. The value for **Modification Date** is today's date. To change the scoring task name or model input and output tables, you must create a new scoring task.

Execute a Scoring Task

To execute a scoring task, follow these steps:

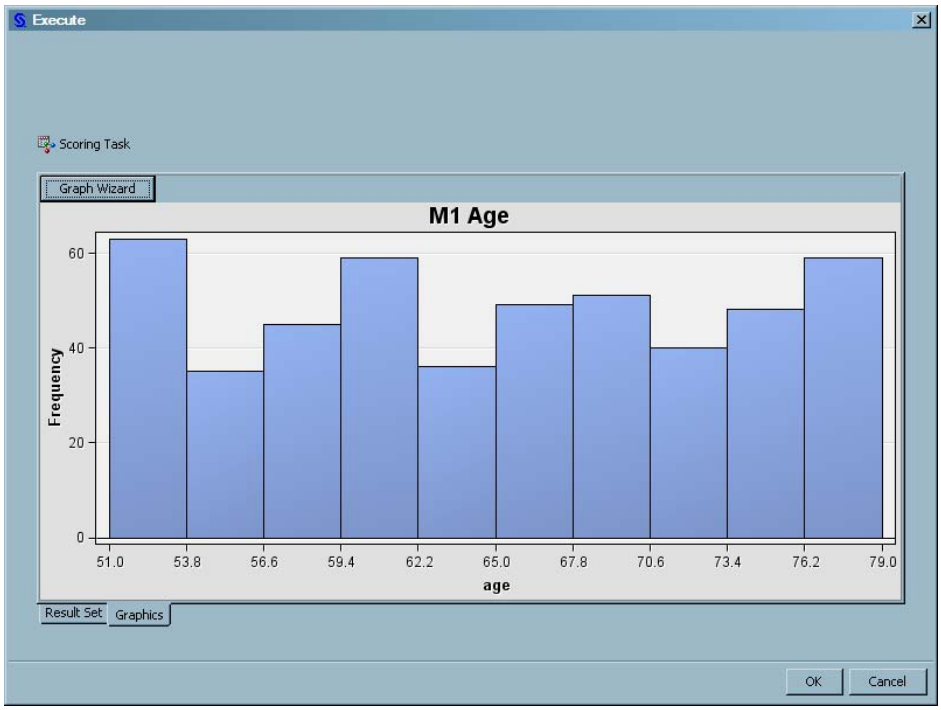
1. Verify that you have mapped the model output variables to the scoring task output variables. For more information, see "Map Model Variables to Project Variables" on page 39.
2. Validate the input variables. Expand the **Scoring** folder, select the **M1** scoring task and click the  toolbar button. Examine the results of **Quick Check** and then click **OK**.
3. Right-click the **M1** scoring task and select **Execute**.
4. When the **Information** dialog box confirms that the report was created successfully, click **OK**. The scoring task results appear in the Execute window:



   If the scoring task was not successful, then review the **Log** tab for error messages.
5. Select the **Result Set** tab to view the scoring task results in a tabular form.
6. Click the **Graph** tab to graph the results.
   a. Select **Histogram** and click **Next**.
   b. In the upper-right corner, click **Use default assignments** and then click **Next**.
   c. Click the **Column name box** and select **age**.
   d. Click the **Operator** box and select **Greater than**.
   e. In the **Value** field, enter `50`. Click **Next**.

f. In the **Title** field of the Chart Titles page, type `M1 Age`. Click **Next** and then click **Finish**.

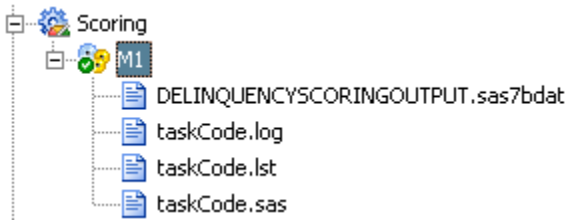
Here is the histogram in the Execute window:



Then click **OK** in the Execute window to save the results. For more information, see the *SAS Model Manager: User's Guide*.

7. Expand the **M1** scoring task to verify that four content files were saved and that the value for **Modification Date** is today's date.

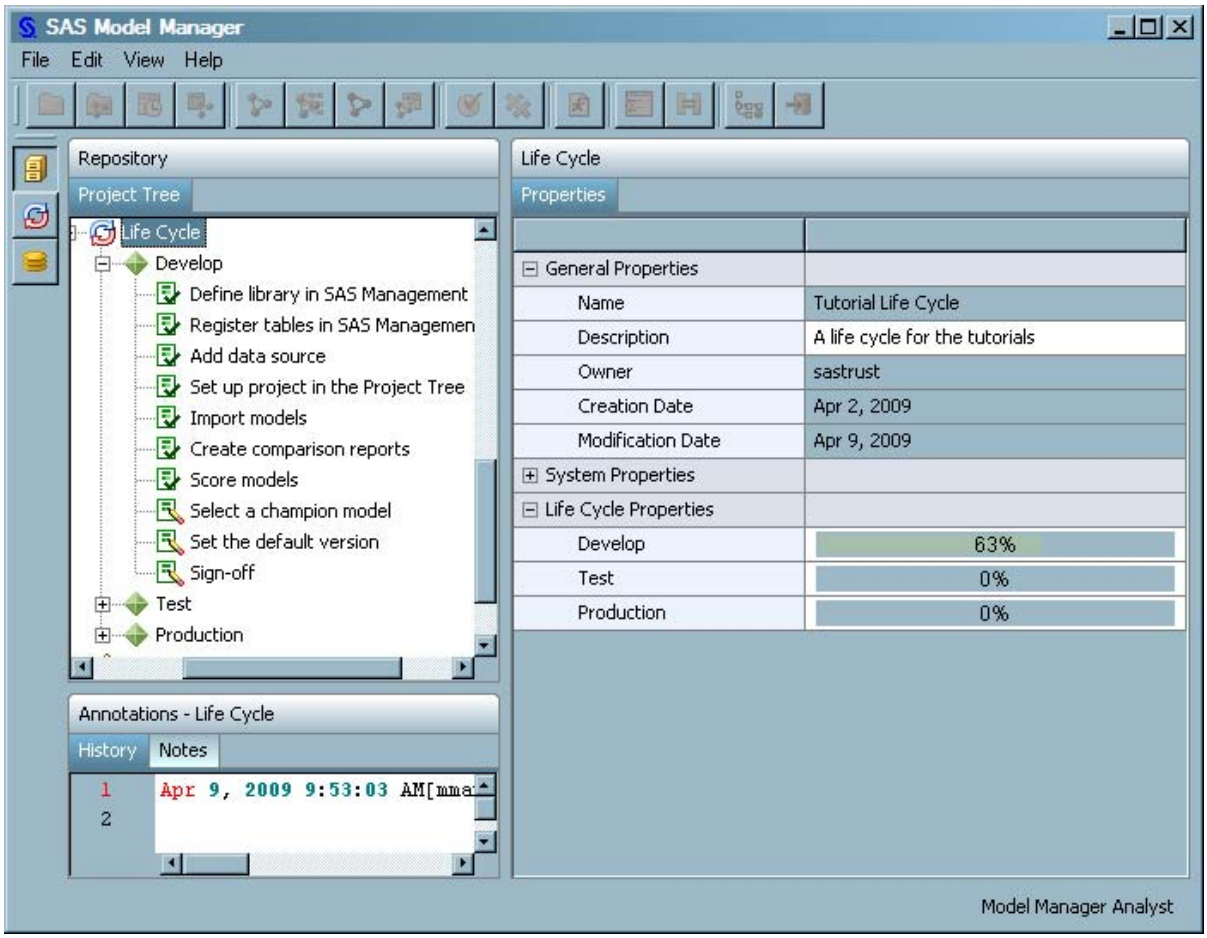
Here is the **Scoring** folder and the files for the **M1** scoring task:



Update the Life Cycle

To update the Development milestone, follow these steps:

1. In the **Delinquency** project, expand **2009** ⇨ **Life Cycle** ⇨ **Development**.
2. Select the **Score models** task. Click the **Status** box and select **Completed**.
3. Select the **Develop** milestone to refresh the property values. Select **Score models**. The **Completed Date** and **Completed By** fields have been updated with today's date and your user ID.
4. Click the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Development** property displays a bar chart that shows the percentage of completed tasks for this milestone.



Declare a Champion Model

In this exercise you declare a champion model.

Set the Champion Model

To set a champion model, follow these steps:

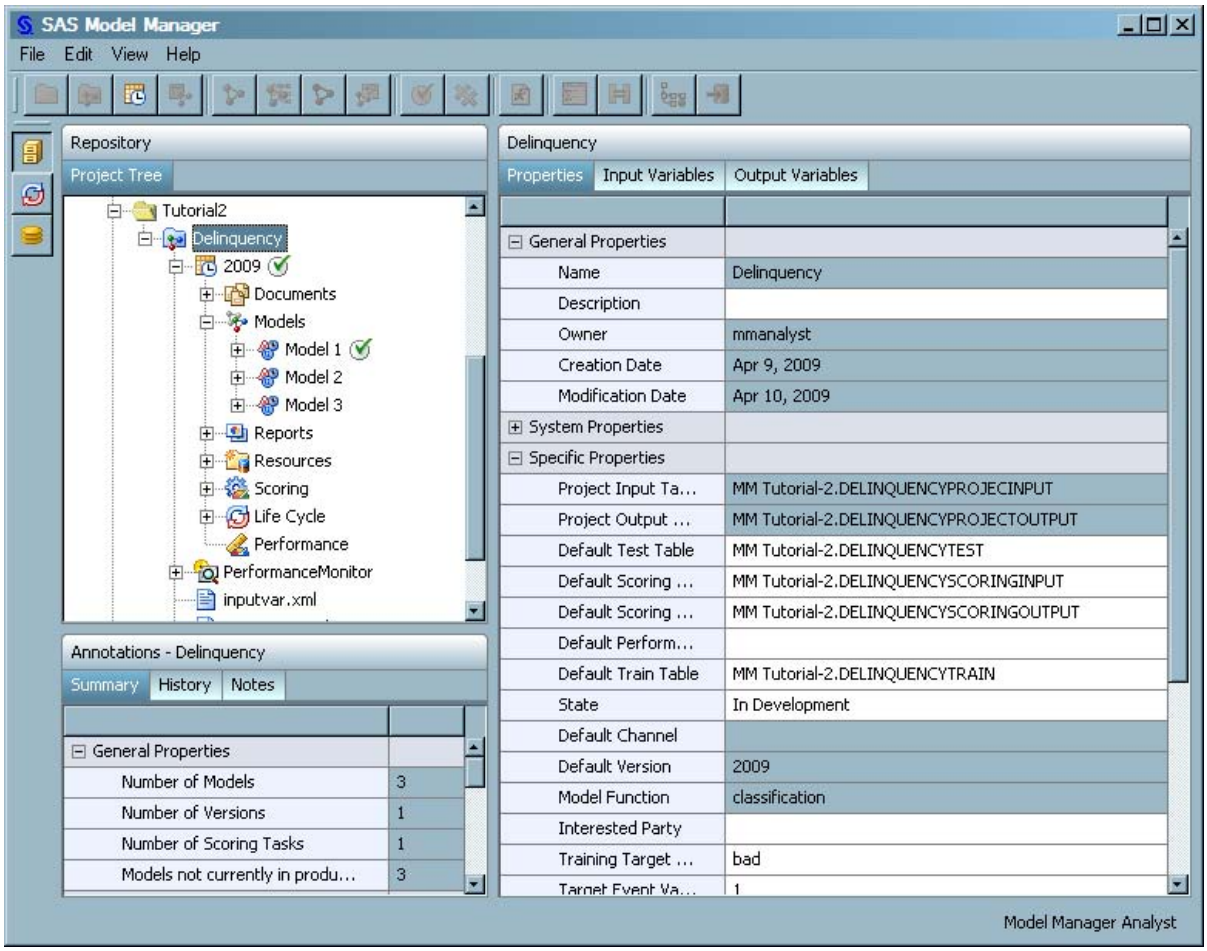
1. Expand the **Models** folder in the **2009** version. Right-click **Model 1**, select **Set Champion Model**, and click **Yes** to confirm.
2. Verify that the ✓ icon appears next to the champion model.
3. Select the version folder to examine its properties. The value for **Modification Date** is today's date. The value for the **Champion Model Id** is the champion model's UUID.

   TIP To document the reasons or assumptions for your selection of the champion model, use the version **Notes** tab. SAS Model Manager automatically annotates the History tab. For more information, see "Using the Annotation View" on page 44.

Set Default Version

You assign a default version after the default champion model for the project is identified. To set the default version, follow these steps:

1. Right-click the **2009** version and select **Set Default Version**. Click **Yes** to confirm.
2. Verify that the ✓ icon appears next to the version folder.
3. Select the project folder to examine its properties. The value for **Modification Date** is today's date. The value for the **Default Version** is the name of the version folder.



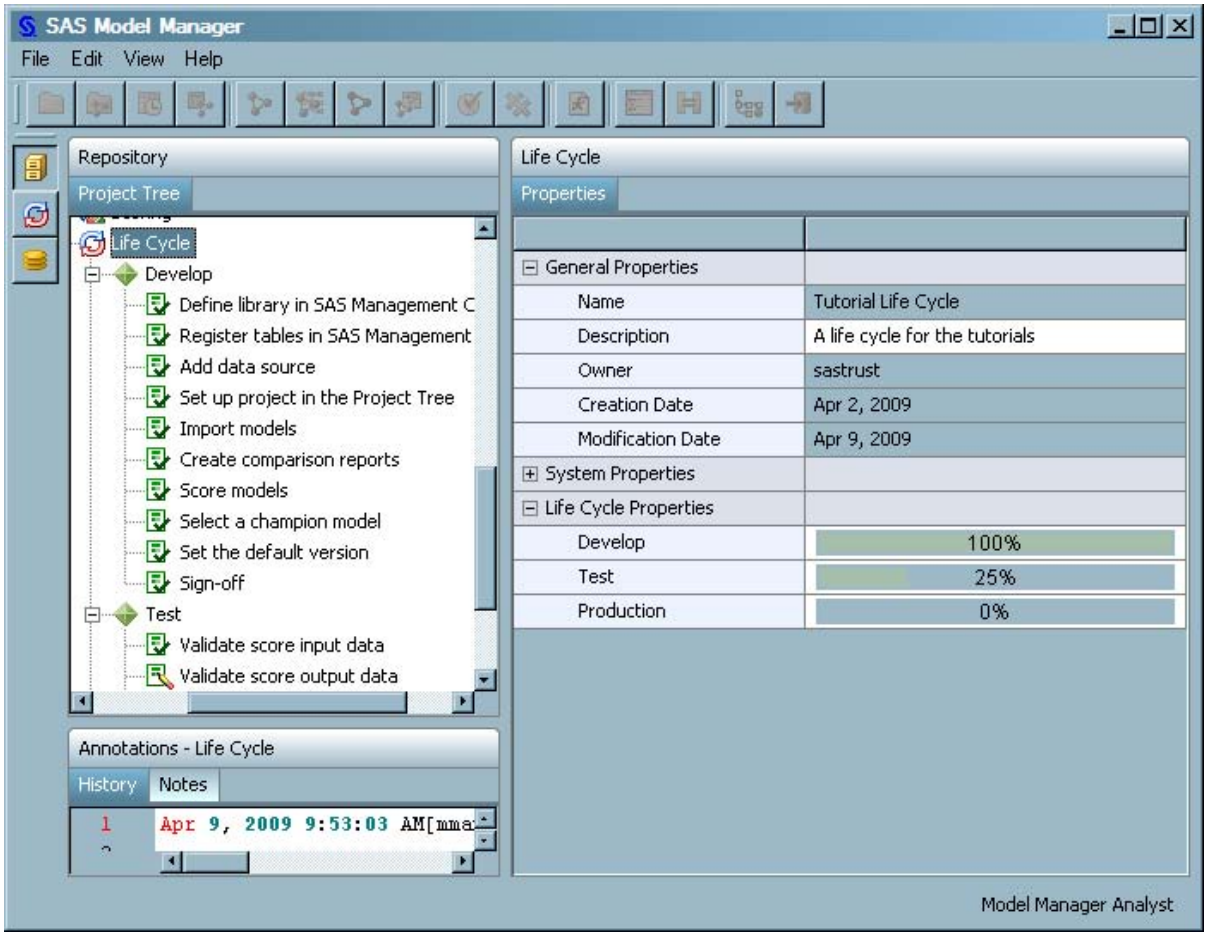
Update the Life Cycle

To update the Development milestone, follow these steps:

1. In the **Delinquency** project, expand **2009** ⇨ **Life Cycle** ⇨ **Development**.
2. Select the **Select a champion model** task. Click the **Status** box and select **Completed**.
3. Select the **Set the default version** task. Click the **Status** box and select **Completed**.
4. Select the **Sign-off** task to indicate that all of the **Develop** milestone tasks are complete. Click the **Status** box and select **Completed**.

5. Select the **Develop** milestone to refresh the property values. Select the **Select a champion model**, **Set the default version**, and **Sign-off** tasks. The **Completed Date** and **Completed By** fields have been updated with today's date and your user ID.
6. Expand the **Test** milestone. Select the **Validate Score Input Data** task. Click the **Status** box and select **Completed**.

   Note: The **Select Champion** task must have been completed before you can complete this task.
7. Click the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Development** and **Test** properties display a bar chart that shows the percentage of completed tasks for this milestone.



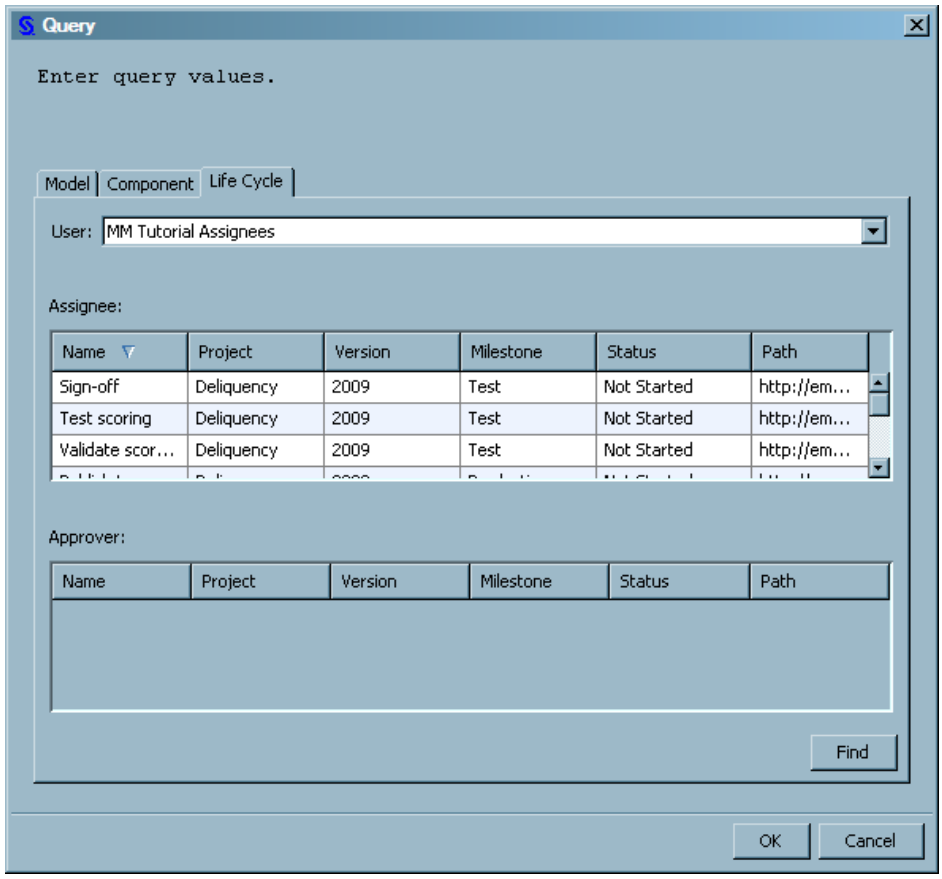
Query for the Remaining Project Tasks to Complete

In this example you search for the status of life cycle tasks by using the Query utility.

To search for the status of life cycle tasks, follow these steps:

1. Right-click the **Tutorial2** folder and select **Query**. The Query window opens.
2. Click the **Life Cycle** tab. Select the **User** box and select **MM Tutorial Assignees** and then click **Find**.

3. Examine the status of the associated milestones and click **OK**. The search results display tasks in the **Assignee** list that are assigned to the user and tasks in the **Approver** list that the user is assigned to approve. The **Assignee** query results return only the tasks that have a status of **Started** or **Not Started**. Results that have a status of **Complete** or **Approved** are omitted.



Chapter 4

Tutorial 3: Importing and Exporting Models

Overview of Importing and Exporting Models

SAS Model Manager provides several methods to import SAS models into a project version. You can import your SAS models into a project version from the SAS Metadata Repository, SAS Enterprise Miner package files, SAS code, and PMML models. SAS

macros are also available so that you can use SAS code to import or register SAS models into your project version.

After the champion model is validated, you can publish models or publish a scoring function for a model to a database. SAS Model Manager publishes models to channels and exports champion models to the SAS Metadata Repository.

The tutorial provides examples and step-by-step directions for performing these tasks.

Prepare Tutorial Data Sets and Models

In this exercise you create a library in SAS Management Console for the data tables that are used in this tutorial. You verify that your user ID is a member of the MM Tutorial Assignees group.

The Required Tutorial Files

The SAS data sets and models that are required for this tutorial are on your local computer after you extract them from the ZIP file SMM22Tutorial.zip. If you have not extracted the tutorial files, see "The Tutorial Files" on page 3.

This tutorial requires the following files and folders in the **<*drive*>\Tutorial3\Samples** folder:

- hmeq_project_input.sas7bdat
- hmeq_project_output.sas7bdat
- hmeq_score_input.sas7bdat
- hmeq_test.sas7bdat
- hmeq_train.sas7bdat
- The **Reg1** folder contains the file miningResult.spk
- The **Tree1** folder contains the file miningResult.spk
- The **Neural** folder contains the file Neutral.xml

Define a Data Library in SAS Management Console

Use SAS Management Console to define a data library and register the tables. Follow these steps:

1. Open SAS Management Console and log on to the SAS Metadata Server. Your user ID must be authorized to modify libraries in the metadata server.
2. Open the New Library Wizard to define the data library. Click **Environment Management** ⇨ **Data Library Manager** on the **Plug-ins** tab. Right-click **Libraries** and select **New Library** from the pop-up menu.
3. In the New Library Wizard, create a SAS library.
   a. Select **SAS BASE Library** and click **Next**.
   b. Specify **MM Tutorial-3** in the **Name** field and click **Next**.
   c. (Optional) Depending on your configuration, if more than one server exists, then select a server.
   d. Specify **smm2tor3** for the libref and click **New**.

e. Specify the server folder that you previously created, `<drive>\Tutorial3\Samples`, for the path specification and click **OK** twice.

f. Click **Next** and **Finish**.

4. Register the data tables in the metadata repository.

a. Right-click **MM Tutorial-3** under the **Libraries** node, select **Register Tables** from the pop-up menu, and click **Next**.

b. If prompted, specify the metadata server, test the server connection, and click **OK**. Then click **Next**.

c. Click **Select All Tables**, click **Next**, and click **Finish**.

5. Verify that table metadata was created and close SAS Management Console. Select **MM Tutorial-3** under the **Libraries** node and examine the right pane.

6. Authorize the **MM Tutorial Assignees** group and the **MM Tutorial Approvers** group to read and write to the library:

a. Right-click the **MM Tutorial-3** library and select **Properties**.

b. Click the **Authorization** tab and then click the **Add** button.

c. Select **MM Tutorial Assignees** from the **Available Identities** list and click  to move the group to the **Selected Identities** list.

d. Select **MM Tutorial Approvers** from the **Available Identities** list and click  to move the group to the **Selected Identities** list.

e. Click **OK** twice.

Verify Your User ID as a Member of the MM Tutorial Assignees Group

This exercise ensures that your user ID is a member of the **MM Tutorial Assignees** group and the **Model Manager Advanced Users** group.

1. Open SAS Management Console and log on to the SAS Metadata Server.

2. On the **Plug-ins** tab, select **User Manager**.

3. In the right pane, double-click the **MM Tutorial Assignees** group and click the **Members** tab.

4. Review the **Current Members** list, and locate your user ID or a group that your user ID is a member of. If your user ID or group is not a member of the **MM Tutorial Assignees** group, ask your administrator to add you to this group. Close the properties window.

5. Find and double-click your user ID in the right pane of SAS Management Console.

6. Click the **Groups and Roles** tab. Review the **Member of** pane and locate the group **Model Manager Advanced Users**. If your user ID is not a member of this group, ask your administrator to add you to this group. Close the properties window.

7. Close SAS Management Console.

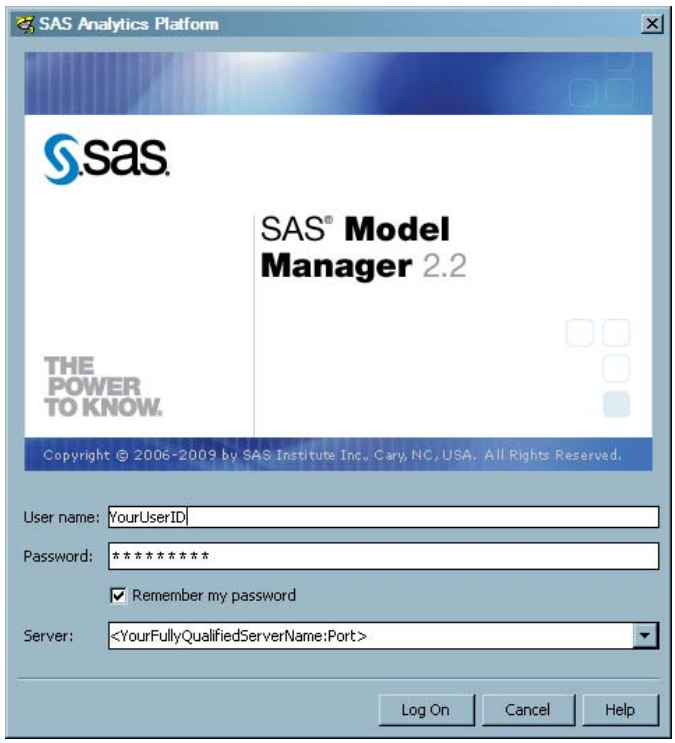
Add Input Data Sources

In this exercise you use SAS Model Manager to access the data tables that the models use.

Start SAS Model Manager

To run SAS Model Manager, follow these steps:

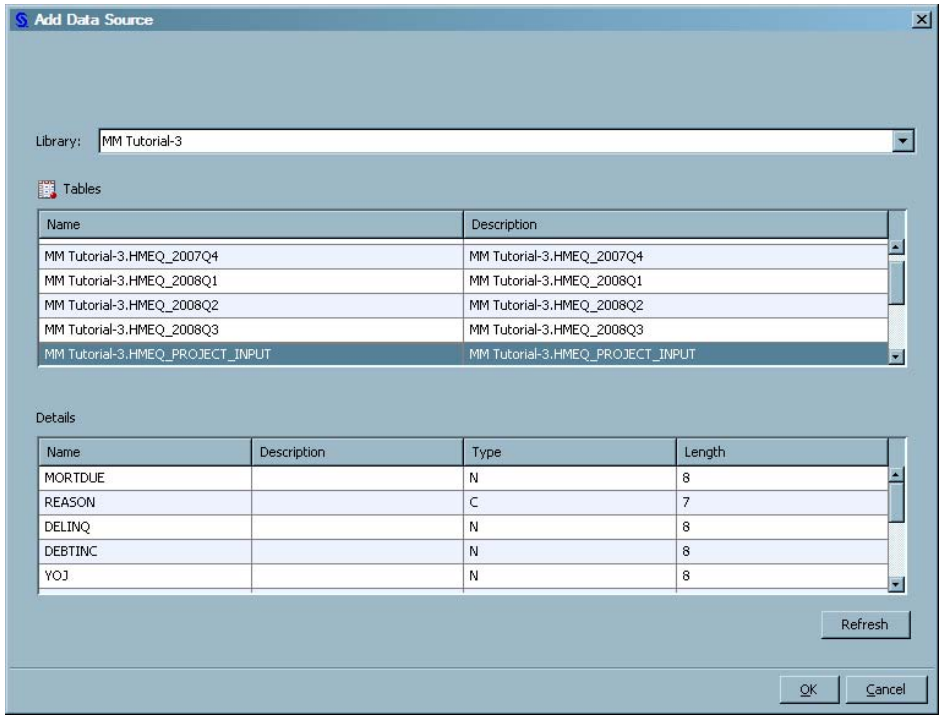
1. On your client machine, start the SAS Model Manager client.
2. Log on to SAS Model Manager. Ensure that your user ID is set up to use the tutorials.



Select Data Sources

To add data tables in the Data Source perspective that you use to define projects, create scoring tasks, and create reports, follow these steps:

1. Log on to SAS Model Manager. Ensure that your user ID is a member of the **Model Manager Advanced Users** group.
2. Click the **Data Sources** perspective button  .
3. Right-click **Project Input Tables** and select **Add Data Source**. The Add Data Source window opens.
4. Click the **Library** box and select **MM Tutorial-3**.
5. In the **Tables** box, select **MM Tutorial-3.HMEQ_PROJECT_INPUT** and click **OK**.



6. Repeat the above steps to add the remaining tables as SAS Model Manager data sources:

Folder Name	Data Table Name
Project Output Tables	**MM Tutorial-3.HMEQ_PROJECT_OUTPUT**
Test Tables	**MM Tutorial-3.HMEQ_TEST**
Scoring Task Input Tables	**MM Tutorial-3.HMEQ_SCORE_INPUT**
Train Tables	**MM Tutorial-3.HMEQ_TRAIN**

Organize the Model Hierarchy

In this exercise you create an organizational folder, a project, and a version for the modeling project.

Create a Folder

To provide an organizational folder to manage your modeling projects, follow these steps:

1. Right-click the **MMRoot** node in the Project Tree and select **New** ⇨ **New Folder**. The New Folder dialog box appears.
2. Specify the following folder properties and click **OK**.

   Name
   enter `Tutorial3`.

Description
enter an optional folder description.

The new folder appears in the Project Tree.

Create a New Project

To create a project, follow these steps:

1. Right-click the **Tutorial3** folder and select ⇨ **New** ⇨ **New Project**. The New Project dialog box appears.
2. Specify the following project properties and click **OK**:

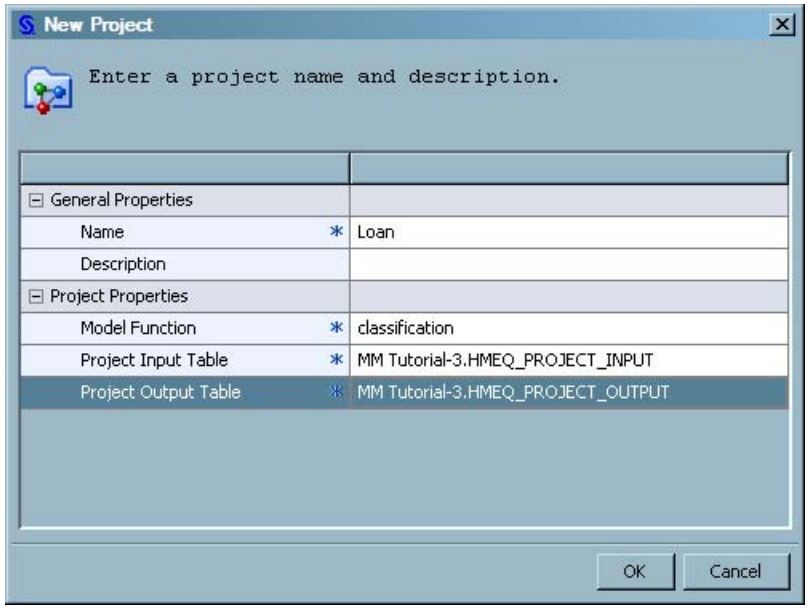
   Name
   enter `Loan`.

   Description
   enter an optional description.

   Model Function
   select **classification**.

   Project Input Table
   select **MM Tutorial-3.HMEQ_PROJECT_INPUT**.

   Project Output Table
   select **MM Tutorial-3.HMEQ_PROJECT_OUTPUT**.



3. Examine the **Tutorial3** folder to verify that it contains the **Loan** project.

Define the Project Properties

To define the properties that SAS Model Manager uses to create reports and score models, follow these steps:

1. Select the **Loan** project in the **Tutorial3** folder and expand **Specific Properties** in the right pane.

2. Specify the default data tables and model variables for the project:

   Default Test Table
   select **MM Tutorial-3.HMEQ_TEST**.

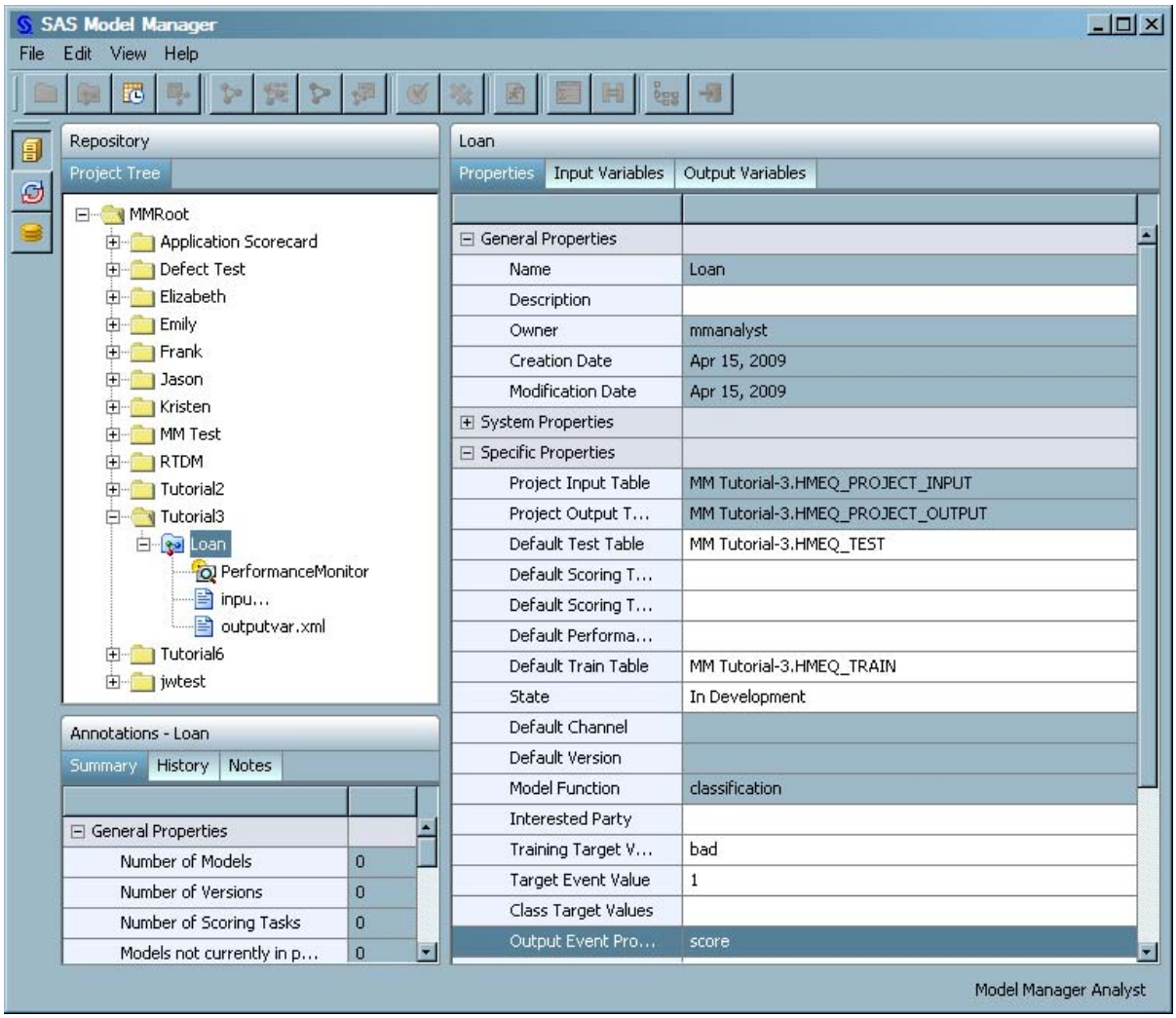
   Default Train Table
   select **MM Tutorial-3.HMEQ_TRAIN**.

   Training Target Variable
   enter **`bad`**.

   Target Event Value
   enter 1.

   Output Event Probability Variable
   select **score**.

   Here are the project properties:



Create a Version

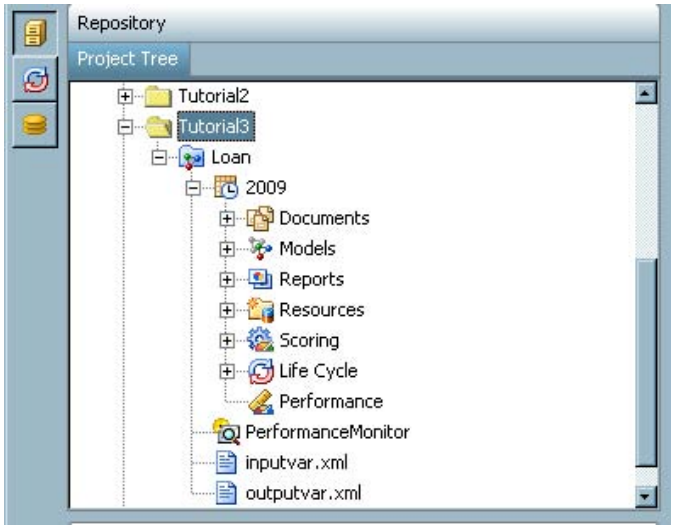
Create a version for the project. The version folder contains life cycle information, auxiliary version documents, candidate model files, model comparison reports, resource files, scoring tasks, and model performance reports.

To create a new version, follow these steps:

1. Right-click the **Loan** project and select **New** ⇨ **New Version**. The New Version dialog box appears.
2. Specify the following version properties and click **OK**.

   Name
   enter `2009`.

   Life Cycle Template
   select the user-defined template **Tutorial Life Cycle** that you created in the first tutorial. For more information, see "Create a Life Cycle Template" on page 9.
3. Examine the **Loan** project to verify that it contains one version called **2009**. Select **Life Cycle**. Verify that the **Name** property is Tutorial Life Cycle.



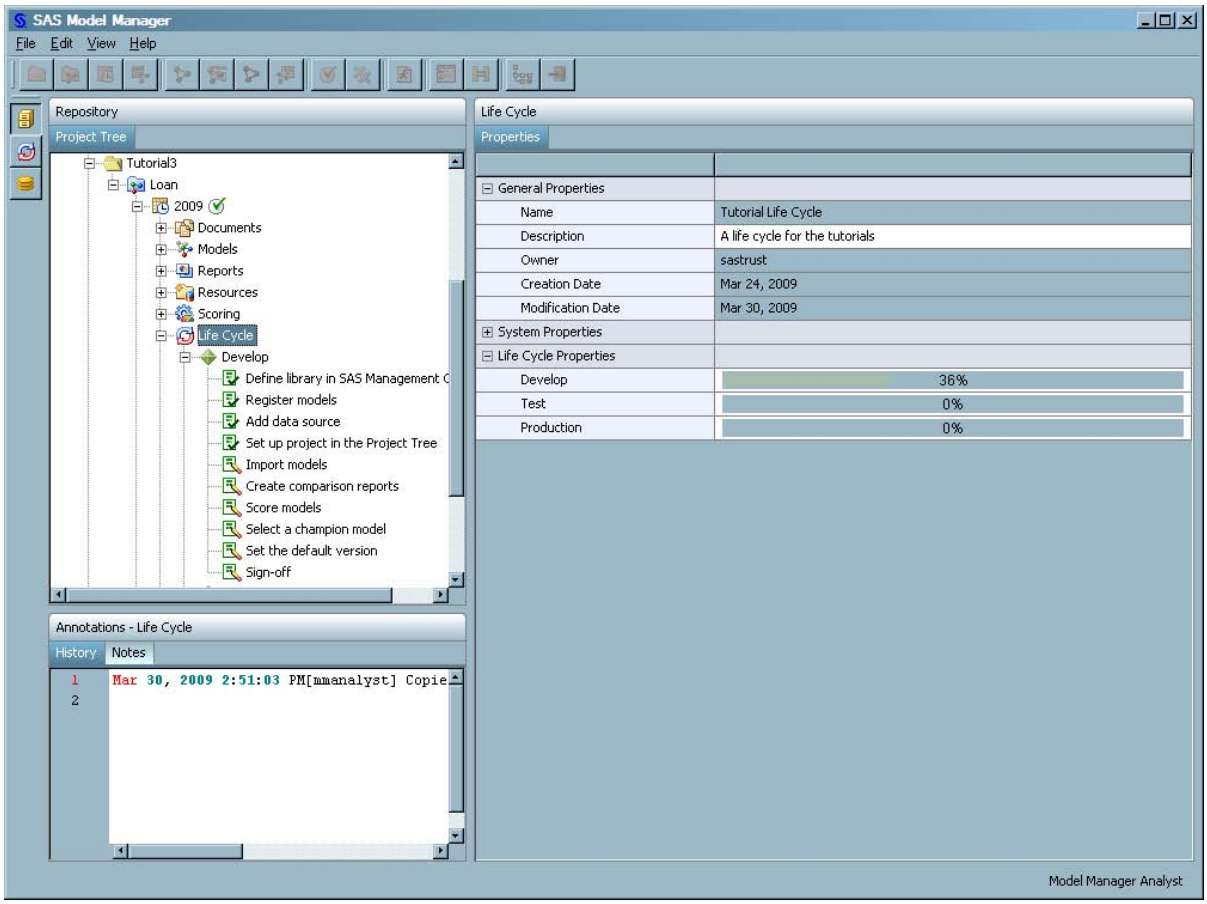
Update the Life Cycle

To complete the milestone task of adding data sources and setting up the project in the Project Tree, follow these steps:

1. In the **Loan** project, expand **2009** ⇨ **Life Cycle** ⇨ **Development**.
2. Select the **Define library in SAS Management Console** task and examine the task properties. The **To Be Complete By** property, assigned in the life cycle template, determines which users or groups from the **Participants** list in the template editor are responsible for this milestone task. Because you are a member of the **MM Tutorial Assignees** group, you are authorized to update the task status.
3. Click the **Status** field and select **Completed**.
4. Select the **Register Tables in SAS Management Console** task and examine the task properties. Click the **Status** field and select **Completed**.
5. Select the **Add data source** task. Click the **Status** box and select **Completed**.
6. Select the **Set up project in the Project Tree** task. Click the **Status** box and select **Completed**.
7. Select any object in the Project Tree to refresh the life cycle properties. Select all of the tasks whose status you updated and examine the properties. Verify that the value

of the **Completed Date** property is today and that the value of the **Completed By** property is your user ID.

8. Select the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Develop** property displays a bar chart that shows the percentage of completed tasks for this milestone.



Import Models

In this exercise you import models into SAS Model Manager from the SAS Metadata Repository, a PMML model file, and a SAS Enterprise Miner package file. Then you map the model variables. Before you import the model, verify that the model type is identical to the value of the project **Model Function** property, classification. For more information, see "Create a New Project" on page 60.

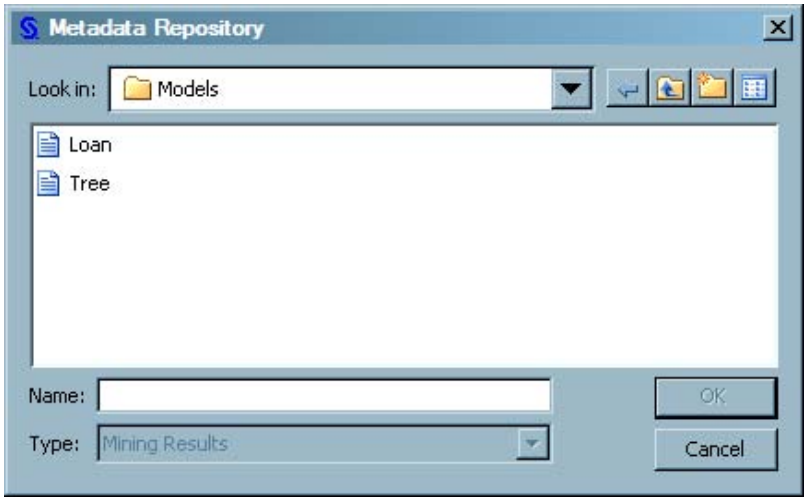
Import Models from a Metadata Repository

If your SAS Enterprise Miner 5.1 (or higher) model files are registered in the SAS Metadata Repository, then you can use SAS Model Manager to import the files. If you do not have SAS Enterprise Miner models that have been registered in the SAS Metadata Repository, you can review these steps.

To import a model that is registered with SAS Enterprise Miner, follow these steps to understand the process:

1. Expand the **2009** version in the **Loan** project and right-click the **Models** folder. Then select **Import From** ⇨ **Metadata Repository**. The Metadata Repository dialog box appears.
2. Navigate to the location of the folder that contains the SAS Enterprise Miner models. Select a model from the folder.

   Note: You can import only one model at a time in the Metadata Repository dialog box.



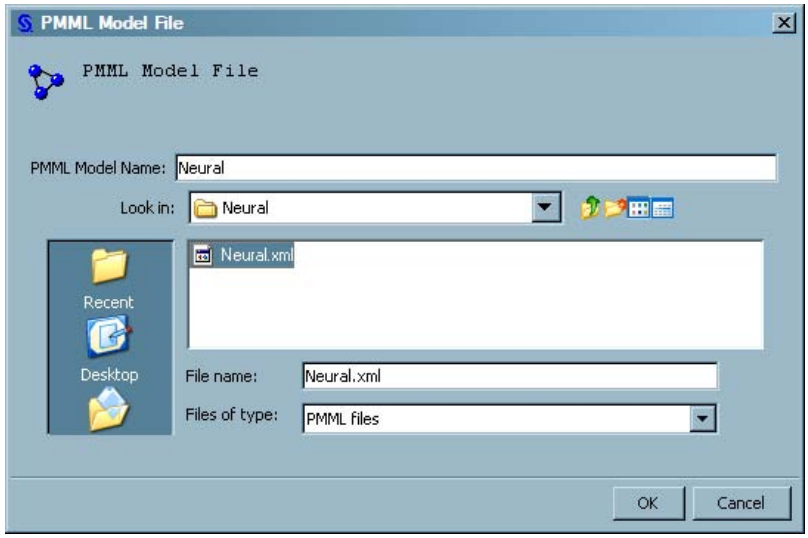
3. Click **OK**. After the SAS Model Manager processes the request to import the model, the new model appears in the **Models** folder of your project's version.

Import PMML Models

PMML (Predictive Modeling Markup Language) 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 use SAS Model Manager to import PMML 3.1 (or higher) models that are produced by another software application, such as SAS Enterprise Miner.

To import a PMML model, follow these steps:

1. Expand the **2009** version in the **Loan** project and right-click the **Models** folder. Then select **Import From** ⇨ **PMML Model File**. The PMML Model File dialog box appears.
2. In the **Model Name** field, enter **`Neural`**.
3. Navigate to the location of the folder that contains the PMML files. For this example, use **`<drive:>\Tutorial3\Samples\Neural\`** that was created when you extracted the tutorial files in a previous step. For more information, see "The Required Tutorial Files" on page 56.
4. Select the **`Neural.xml`** file and click **OK**.



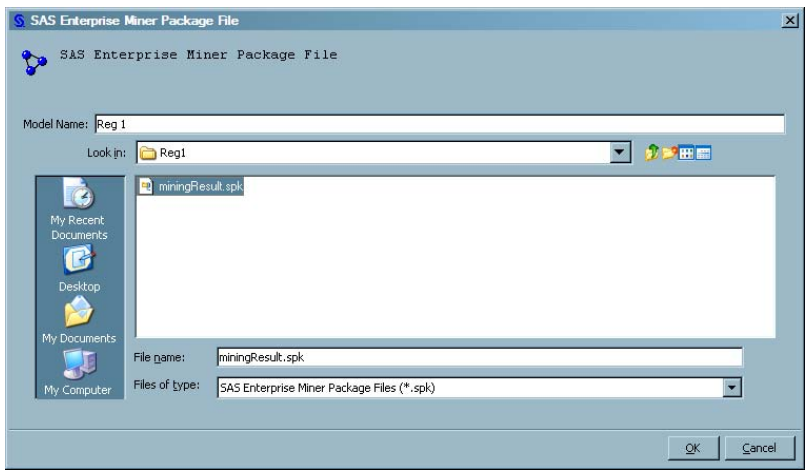
5. Examine the **Models** folder to verify that it contains the models. Right-click the folder and select **Expand All Items** to examine the model file.

Import SAS Enterprise Miner Model Package Files

SAS Enterprise Miner package files, or SPK files, contain complete model information. You can import SAS Enterprise Miner models even if they are not registered in the SAS Metadata Repository. For information about how to create a package file, see the *SAS Model Manager: User's Guide*.

To import a SAS Enterprise Miner model that was saved as package file, follow these steps:

1. Expand the **2009** version in the **Loan** project and right-click the **Models** folder. Then select **Import From** ⇨ **SAS Enterprise Miner Package File**. The SAS Enterprise Miner Package File dialog box appears.
2. In the **Model Name** field, enter **`Reg 1`**.
3. Navigate to the location of the folder that contains the SAS Enterprise Miner package files. For this example, use **`<drive:>\Tutorial3\Samples\Reg1`** that was created when you extracted the tutorial files in a previous step. For more information, see "The Required Tutorial Files" on page 56.
4. Select the **`miningResult.spk`** file and click **OK**.



5. Repeat the above steps to import a second package file that is located in `<drive:>\Tutorial3\Samples\Tree1`. Name the model **Tree 1**.
6. Examine the **Models** folder to verify that it contains the models. Right-click the folder and select **Expand All Items** to examine the model files.

Map Model Variables to Project Variables

When the names for the model output variable are not identical to the names for the project output variables, you must map the variables.

To map model output variables to project output variables, follow these steps:

1. Map model variables for the first model. Select **Reg 1** in the **Models** folder, click the **Model Mapping** tab in the right pane, and set the following mapping:

Project Variables	Model Variables
score	EM_EVENTPROBABILITY

2. Map model variables for the second model. Select **Tree 1** in the **Models** folder, click the **Model Mapping** tab in the right pane, and set the following mapping:

Project Variables	Model Variables
score	EM_EVENTPROBABILITY

Update the Model Life Cycle

To complete the milestone task for adding the models, follow these steps:

1. In the **Loan** project, expand **2009** ⇨ **Life Cycle** ⇨ **Development**.
2. Select the **Import models** task. Select the **Status** box and select **Completed**. Select any node in the Project Tree to refresh the view and select **Import models**. The **Completed Date** and **Completed By** fields have been updated with today's date and your user ID.
3. Click the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Development** property displays a bar chart that shows the percentage of completed tasks for this milestone.

Declare a Champion Model

In this exercise, you declare a champion model in the model and version folders.

Set the Champion Model

To assign a champion model, follow these steps:

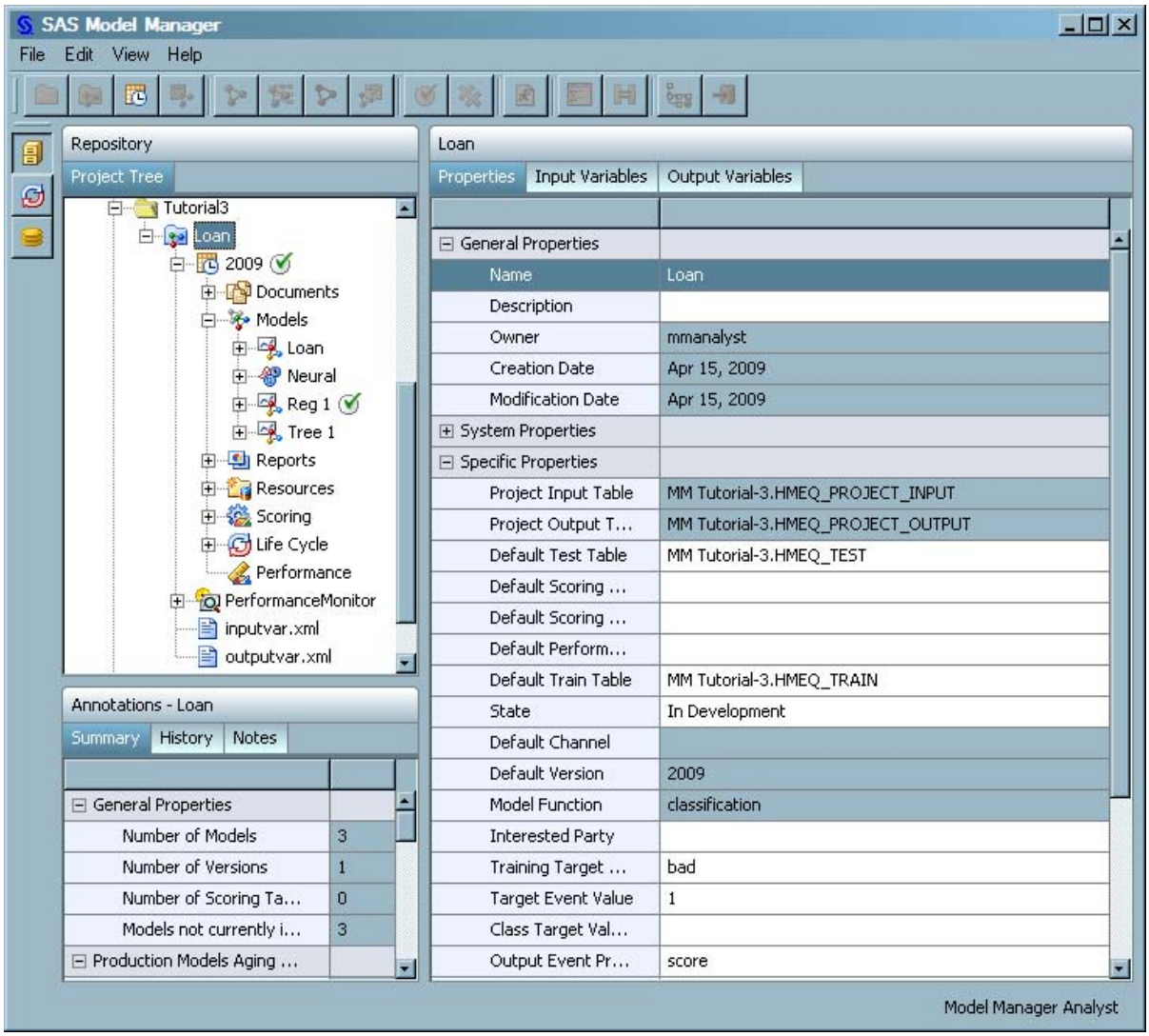
1. Expand the **Models** folder in the **2009** version for the **Loan** project. Right-click **Reg 1**, select **Set Champion Model**, and click **Yes** to confirm.
2. Verify that the icon appears next to the champion model.

Set Default Version

You assign a default version after the default champion model for the project is identified. To set the default version, follow these steps:

1. Right-click the **2009** version and select **Set Default Version**. Click **Yes** to confirm.
2. Verify that the icon appears next to the version folder.
3. Select the project folder to examine its properties. The value for **Modification Date** is today's date. The value for the **Default Version** is the name of the version folder.

   Note: The champion model for the default version is used as the champion model for the project.



TIP SAS Model Manager automatically annotates the **History** tab. To document the reasons or assumptions for your selection of the default version, use the project **Notes** tab. For more information, see "Using the Annotation View" on page 44.

Update the Life Cycle

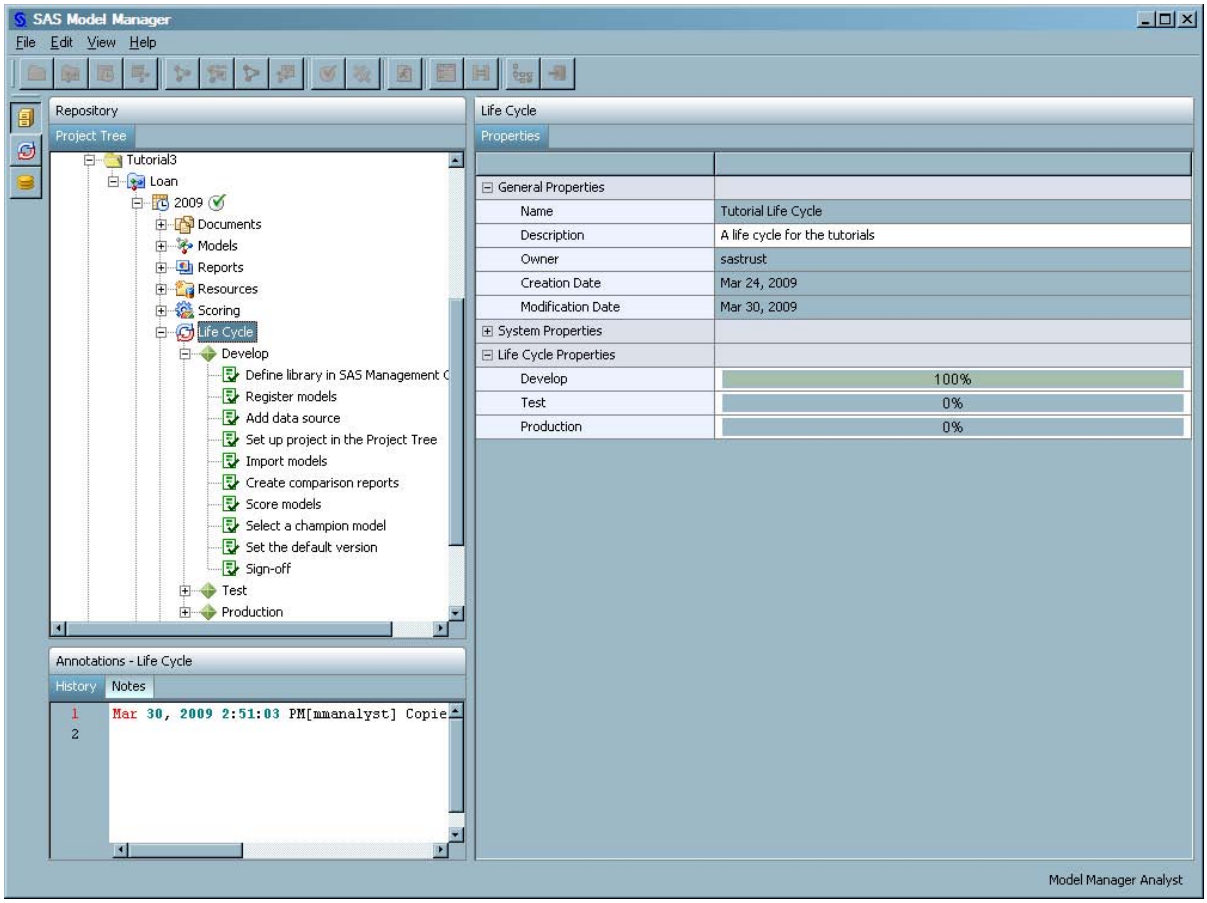
To update the life cycle milestones, follow these steps:

1. In the **Loan** project, expand **2009** ⇨ **Life Cycle** ⇨ **Development**.
2. Select the **Create comparison reports** task. Click the **Status** box and select **Completed**.

   Note: Although this task was not part of this exercise, dependencies in the life cycle require that you mark this task complete.
3. Select the **Score models** task. Click the **Status** box and select **Completed**.

   Note: Although this task was not part of this exercise, dependencies in the life cycle require that you mark this task complete.
4. Select the **Select a champion model** task. Click the **Status** box and select **Completed**.

5. Select the **Set the default version** task. Click the **Status** box and select **Completed**.
6. Select the **Sign-off** task. Click the **Status** box and select **Completed**.
7. Click the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Development** property displays a bar chart that shows the percentage of completed tasks for this milestone.



Deliver Models

In this exercise you use the comprehensive publishing environment for model delivery to share models. Model delivery most often includes model score code and its associated input and output metadata. You export a model and the champion model for a project to the SAS Metadata Repository and publish a model to a publish channel. For each model you export, SAS Model Manager creates a MiningResult object in the SAS Metadata Repository. Only models that have a **Score Code Type** of **Data Step** can be exported to the metadata repository. Application software, such as SAS Data Integration Studio or SAS Enterprise Guide, can access the MiningResult object through the SAS Metadata Server and submit on-demand or batch scoring jobs.

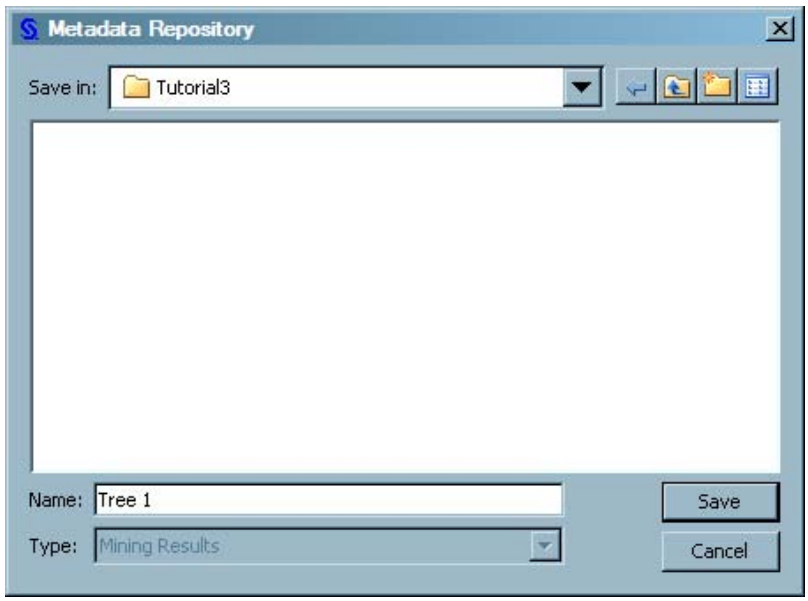
Export a Model to the SAS Metadata Repository

SAS Model Manager uses the SAS Folder view to export the model to any folder that is accessible to the user. You can export a model to folders in the SAS Foundation repository

or to folders in custom repositories that are created in SAS Management Console to reflect the structure of your business organization.

To export a model to a SAS Metadata Repository, follow these steps:

1. Expand **Loan** ⇨ **2009** ⇨ **Models** and right-click the **Tree 1** model. Then select **Export Model** ⇨ **Export Model into SAS Metadata Repository**. The Metadata Repository dialog box appears
2. Navigate to the folder where you want to store the model.



3. Enter **Tree 1** as the name and click **Save**. If a MiningResult object is in the repository that has the same name, then you are prompted to decide whether to overwrite the metadata for this stored object.

   CAUTION:
   Do not overwrite an existing MiningResult object unless you are certain that the model is from the same project in SAS Model Manager.

4. In the Information dialog box, click **OK**.

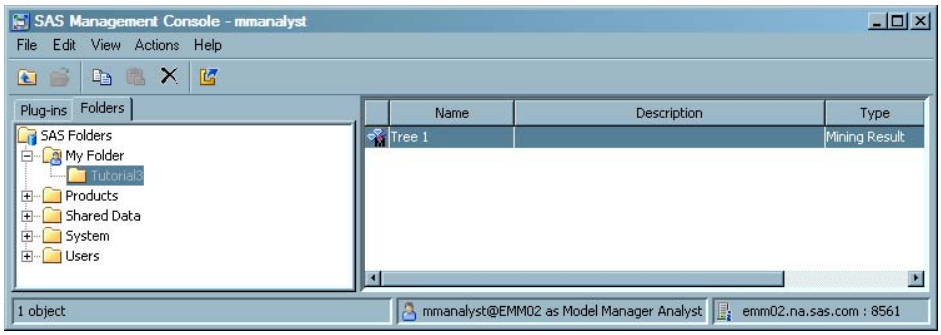
If you change the score code for the model, then export the model again to ensure that your score application uses the current scoring code.

Verify the Model Export

To verify that SAS Model Manager successfully created the MiningResult object in the metadata repository for an exported model, use SAS Management Console. To view the contents of the exported model or project, you can use SAS Data Integration Studio. You can also use SAS Management Console to export the MiningResult object to a SAS package.

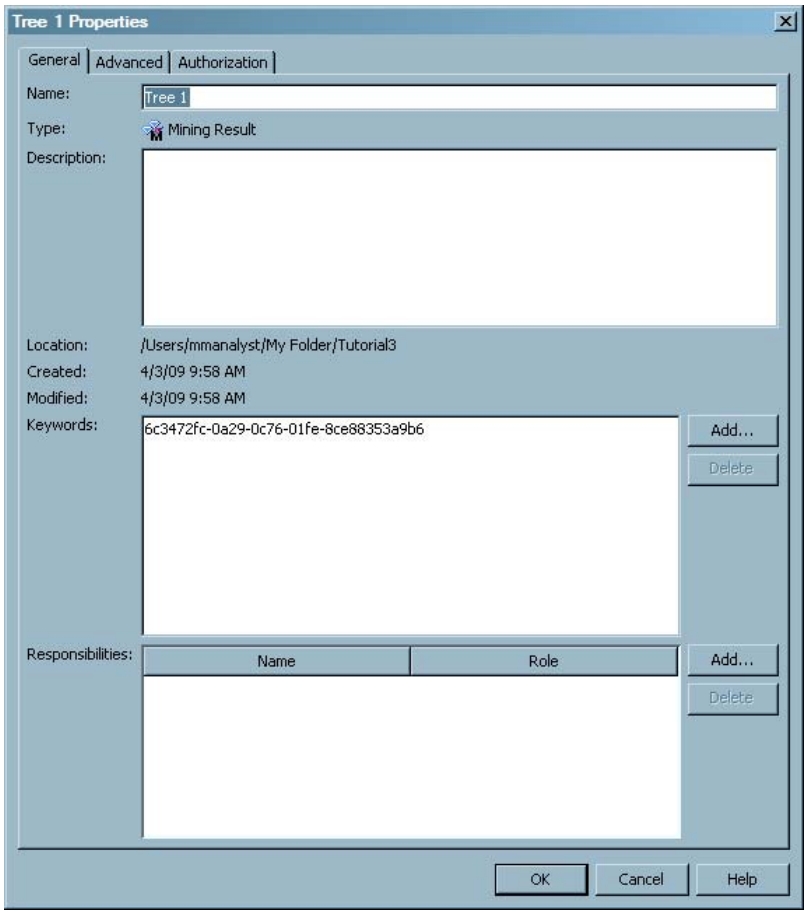
To view a MiningResult object in the metadata repository, follow these steps:

1. Open SAS Management Console and log on to the SAS Metadata Server using the same user ID that you use to log on to SAS Model Manager.
2. Use the **Folders** tab and expand the folders to locate the model that you exported. When you select the folder, the right pane lists the MiningResult objects for the exported models.



3. Right-click the **Tree 1** MiningResult object and select **Properties** from the pop-up menu. The Properties window opens.

4. Examine the **Keywords** box on the **General** tab to verify that the MiningResult object contains the universal unique identifier (UUID) of the exported model. The UUID is a system property that SAS Model Manager automatically assigns to each model. To view a system property in SAS Model Manager, click the + icon beside the **System Properties** heading to expand the section.



TIP You can use the UUID to conduct filtered searches and query the exported models. For more information, see the *SAS Model Manager: User's Guide*.

5. Examine the metadata on the **Advanced** tab to determine when the MiningResult object was created or most recently updated.

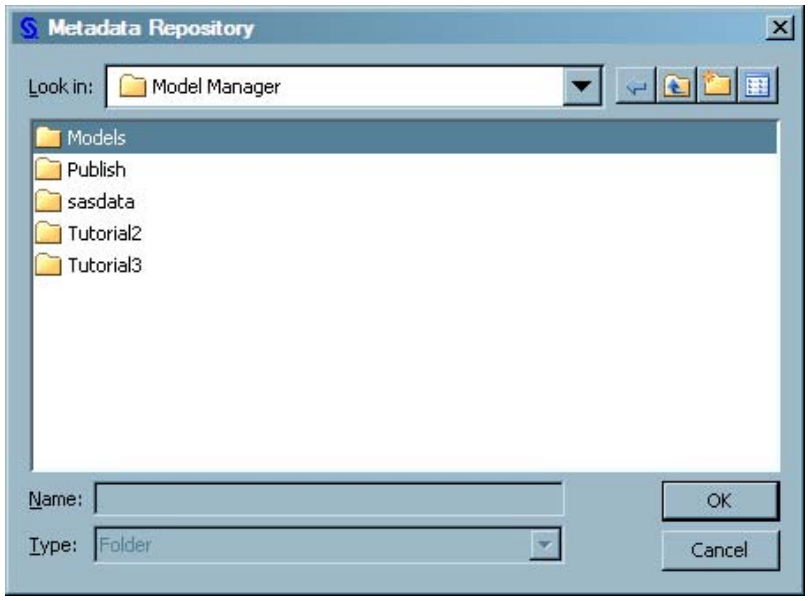
6. Click **OK**.

Export the Champion Model

To export the champion model for a project, you must have already assigned the default version for the project. SAS Model Manager examines the project and always exports the champion model in the default version. When the default version for a project changes and you export the model again at the project level, the scoring application automatically uses the latest score code. SAS Model Manager exports only models whose model property **Score Code Type** is set to **Data Step**.

To export the champion model for a project, follow these steps:

1. Verify that the project has a default version assigned. Select the **Loan** project folder to examine its properties. The **Default Version** property contains the name of the default version.
2. Right-click the **Loan** project and select **Export Project Champion Model** from the pop-up menu. The Metadata Repository dialog box appears.
3. Navigate to the folder where you want to store the model.



4. Select the folder and click **OK**. If a MiningResult object is in the repository that has the same name, then you are prompted to decide whether to overwrite the metadata for this stored object.

   CAUTION:
   Do not overwrite an existing MiningResult object unless you are certain that the model is from the same project in SAS Model Manager.

5. In the Information dialog box, click **OK**.

Publish Models

SAS Model Manager uses SAS Publishing Framework to publish models to defined channels. SAS Model Manager creates a SAS Package file (SPK) for the model in a publication channel. Authorized users who subscribe to the channel can choose to receive e-mail notifications when updated models are ready to deploy to testing or production scoring servers and when the SPK file is published to a publication channel. Then you can

extract and validate the scoring logic, deploy champion models to a production environment, and monitor the performance of your models.

To publish a model to a channel, follow these steps:

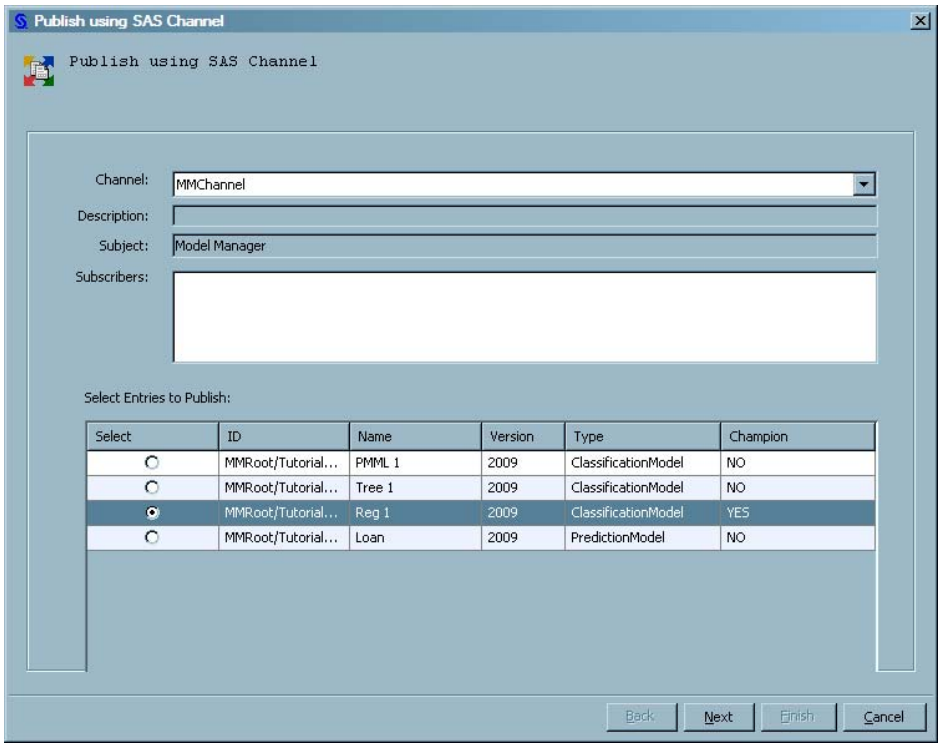
1. Expand **Loan** ⇨ **2009** and right-click the **Models** folder. Then select **Publish**. The Channel Usage window opens.

   TIP You can publish models from the organization, project, version, or model folder in the Project Tree.

2. Select a publication channel from the **Channel** list.

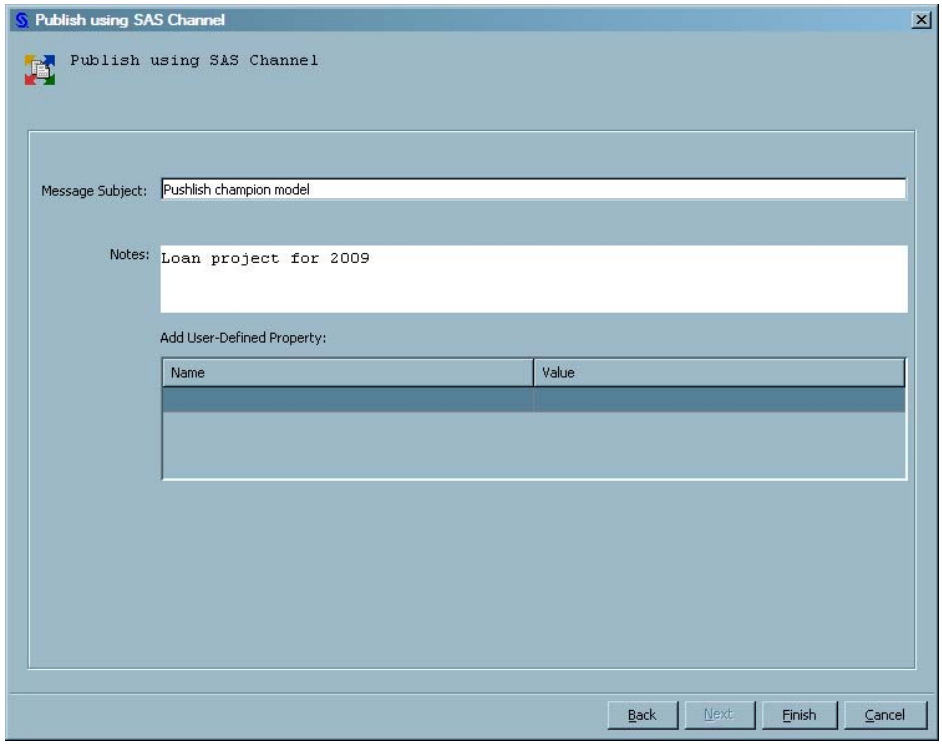
   Note: The channel values for **Description**, **Subject**, and **Subscribers** are defined in the SAS Metadata Repository with SAS Management Console.

3. Select **Reg 1** as the model to publish in the **Select Entries to Publish** table. SAS Model Manager lists all of the models in the version folder. To view the entire folder name, expand the ID column heading. Click **Next**.

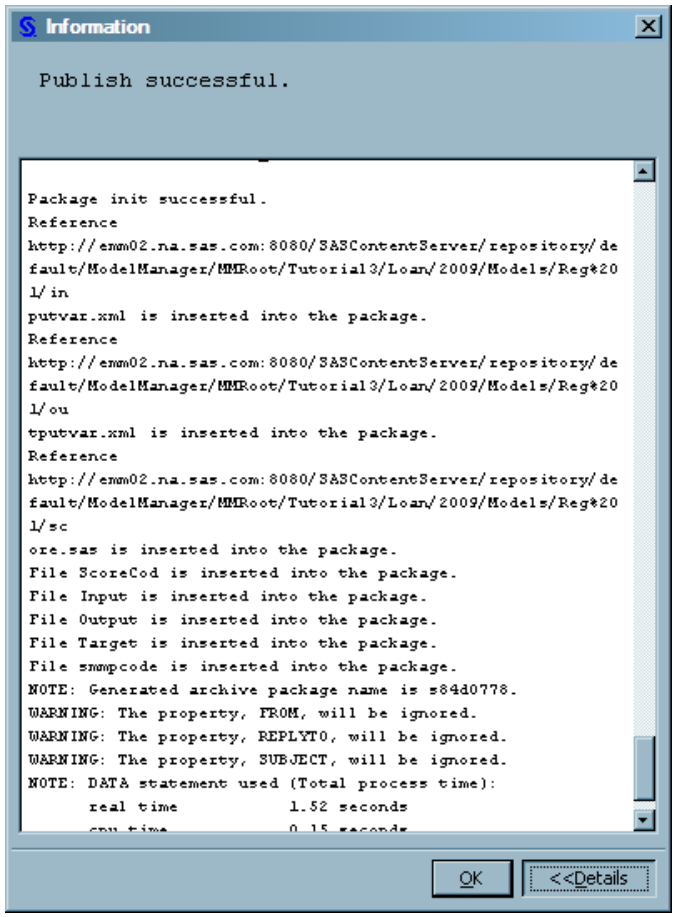


4. Specify an optional subject line for the e-mail message in the **Message Subject** box. For this example, enter `Publish champion model`. By default, SAS Model Manager uses the value that is defined in the publication channel. If you omit the subject line, then the name of the published model is used.

5. In the **Notes** box include information about the model that might be useful to other users who are involved with the project. For this example, enter `Loan project for 2009`.



6. Click **Finish**. The Information dialog box opens and provides information about whether SAS Model Manager successfully published the model. Click **Details** to display a log of the publication process and any messages.



7. Click **OK**.

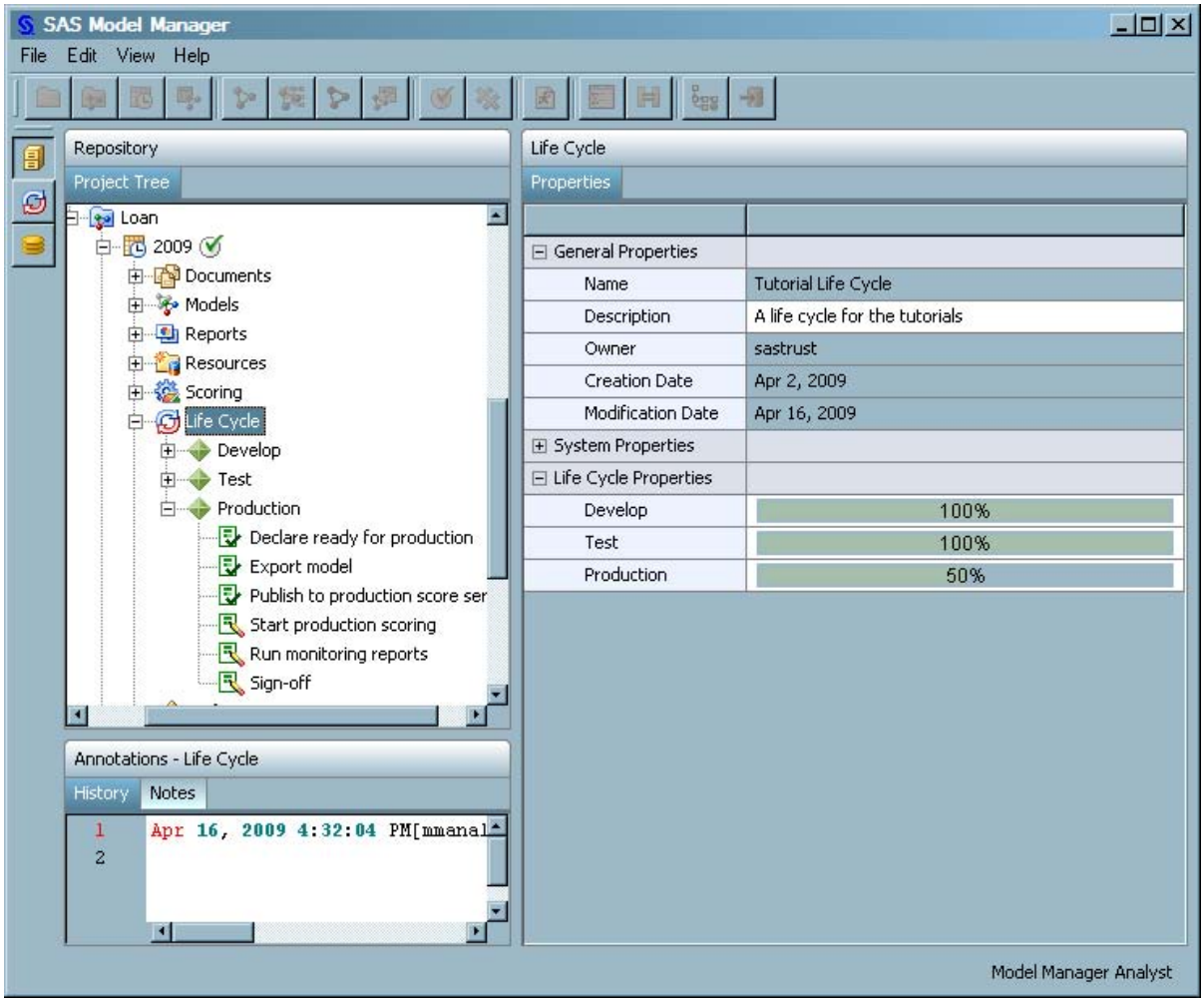
The SAS package that is sent to the publication channel contains the model input, output, SAS code, and its properties. You can submit a SAS DATA step program that calls the SAS Publish API (Application Programming Interface) to extract and deploy the model to a testing or scoring server. SAS Model Manager also provides a SAS macro program, called MM_GetModels, that extracts the SAS code and metadata to score the model. Typically, extracted files are placed on a local drive of the scoring server that is used to deploy the published model. For more information, see the *SAS Model Manager: User's Guide*.

Update the Life Cycle

To update the life cycle milestones, follow these steps:

1. In the **Loan** project, expand **2009** ⇨ **Life Cycle** ⇨ **Test**.
2. Select each milestone task for **Test**. Click the **Status** box and select **Completed**.

   Note: Although this task was not part of this exercise, dependencies in the life cycle require you to mark this task complete.
3. Expand the **Production** milestone. Select the **Declare ready for production** task. Click the **Status** box and select **Completed**.
4. Select the **Export model** task. Click the **Status** box and select **Completed**.
5. Select the **Publish to production score server** task. Click the **Status** box and select **Completed**.
6. Click the **Life Cycle** node to examine its properties. The value for **Modification Date** is today's date. The **Test** and **Production** properties display bar charts that show the percentage of completed tasks for these milestones.


SAS Model Manager
File Edit View Help
Repository
Project Tree
Loan
2009
Documents
Models
Reports
Resources
Scoring
Life Cycle
Develop
Test
Production
Declare ready for production
Export model
Publish to production score ser
Start production scoring
Run monitoring reports
Sign-off
Annotations - Life Cycle
History
Notes
1 Apr 16, 2009 4:32:04 PM[mmanal
2
Life Cycle
Properties
General Properties
Name Tutorial Life Cycle
Description A life cycle for the tutorials
Owner sastrust
Creation Date Apr 2, 2009
Modification Date Apr 16, 2009
System Properties
Life Cycle Properties
Develop 100%
Test 100%
Production 50%
Model Manager Analyst

Chapter 5

Tutorial 4: Using Advanced Reporting

Overview of Advanced Reporting

The advanced reporting capability of SAS Model Manager enables you to create two different types of reports.

- User-defined reports enable your company to add enterprise-specific reports to the existing list of reports that are available via the New Report Wizard dialog box.
- Ad hoc reports enable you to create one-of-a-kind reports as you need them.

To make it easier to create these reports, SAS Model Manager provides a number of SAS macro variables and SAS macro programs. These macros can be used to gain access to model-specific information as well as to more general folder and user information.

This tutorial shows you the basic tasks that are involved in creating new SAS Model Manager reports. It contains examples and step-by-step directions about setting up and running ad hoc and user-defined reports.

Prerequisites

Models Used in Tutorial 2

The exercises in this tutorial depend upon some of the properties of the specific models that were added in Tutorial 2. While it should be possible to substitute other projects, versions, or models in place of those mentioned in the following actions, this tutorial is

written to follow Chapter 3, "Tutorial 2: Performing Basic SAS Model Manager Tasks," on page 23.

The Required Tutorial Files

The SAS programs that are required for this tutorial are on your local computer after you extract them from the ZIP file SMM22Tutorial.zip. If you have not extracted the tutorial files, see "The Tutorial Files" on page 3.

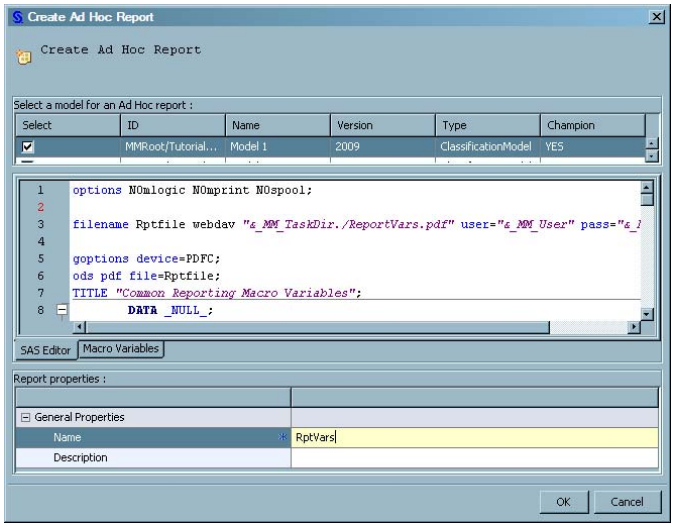
This tutorial requires the following files in the **`<drive>\Tutorial4\Samples`** folder:

- RptVars.sas
- ScoreRange.sas
- ScoreRangeMacro.sas
- ScoreRangeTemplate.xml

Create a Simple Ad Hoc Report

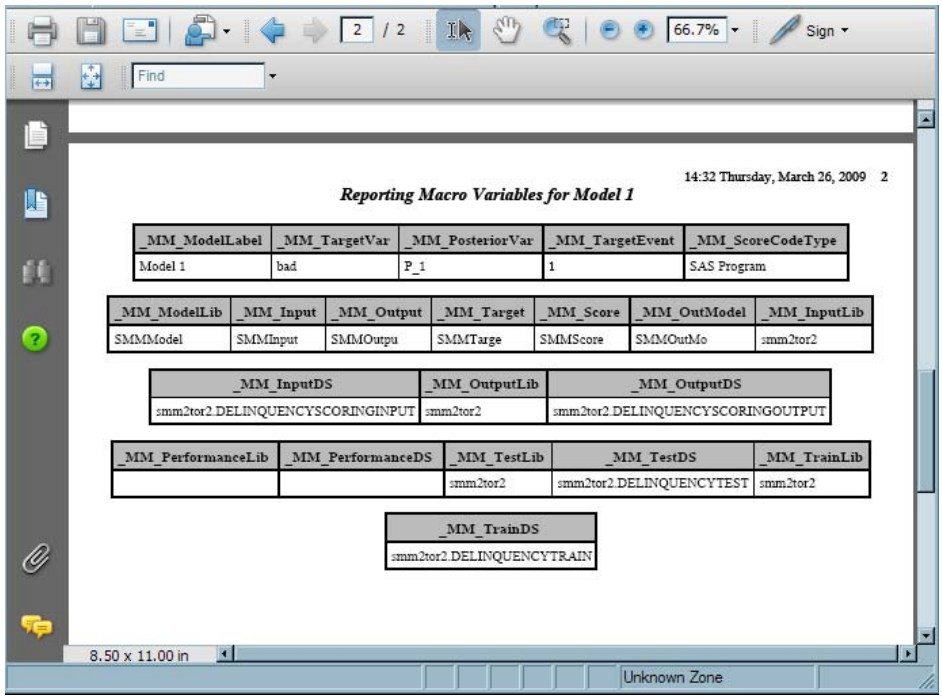
In this exercise, you create a PDF file to display the macro variables that are available in the SAS Model Manager reporting environment:

1. In the **`<drive>/Tutorial4/Samples`** folder, open the tutorial example report RptVars.sas in a text editor.
2. Copy the code from RptVars.sas.
3. Log on to SAS Model Manager.
4. Open the Create Ad Hoc Report window:
   a. Expand the **Tutorial2** folder, the **Delinquency** project, and the **2009** version.
   b. Right-click the **Reports** folder and select **Reports ⇨ Create Ad Hoc Report**.
5. In the Create Ad Hoc Report window, select **Model 1** in the **Select a Model for Ad Hoc Report** table.
6. In the **SAS Editor**, paste the code that you copied in Step 2.
7. In the **Name** field of the **Report Properties** table, enter **`RptVars`**.



8. Click, **OK**. SAS Model Manager creates the report and highlights the report in the **Reports** folder.
9. To view the report, expand the new report **RptVars**, right-click **RptVars.pdf,** and select **Open Item**.

Here is page 2 of the PDF report output:



14:32 Thursday, March 26, 2009 2

Reporting Macro Variables for Model 1

_MM_ModelLabel	_MM_TargetVar	_MM_PosteriorVar	_MM_TargetEvent	_MM_ScoreCodeType
Model 1	bad	P_1	1	SAS Program

_MM_ModelLib	_MM_Input	_MM_Output	_MM_Target	_MM_Score	_MM_OutModel	_MM_InputLib
SMMModel	SMMInput	SMMOutpu	SMMTarge	SMMScore	SMMOutMo	smm2tor2

_MM_InputDS	_MM_OutputLib	_MM_OutputDS
smm2tor2.DELINQUENCYSCORINGINPUT	smm2tor2	smm2tor2.DELINQUENCYSCORINGOUTPUT

_MM_PerformanceLib	_MM_PerformanceDS	_MM_TestLib	_MM_TestDS	_MM_TrainLib
		smm2tor2	smm2tor2.DELINQUENCYTEST	smm2tor2

_MM_TrainDS
smm2tor2.DELINQUENCYTRAIN

Create an Ad Hoc Score Range Report

In this exercise you create an ad hoc report to categorically display score ranges in an HTML report. To create output in HTML from an ad hoc report, ensure that the report code is enclosed by a SAS Model Manager formatting macro. To do this, follow these steps:

1. In the **`<drive>/Tutorial4/Samples`** folder, open the example report ScoreRange.sas and copy the code.
2. If necessary, log on to SAS Model Manager.
3. In the Project Tree, expand the **Tutorial2** folder, the **Delinquency** project, and the **2009** version.
4. Right-click the **Reports** folder and select **Reports** ⇨ **Create Ad Hoc Report**.
5. In the Create Ad Hoc Report window, select **Model 1** in the **Select Model for Ad Hoc Report** table.
6. In the **SAS Editor**, paste the code that you copied in Step 1.
7. Modify the Score Range code to format the report in HTML.

   The ScoreRange.sas program uses the SAS Model Manager formatting macros, which enable user reports to be formatted in PDF, HTML, RTF, and Excel. A beginning

formatting macro code precedes the report code. The ending formatting macro must be the last line of code in the report program.

a. Add the argument **`reportFormat=html`** to the %MM_ExportReportsBegin macro argument list. Here is the modified macro:

```
%MM_ExportReportsBegin(reportFormat=html, fileName=ScoreRange);
```

b. Add the argument **`reportFormat=html`** to the %MM_ExportReportsEnd macro argument list. Here is the modified macro:

```
%MM_ExportReportsEnd(reportFormat=html);
```

8. In the **Name** field of the **Report Properties** table, enter **`ScoreRange`**.
9. Click **OK**. SAS Model Manager runs the report and creates the **ScoreRange** folder under the **Reports** folder.
10. To view the **ScoreRange** report, expand the **ScoreRange** folder, right-click **ScoreRange.html**, and select **Open Item**.

Here is the output from the FREQ procedure as a table and as a graph:

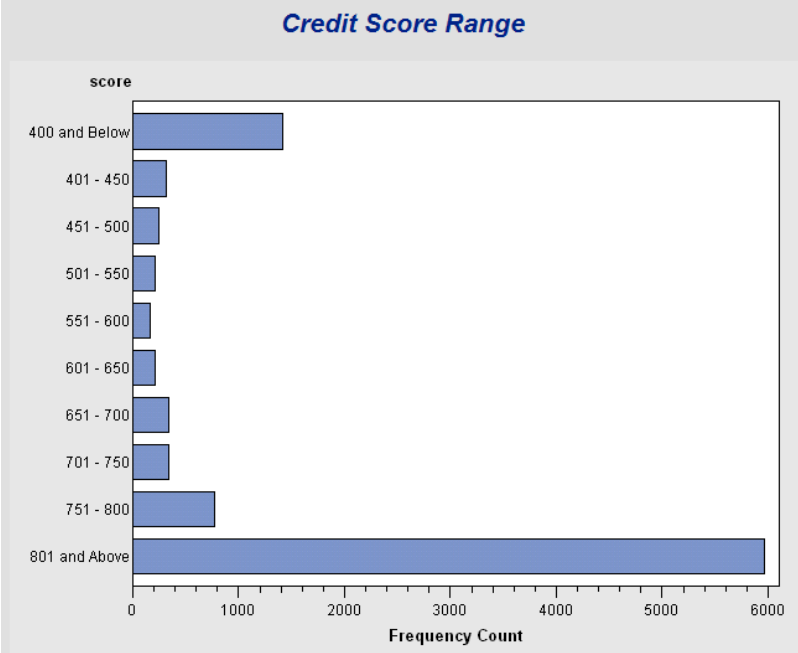
Display 5.1 *The Score Range Report Table*

Credit Score Range

The FREQ Procedure

score	Frequency	Percent	Cumulative Frequency	Cumulative Percent
400 and Below	1419	14.19	1419	14.19
401 - 450	322	3.22	1741	17.41
451 - 500	249	2.49	1990	19.90
501 - 550	206	2.06	2196	21.96
551 - 600	161	1.61	2357	23.57
601 - 650	213	2.13	2570	25.70
651 - 700	340	3.40	2910	29.10
701 - 750	344	3.44	3254	32.54
751 - 800	777	7.77	4031	40.31
801 and Above	5969	59.69	10000	100.00

Display 5.2 *The Score Range Report Graph*



Install a User-defined Score Range Report

In this exercise you install the Score Range report to the middle-tier server that hosts the SAS Analytics Platform server. After this server is restarted, you can run the Score Range report from the New Reports Wizard. This exercise has two parts. In the first exercise, you prepare and install the report files to the middle-tier server. In the second exercise, you create a Score Range report from the New Reports Wizard.

Install a User-defined Report

To install the Score Range report to the New Reports Wizard, follow these steps:

1. From the **<*drive*>/Tutorial4/Samples** folder, copy these files to the appropriate user template directories:

File	Directory
ScoreRangeTemplate.xml	***sasconfigdir*\Lev#\AnalyticsPlatform \apps\ModelManager\ext**
ScoreRange.sas ScoreRangeMacro.sas	***sasconfigdir*\Lev#\AnalyticsPlatform \apps\ModelManager\ext\SASCode**

You must have Write access to this directory. If you are not authorized to write to this directory, ask your system administrator to copy the file to the directory. *sasconfigdir* is a wildcard placeholder for the path to your configured SAS directory. For Windows, the default value for *sasconfigdir* is **c:\SAS\Config.** **Lev#** is a value that is selected by a user in the range of **Lev0** through **Lev9**. The default value is

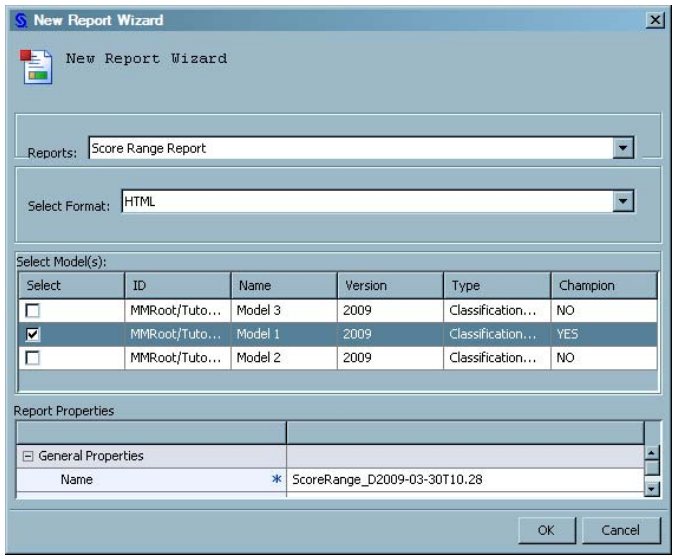
Lev1. For UNIX systems, the default home folder for SAS Model Manager is **`/usr/local/SAS`**.

2. Exit SAS Model Manager.
3. Stop and restart the middle-tier server. This must be done by a system administrator.

Run the New User-defined Report

To execute the installed score range report, follow these steps:

1. Log on to SAS Model Manager.
2. Expand the **Tutorial2** folder, the **Delinquency** project, and the **2009** version.
3. Right-click the **Reports** folder and select **Reports** ⇨ **New Report Wizard**. The New Report Wizard opens.
4. In the **Reports** box, select **Score Range Report**.
5. In the **Select Format** box, select **HTML**.
6. In the **Select Model** table, select **Model 1**.
7. If a ScoreRange report exists in the Reports folder from the previous exercise, you can name the report using the default filename. If **ScoreRange** does not exist in the **Reports** folder, type **`ScoreRange`** in the **Name** box of the **General Properties** table.



8. Click **OK**.
9. Click **OK** on the Report Execution Successful dialog box.
10. To view the new report, expand the new score range report, right-click **ScoreRange.html**, and select **Open Item**. To view the report output, see Display 5.1 on page 80 and Display 5.2 on page 81.

For more information about this task, see the *SAS Model Manager 2.2: User's Guide*.

Chapter 6

Tutorial 5: Publishing Teradata Scoring Functions

Overview of Publishing a Teradata Scoring Function

SAS Model Manager enables you to publish classification and prediction model types to the Teradata Enterprise Data Warehouse (EDW). When you publish a Teradata scoring function for a project, SAS Model Manager exports the project's champion model to the SAS Metadata Repository and calls the SAS Scoring Accelerator to create scoring functions. The scoring functions are deployed inside Teradata based on the project's champion model score code. The scoring function is then validated automatically against a default train table to ensure that the scoring results are correct. A scoring application (for example, a call center application that calls the SAS Model Manager Java Scoring API) can then execute the scoring functions in the Teradata EDW.

This tutorial shows you the tasks that are involved in publishing a project champion model to the Teradata Enterprise Data Warehouse. It contains examples and step-by-step directions about preparing Teradata for use with SAS Model Manager and publishing a scoring function.

Prerequisites

Models Used in Tutorial 3

The exercises in this tutorial depend upon some of the properties of the specific models that were added in Tutorial 3. While it should be possible to substitute other projects, versions, or models in place of those mentioned in the following actions, this tutorial is written to follow Chapter 4, “Tutorial 3: Importing and Exporting Models,” on page 55.

Prepare Teradata for Use with SAS Model Manager

To use the SAS Model Manager Publish Teradata Scoring Function, the Database Administrator (DBA) needs to prepare the Teradata Enterprise Data Warehouse (EDW). In this exercise, the DBA sets up the publishing and scoring aspects of SAS Model Manager.

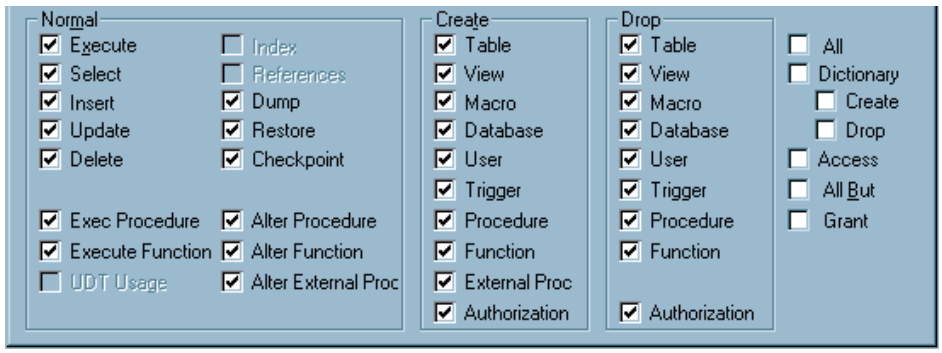
The SQL scripts that are required for this tutorial are on your local computer after you extract them from the ZIP file SMM22Tutorial.zip. If you have not extracted the tutorial files, see "The Tutorial Files" on page 3.

Note: Contact your system administrator if you do not have the appropriate permissions to the installation and configuration directories on the SAS Model Manager server.

To enable the publishing of scoring functions to Teradata from SAS Model Manager, follow these steps:

1. Create a Teradata database. A default user is created who has the same name as the database. For example if you create a database that has the name **pubs**, the default user ID is **pubs**. You need to provide the server name, user ID, and password to users, so they can publish a Teradata scoring function from SAS Model Manager.

2. Using the Teradata Administrator 12.0 tool, set the user access permissions for the Teradata database that you created in the previous step. Grant all **Normal**, **Create**, and **Drop** permissions that are available in the Grant/Revoke Objects dialog box, as shown in the following example:



3. Run the **createTables.sql** script to create the SAS Model Manager metadata tables in the Teradata database. The SQL file is located in the **Utilities** subdirectory of the SAS Model Manager In-Database Scoring for Teradata installation directory on the middle-tier server. Here is the default installation directory for a Microsoft Windows server:

   ```
   C:\Program Files\SAS
   \SASModelManagerInDatabaseScoringForTeradata\2.2
   ```

4. Run the **hmeq.sql** script to create the **hmeqid** table in the Teradata database. The SQL file is located in the`<drive>\Tutorial5` folder. To run the **hmeq.sql** file, follow these steps:

   a. Start a BTEQ session.

   b. Issue a login statement. For example:

      ```
      .login S191200/pubs,pubs1
      ```

   c. Set the scoring database as the active database in the BTEQ session. For example:

      ```
      database pubs;
      ```

d. Issue the command to run the SQL script. For example:

```
.run file="<drive>\Tutorial5\hmeq.sql"
```

5. Install the SAS 9.2 Formats Library for Teradata in the Teradata EDW. This library contains many of the formats that are available in Base SAS.

   For information about how to install and configure the SAS 9.2 Formats Library for Teradata, see the chapter on post-installation configuration for the SAS Accelerator Publishing Agent software in the *Configuration Guide for SAS 9.2 Foundation* for your operating environment.

6. Download the Teradata JDBC Driver for jar files (**terajdbc4.jar** and **tdgssconfig.jar**) for Teradata 12.0 from the Teradata Web site (http://www.teradata.com/downloadcenter) and place the jar files in the ***sasconfigdir*\Lev#\AnalyticsPlatform\apps\ModelManager\lib** directory.

7. Send a request to your system administrator to restart the SAS Analytics Platform to finish the installation of the Teradata JDBC jar files.

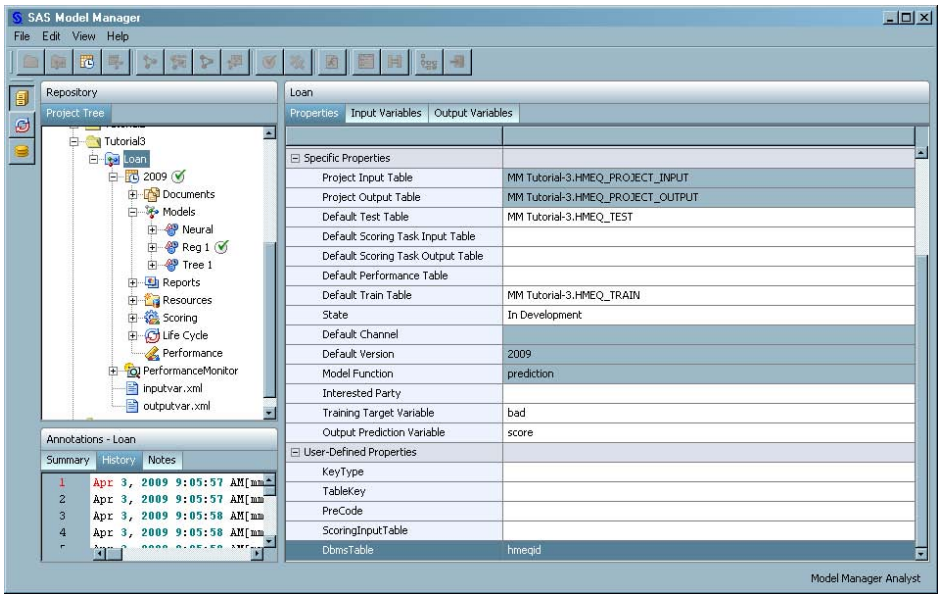
Publish a Teradata Scoring Function

In this exercise, you publish a scoring function for a project's champion model to the Teradata EDW.

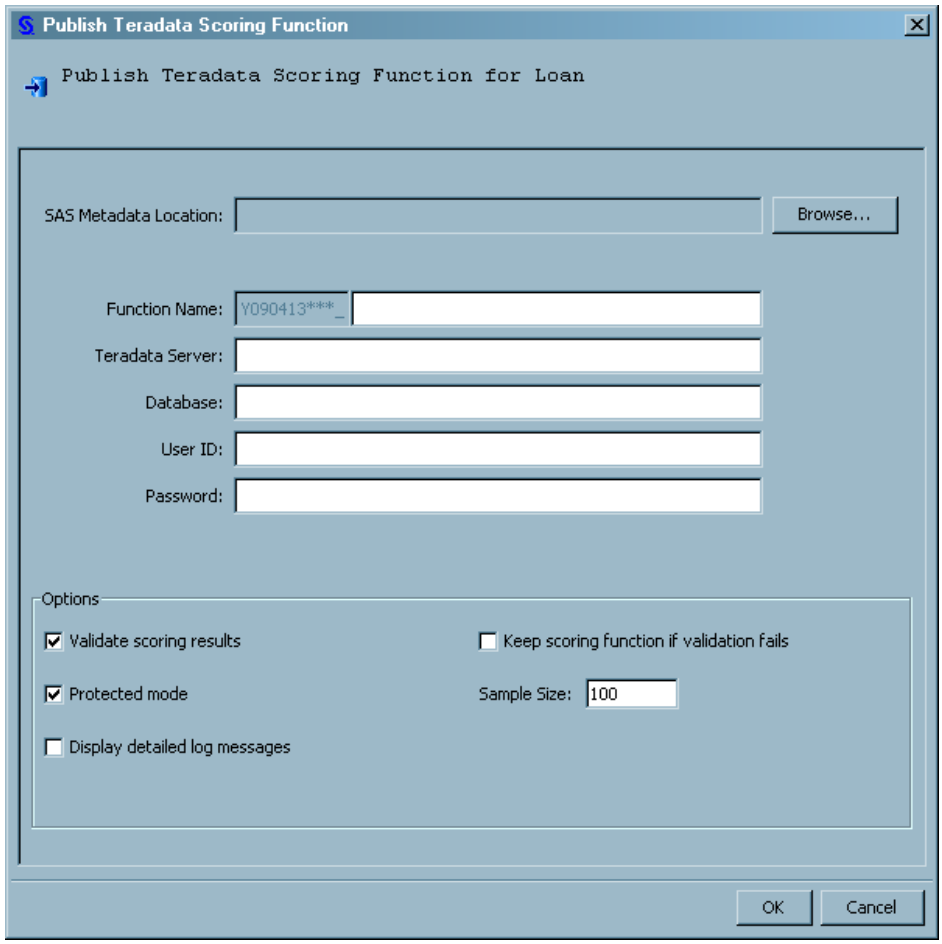
To publish a Teradata scoring function, follow these steps:

1. Verify that you have set the default version for the project and have set the champion model for the default version in Tutorial 3.

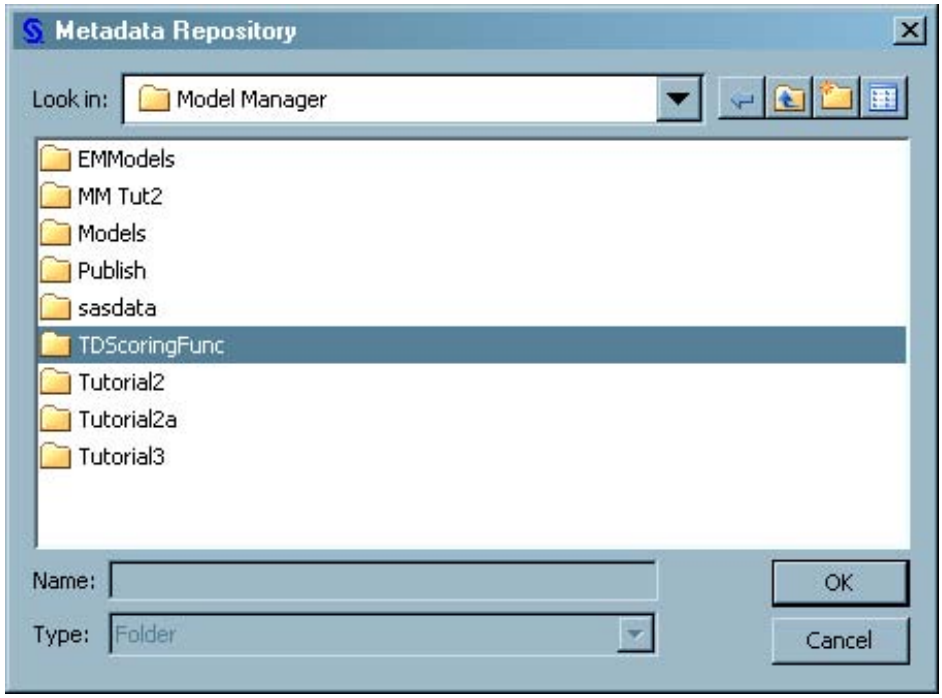
2. Select the **Loan** project folder and enter a value for the **DbmsTable** user-defined property. This value is the scoring input table that was created by the DBA in the Teradata database.



3. Right-click the **Loan** project in the Project Tree and select **Publish Teradata Scoring Function**. The Publish Teradata Scoring Function window opens.



4. Select a location in which to publish the model. Click **Browse**, select a folder name, and then click **OK**.

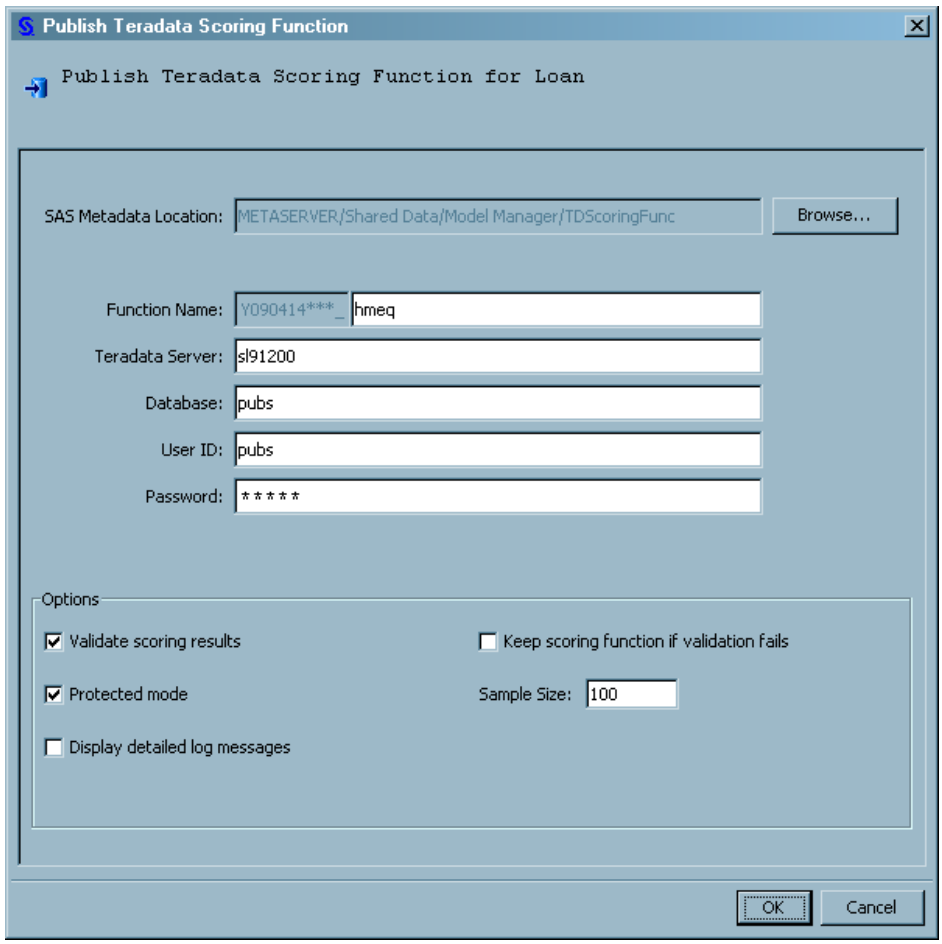


5. Enter a name for the scoring function, using the following naming conventions:

- The user-defined value must be unique across all projects.
- The maximum length for the user-defined value is 19, and no spaces are allowed.
- The only special character that can be included in the function name is an underscore.

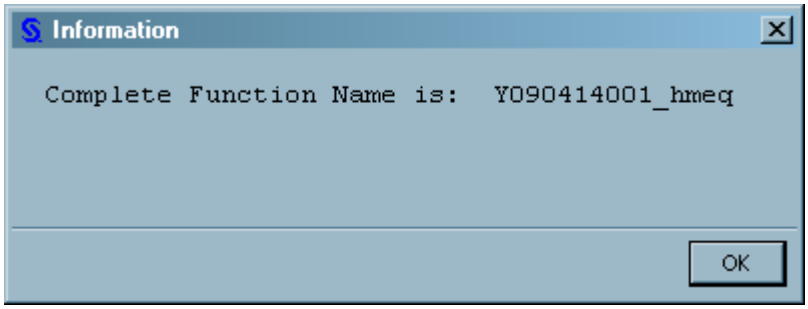
6. For the following fields, enter the values for the Teradata database that your DBA created for this tutorial:
   - **Teradata Server**
   - **Database**
   - **User ID**
   - **Password**
7. In the **Options** section, select one of the following check boxes for the desired validation options.
   - **Validate scoring results**
   - **Keep scoring function if validation fails**
   - **Protected mode**
   - **Display detailed log messages**

   Note: The **Validate scoring results** and **Protected mode** options are selected by default.
8. Enter a numeric value for **Sample Size**. The default sample size is 100.

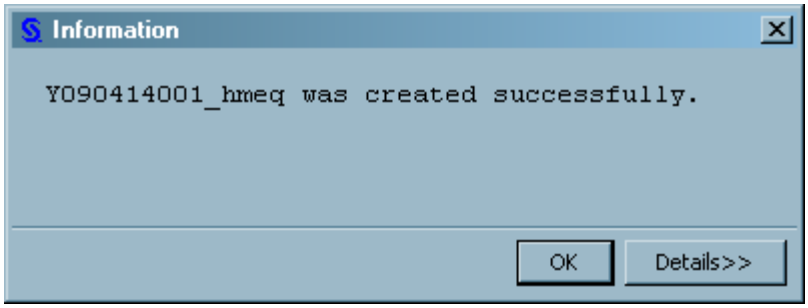


9. Click **OK**. A message is displayed that contains the scoring function name.

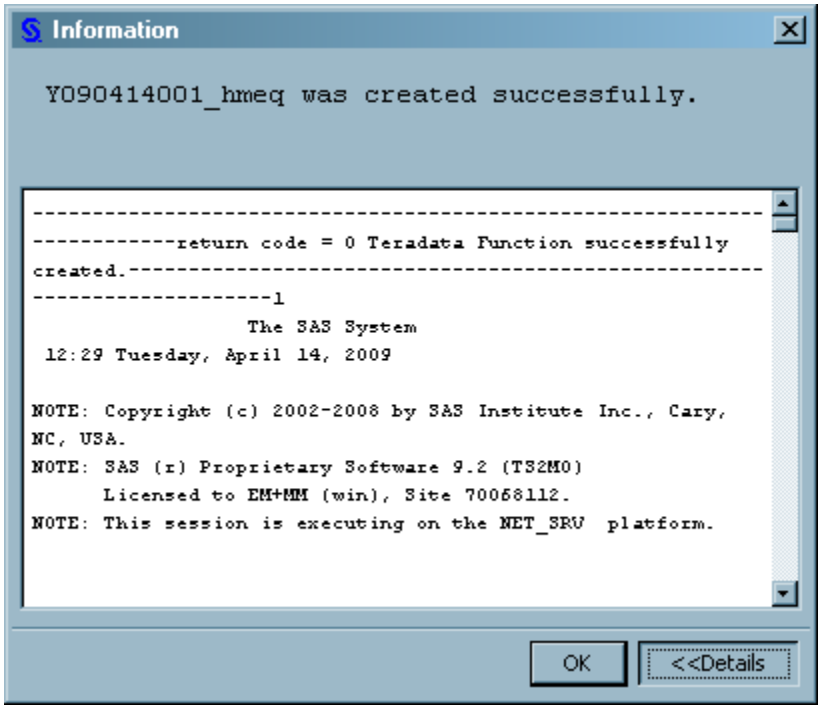
   Note: The user-defined value of the **Function Name** is validated against the target Teradata database. If the user-defined value is not unique across projects, an error message is displayed.



10. Click **OK**. A message indicating that the scoring function has been created successfully or unsuccessfully is displayed.

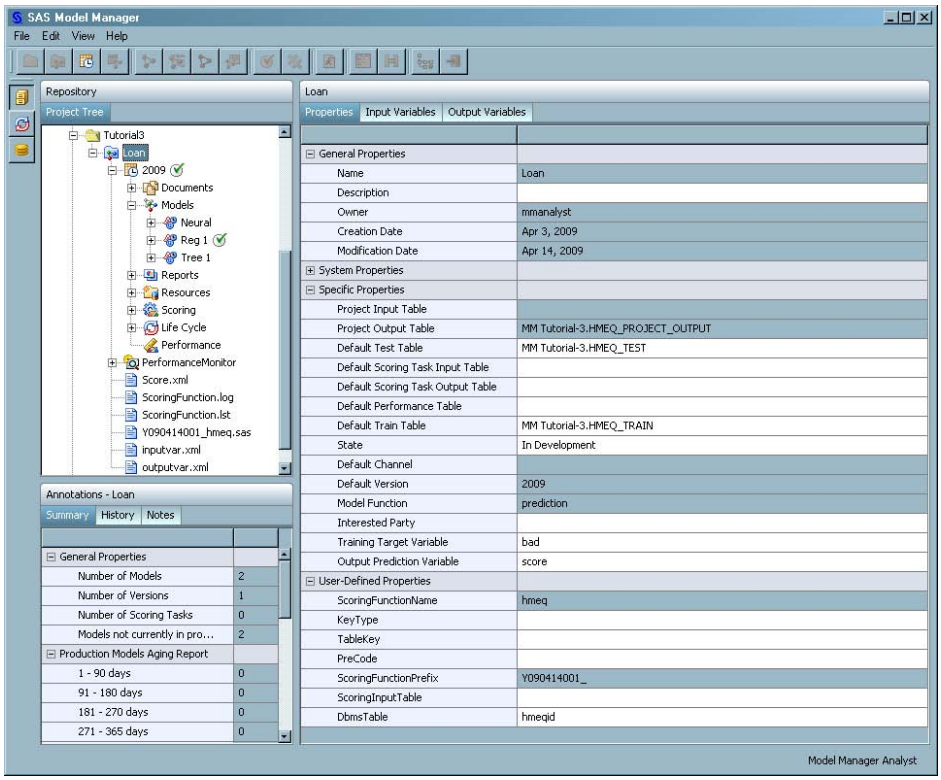


11. Click **Details** to display information about the publish scoring function actions or error messages.



12. Click **OK** to complete the publishing process. The SAS score code for the scoring function (for example, **`Y090414001_hmeq.sas`**) is added to the project file list and

the **ScoringFunctionName** and **ScoringFunctionPrefix** user-defined project properties are populated.



13. The actions that are performed during the publishing process are displayed in the history. To view the history of the project, select the project name and then click the **History** tab in the **Annotations - Loan** pane.

Here is an example of the history information:

```
Apr 14, 2009 11:35:20 AM[mmanalyst] updated MiningResult Loan(SBIP://METASERVER/
Shared Data/Model Manager/TDScoringFunc/Loan(MiningResults)) on
emm02.na.sas.com.
Apr 14, 2009 11:35:21 AM[mmanalyst] attaches Loan to SBIP://METASERVER/Shared
Data/Model Manager/TDScoringFunc successfully.
Apr 14, 2009 11:35:58 AM[mmanalyst] The content of the file ScoringFunction.lst
has been updated.
Apr 14, 2009 11:35:58 AM[mmanalyst] The content of the file ScoringFunction.log
has been updated.
Apr 14, 2009 11:36:02 AM[mmanalyst] attaches Loan(http://emm02.na.sas.com:8080/
SASContentServer/repository/default/ModelManager/MMRoot/Tutorial3/Loan) to
pubs(pubs) return code is return code = 0 Teradata Function successfully
created.
```

Note: After you complete the publishing process you can view a log of the actions that were performed during the scoring function publishing process in the **`ScoringFunction.log`** file. The file is located in the project folder. For more information about the contents of the scoring function log, see the *SAS Model Manager: User's Guide*.

Chapter 7

Tutorial 6: Using Advanced SAS Model Manager Features

Overview of Using Advanced Features

This tutorial is designed to enable a user who is already familiar with the administrative and the basic functions of SAS Model Manager to perform the following tasks:

- create a new model template using the SAS Model Manager Template Editor
- deploy the new model template to the middle-tier server
- import a model that is described by the template
- generate a report on the model

The tutorial contains examples and step-by-step directions for performing these tasks.

Example Scope

The model used in this tutorial is based on the ARBORETUM procedure, which is a SAS Enterprise Miner procedure.

Note: Outside its use in SAS Enterprise Miner, PROC ARBORETUM is an experimental procedure as of January 2007. SAS Technical Support does not support it until it becomes a production procedure.

Prepare Tutorial Data and Model Files

In this exercise you create a file system folder to store SAS data tables that the SAS Workspace Server accesses and SAS programs that SAS Model Manager accesses to import models.

The Required Tutorial Files

The SAS data sets and models that are required for this tutorial are on your local computer after you extract them from the ZIP file SMM22Tutorial.zip. If you have not extracted the tutorial files, see "The Tutorial Files" on page 3.

This tutorial requires the following files and folders in the **`<drive>\Tutorial6\Samples`** folder:

- HmeqProjectInput.sas7bdat
- HmeqProjectOutput.sas7bdat
- HmeqScoreInput.sas7bdat
- HmeqScoreOutput.sas7bdat
- HmeqTest.sas7bdat
- HmeqTrain.sas7bdat
- VarImportance.sas
- The **Model6** folder contains the following files:
  - importance6.sas7bdat
  - modelinput6.sas7bdat
  - modeloutput6.sas7bdat
  - nodestat6.sas7bdat
  - path6.sas7bdat
  - rules6.sas7bdat
  - score6.sas
  - target6.sas7bdat

Define a Data Library in SAS Management Console

Use SAS Management Console to define a data library and register the tables. Follow these steps:

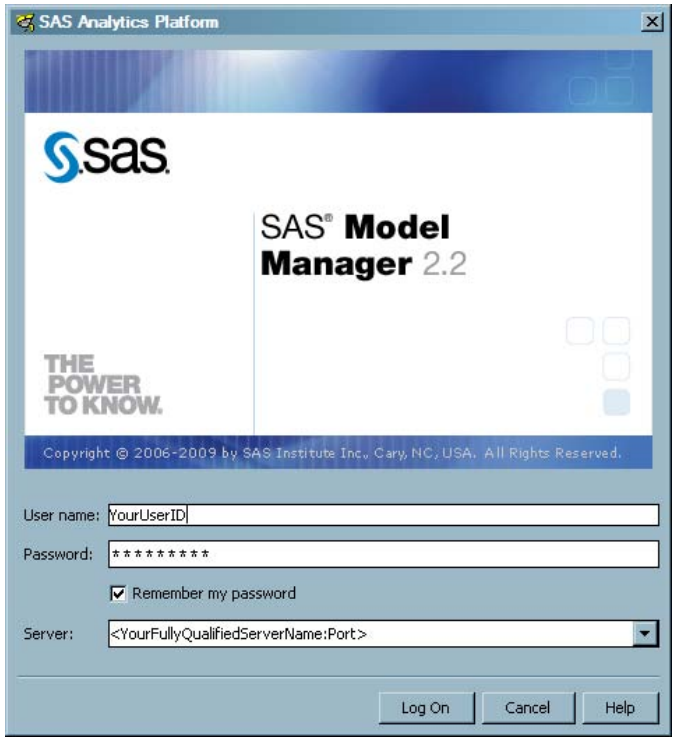
1. Open SAS Management Console and log on to the SAS Metadata Server. Your user ID must be authorized to modify libraries in the metadata server.
2. Open the New Library Wizard to define the data library. Click **Environment Management** ⇨ **Data Library Manager** on the **Plug-ins** tab. Right-click **Libraries** and select **New Library** from the pop-up menu.
3. In the New Library Wizard, create a SAS library.
   a. Select **SAS BASE Library** and click **Next**.
   b. Specify `MM Tutorial-6` in the **Name** field.
   c. Ensure that the **Location** box specifies **/SharedData/ModelManager** and click **Next**.
   d. (Optional) Depending on your configuration, if more than one server exists, then select a server in the **Available servers** list and click  to move the server name to the **Selected servers** list.
   e. Specify `smm2tor6` for the libref and click **New**.
   f. Specify the server folder that you previously created, ***drive*`\Tutorial6\Samples`**, for the path specification and click **OK** twice.
   g. Click **Next** and **Finish**.
4. Register the data tables in the metadata repository.
   a. Right-click **MM Tutorial-6** under the **Libraries** node, select **Register Tables** from the pop-up menu, and click **Next**.
   b. If prompted, specify the user ID and password for the server and click **OK**. Then click **Next**.
   c. Click **Select All Tables**, click **Next**, and click **Finish**.
5. Verify that table metadata was created and close SAS Management Console. Select **MM Tutorial-6** under the **Libraries** node and examine the right pane.

Add Data Sources

Start SAS Model Manager

To run SAS Model Manager, follow these steps:

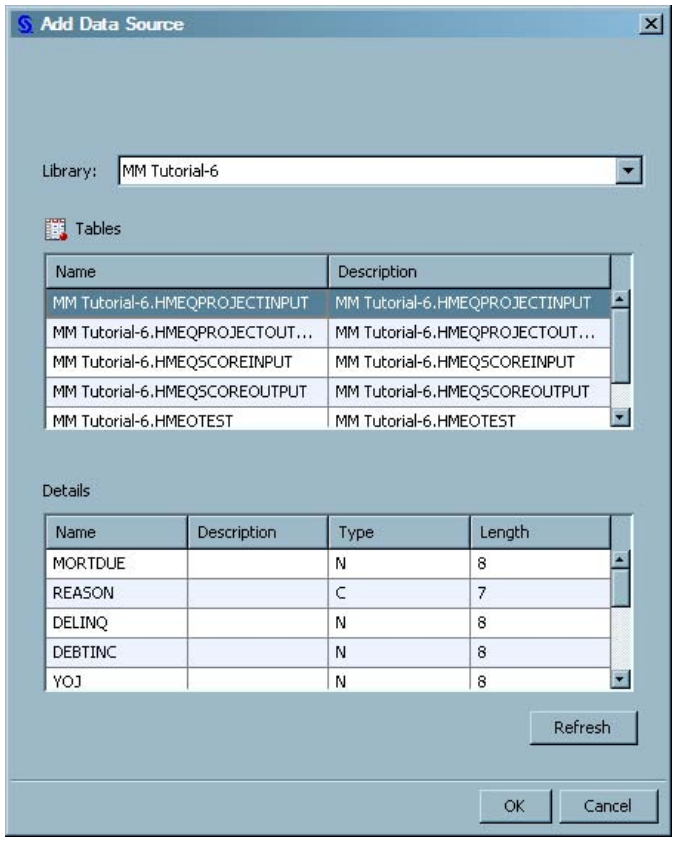
1. On your client machine, start the SAS Model Manager client.
2. Log on to SAS Model Manager. Ensure that your user ID is set up to use the tutorials.



Select Data Tables

To add data tables in the Data Source perspective that you use to define projects, create scoring tasks, and create reports, follow these steps:

1. Click the **Data Sources** perspective button.
2. Right-click **Project Input Tables** and select **Add Data Source**. The Add Data Source window opens.
3. In the **Library** box, select the library **MM Tutorial-6**.
4. In the **Tables** box, select **MM Tutorial-6.HMEQPROJECTINPUT**.



5. Click **OK**.

Repeat steps 2 through 5 to select the following table names and select the corresponding item to add the tables in the Data Sources perspective:

Folder name	Data Table name
Project Output Tables	**MM Tutorial-6.HMEQPROJECTOUTPUT**
Test Tables	**MM Tutorial-6.HMEQTEST**
Scoring Task Input Tables	**MM Tutorial-6.HMEQSCOREINPUT**
Scoring Task Output Tables	**MM Tutorial-6.HMEQSCOREOUTPUT**
Training Tables	**MM Tutorial-6.HMEQTRAIN**

Organize the Model Hierarchy

In this exercise, you use the Project Tree to create a modeling project.

Create a Folder

To provide an organizational folder to manage your modeling projects, follow these steps:

1. Click the Projects perspective button  .
2. Right-click the **MMRoot** node in the Project Tree and select **New** ⇨ **New Folder**. The New Folder dialog box opens.
3. Specify the following folder properties and click **OK**.

   Name
   enter `Tutorial6`.

   Description
   enter an optional folder description.

Create a New Project

To create a project and define the model function, follow these steps:

1. Right-click **Tutorial6** and select ⇨ **New** ⇨ **New Project**. The New Project dialog box opens.
2. Specify the following project properties and click **OK**:

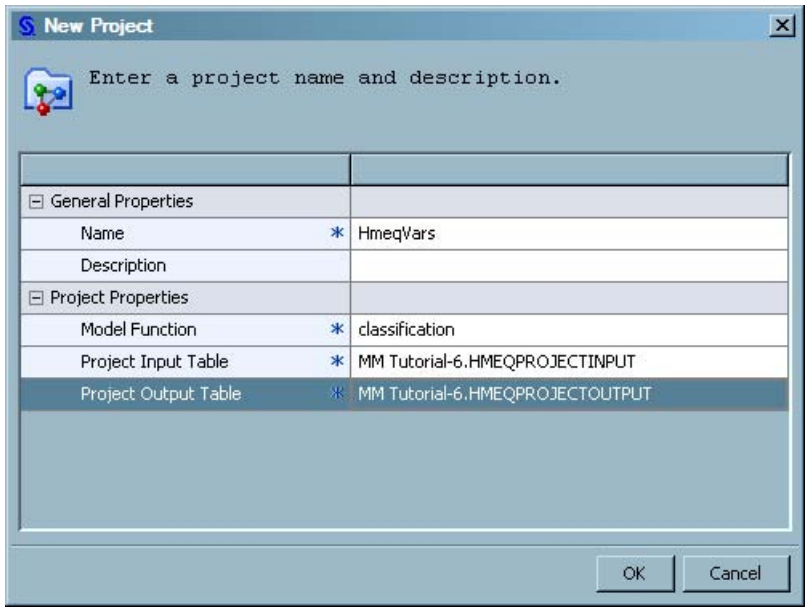
   Name
   enter `HmeqVars`.

   Description
   enter an optional description.

   Model Function
   select **classification**.

   Project Input Table
   select **MM Tutorial-6.HMEQPROJECTINPUT**

   Project Output Table
   select **MM Tutorial-6.HMEQPROJECTOUTPUT**



Define the Project Properties

To define the properties that SAS Model Manager uses to create reports and model scores, follow these steps:

1. Select the **HmeqVars** project in the **Tutorial6** folder and expand **Specific Properties** in the right pane.
2. Specify the default data tables and model variables for the project:.

   Default Test Table
   select **MM Tutorial-6.HMEQTEST**.

   Default Scoring Task Input Table
   select **MM Tutorial-6.HMEQSCOREINPUT**.

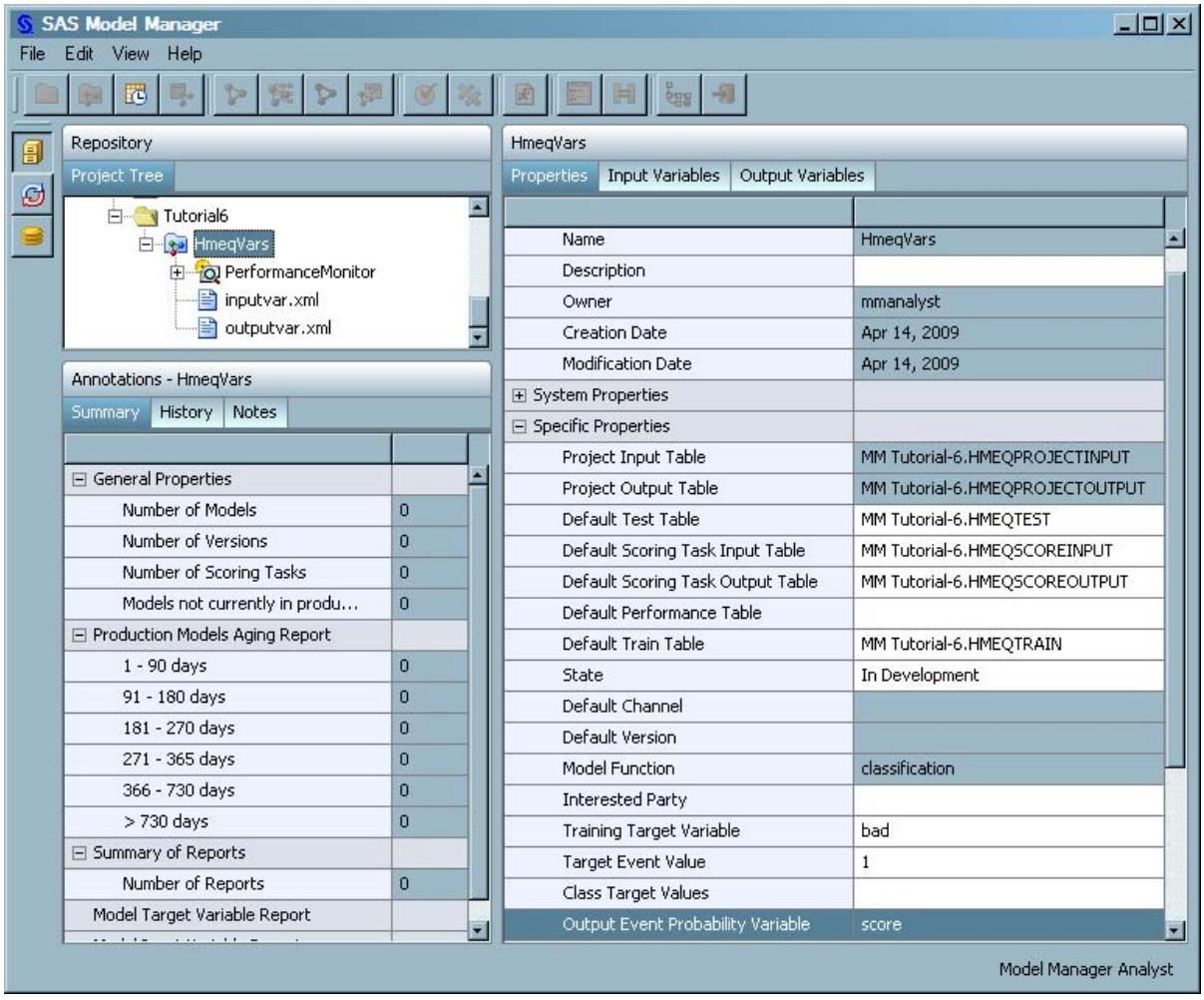
   Default Scoring Task Output Table
   select **MM Tutorial-6.HMEQSCOREOUTPUT**.

   Default Train Table
   select **MM Tutorial-6.HMEQTRAIN**.

   Training Target Variable
   enter `bad`.

   Training Event Variable
   enter `1`.

   Output Event Probability Variable
   select **score**.



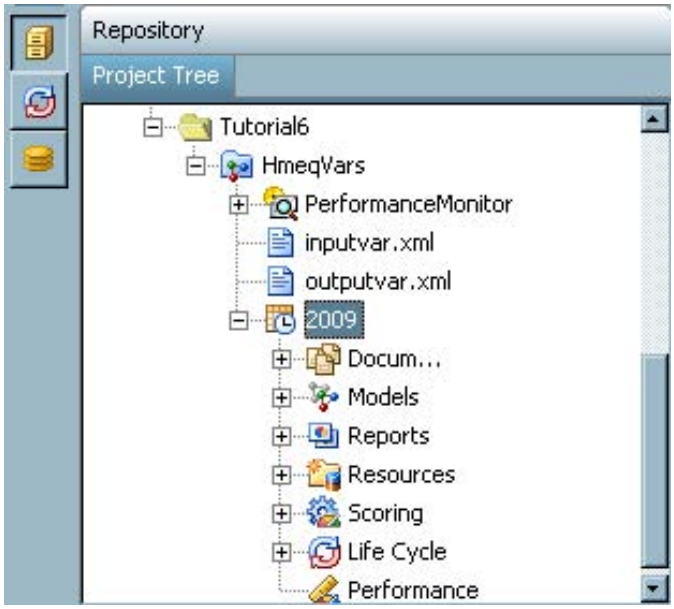
Create a Version

Create the version for the project. The version folder contains life cycle information, auxiliary version documents, candidate model files, model comparison reports, resource files, scoring tasks, and model performance reports.

1. Right-click the **HmeqVars** project and select **New** ⇨ **New Version**. The New Version dialog box opens.
2. Specify the following version properties and click **OK**.

   Name
   enter `2009`.

   Life Cycle Template
   select the user-defined template **Tutorial Life Cycle** that you created in the first tutorial. For more information, see "Create a Life Cycle Template" on page 9. If **Tutorial Life Cycle** is not a selection in the list, select any life cycle template.
3. Examine the **HmeqVars** project to verify that it contains one version, **2009**.



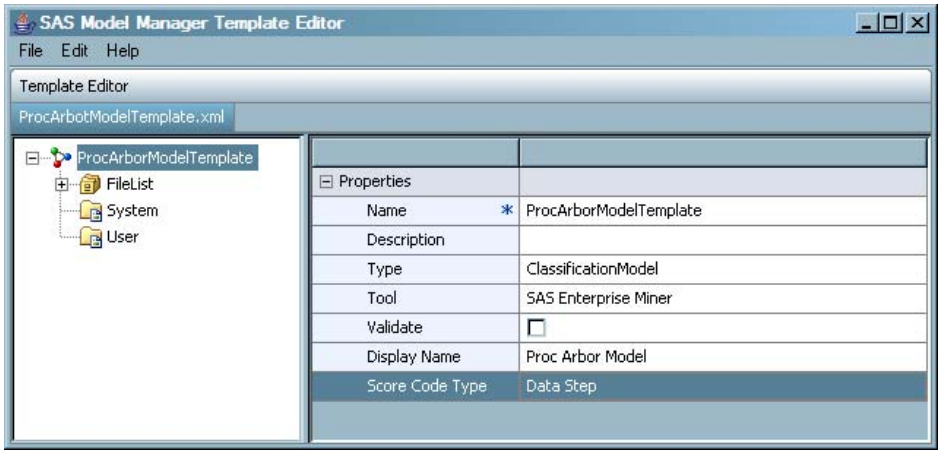
Create a Model Template

In this exercise, you create a new model template using the SAS Model Manager Template Editor. For information about creating a model template, see the *SAS Model Manager: User's Guide*.

To create a model template, follow these steps:

1. Open the SAS Model Manager Template Editor. Select **File** ⇨ **Manage Templates**.
2. Open an empty model template in the template editor. Select **File** ⇨ **New Model Template**. An empty, untitled model template opens.
3. Specify the model template properties.
   a. In the **Name** field, replace **Untitled Template** with **`ProcArborModelTemplate`**.
   b. Click the **Type** box and select **ClassificationModel**.
   c. In the **Tool** field, enter **`SAS Enterprise Miner`**.
   d. In the **Display Name** field, enter **`Proc Arbor Model`**.
   e. Click the **Score Code Type** box and select **Data Step**.

   Here is the template editor after this step has been completed:



4. Add the model component files and the model component file properties.

   The following table lists the model component files that comprise the model, and the properties for each file. For each model component file, add an entry under **FileList**. Then, select the file under **FileList** and enter the properties for that file. A value of n/a indicates that you do not need to set a value for that property.

   To add the model component files, right-click **FileList** and select **New File Item**. In the **Name** field, enter the name from the table and click **OK**.

   Here is the template after adding the file score.code as a model component file:

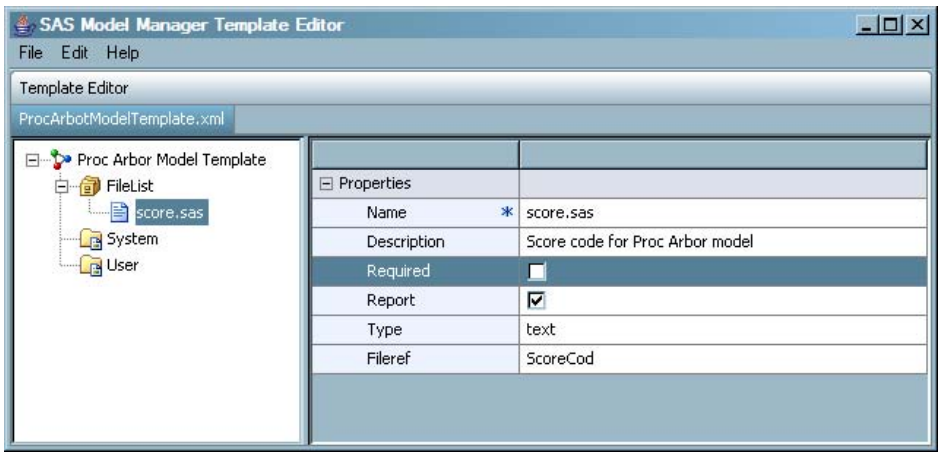


Table 7.1 *ProcArborModelTemplate Component Files and Component File Properties*

Filename	Description	Required	Report	Type	Fileref
score.sas	Score code for Proc Arbor model	n/a	select the box	text	ScoreCod
modelinput.sas7bdat	Model input table	n/a	select the box	binary	n/a
modeloutput.sas7bdat	Model output table	n/a	select the box	binary	n/a
inputvar.xml	Input variable XML file	n/a	select the box	text	Input
outputvar.xml	Output variable XML file	n/a	select the box	text	Output

Filename	Description	Required	Report	Type	Fileref
target.sas7bdat	Target variable table	select the box	select the box	binary	Target
importance.sas7bdat	Variable relative importance	n/a	select the box	binary	n/a
path.sas7bdat	Path information	n/a	select the box	binary	n/a
rules.sas7bdat	Node rules	n/a	select the box	binary	n/a
nodestat.sas7bdat	Node statistics	n/a	select the box	binary	n/a

5. Add a system property.

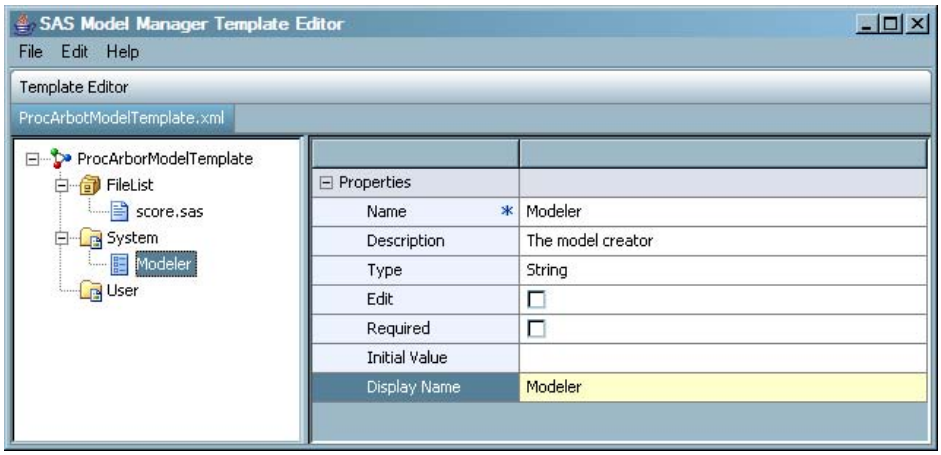
   a. In the left pane, right-click **System** and select **New Property**. In the **Name** field, enter `Modeler` and click **OK**.

   b. Select **Modeler** and enter the following property values:

   Description
   The model creator.

   Display Name
   Modeler

   Default values are used for all other property values. Here is the template after the system property has been added:



6. Add user properties.

   The table below contains user properties for the model template and the properties of each user property. For each user property, add an entry under **User**. Then, select the property under **User** and enter the properties for the user property. A value of n/a indicates that you do not need to set a value for that property.

   To add a user property, right-click **User** and select **New Property**. In the **Name** field, enter the name from the table and click **OK**.

   Here is the template after the user property **Citi1** has been added:

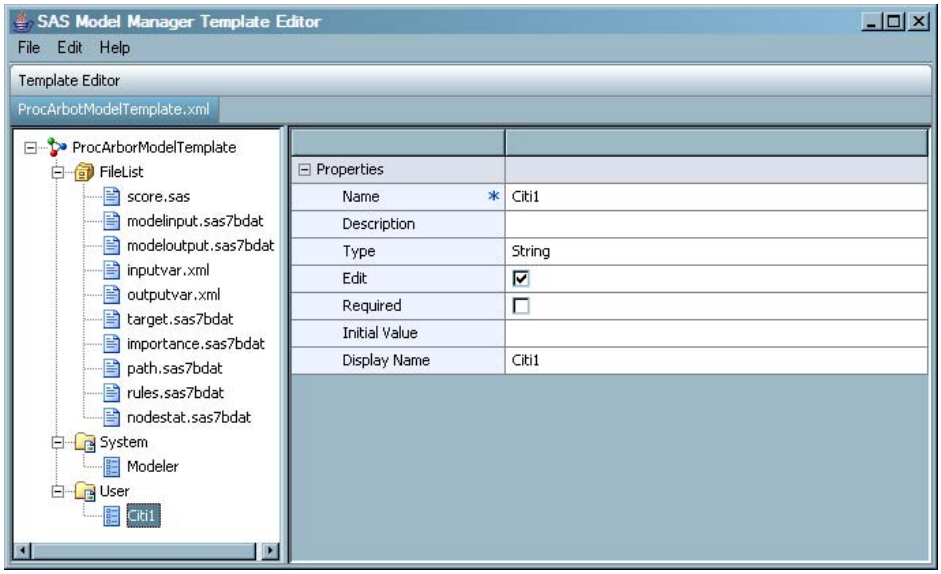
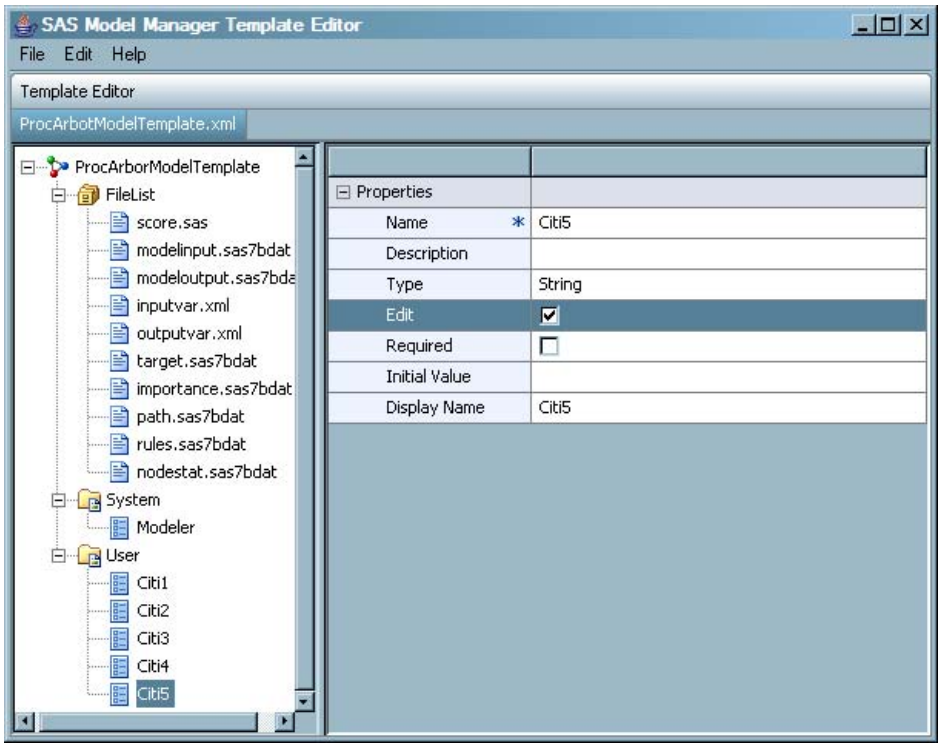


Table 7.2 *ProcArborModelTemplate User Properties and User Property Properties*

Filename	Description	Type	Edit	Required	Initial Value	Display Name
Citi1	n/a	String	select the box	n/a	n/a	Citi1
Citi2	n/a	String	select the box	n/a	n/a	Citi2
Citi3	n/a	String	select the box	n/a	n/a	Citi3
Citi4	n/a	String	select the box	n/a	n/a	Citi4
Citi5	n/a	String	select the box	n/a	n/a	Citi5

7. Save the template. Select **File** ⇨ **Save As** and enter `ProcArborModelTemplate.xml` in the **File name** field. Click **OK**.

   Here is the model template after all files and properties have been added to the template:



8. Select **File** ⇨ **Exit** to close the SAS Model Manager Template Editor.
9. Open ProcArborModelTemplate.xml in a text editor and modify the **`<FileList libref="">`** element so that it reads **`<FileList libref="SMMmodel">`**.
10. Save the file and exit the text editor.

Install the New Model Template

In this exercise, you copy the new model template to the middle-tier server where SAS Model Manager is installed. Only users who have Write access to the SAS Model Manager user template directory on the middle-tier server can copy the template to the server. The SAS Analytics Platform server must then be restarted.

1. From the folder where you saved the model template, copy the ProcArborModelTemplate.xml file to this directory on the middle-tier server:

   `\`*`sasconfigdir`*`\Lev#\AnalyticsPlatform\apps\ModelManager\ext`

   sasconfigdir is the directory where SAS is installed.

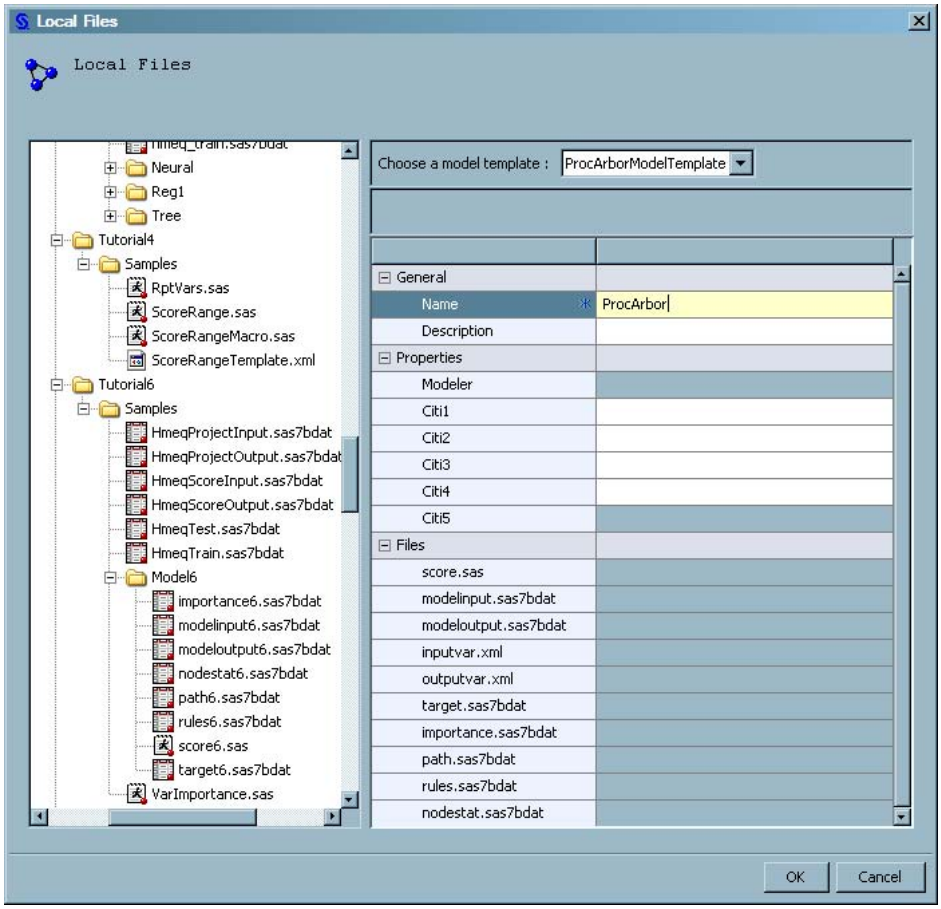
   Lev# is a level number, where # indicates a number. For example, Lev1.
2. Exit from SAS Model Manager.
3. Contact a system administrator to shut down and restart the SAS Analytics Platform server in order to activate the new model template. The model template now appears as a selection when you import a model using the Local Files dialog box.

Import a Model

Import a Model

In this exercise, you import a model using the user model template:

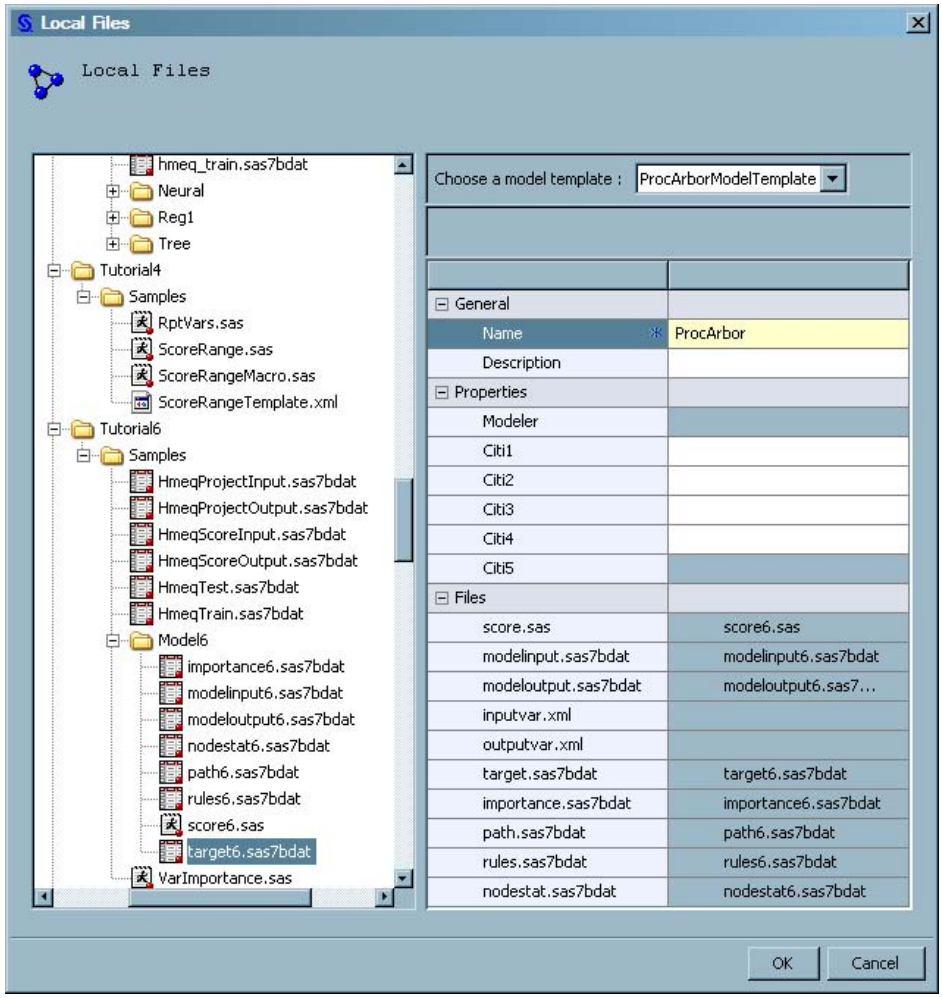
1. Log on to SAS Model Manager.
2. In the **2009** version of the **HmeqVars** project, right-click **Models** and select **Import From ⇨ Local Files**.
3. From the **Choose a model template** box, select **ProcArborModelTemplate**. The custom properties and files appear in the right pane.
4. In the Name field, enter **ProcArbor**.
5. In the left pane, expand **Desktop** to **`<drive>:\Tutorial6\Samples\Model6`**. The Local Files dialog box displays the template in the right pane and the files to import in the left pane:



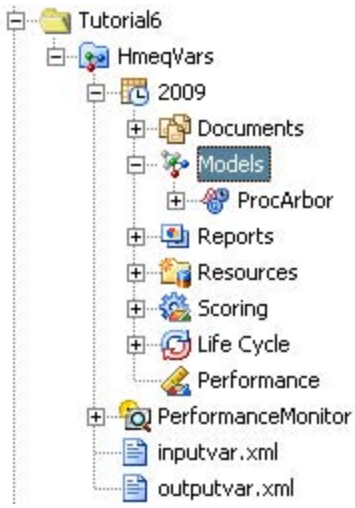
6. From the left pane, drag the following files to the corresponding fields in the model template:

Filename	Template Field Name
importance6.sas7bdat	**importance.sas7bdat**
modelinput6.sas7bdat	**modelinput.sas7bdat**
modeloutput6.sas7bdat	**modeloutput.sas7bdat**
nodestat6.sas7bdat	**nodestat.sas7bdat**
path6.sas7bdat	**path.sas7bdat**
rules6.sas7bdat	**rules.sas7bdat**
score6.sas	**score.sas**
target6.sas7bdat	**target.sas7bdat**

Here is the Local Files dialog box after the files have been assigned:



7. Click **OK**. The **ProcArbor** model appears under the **Models** folder.

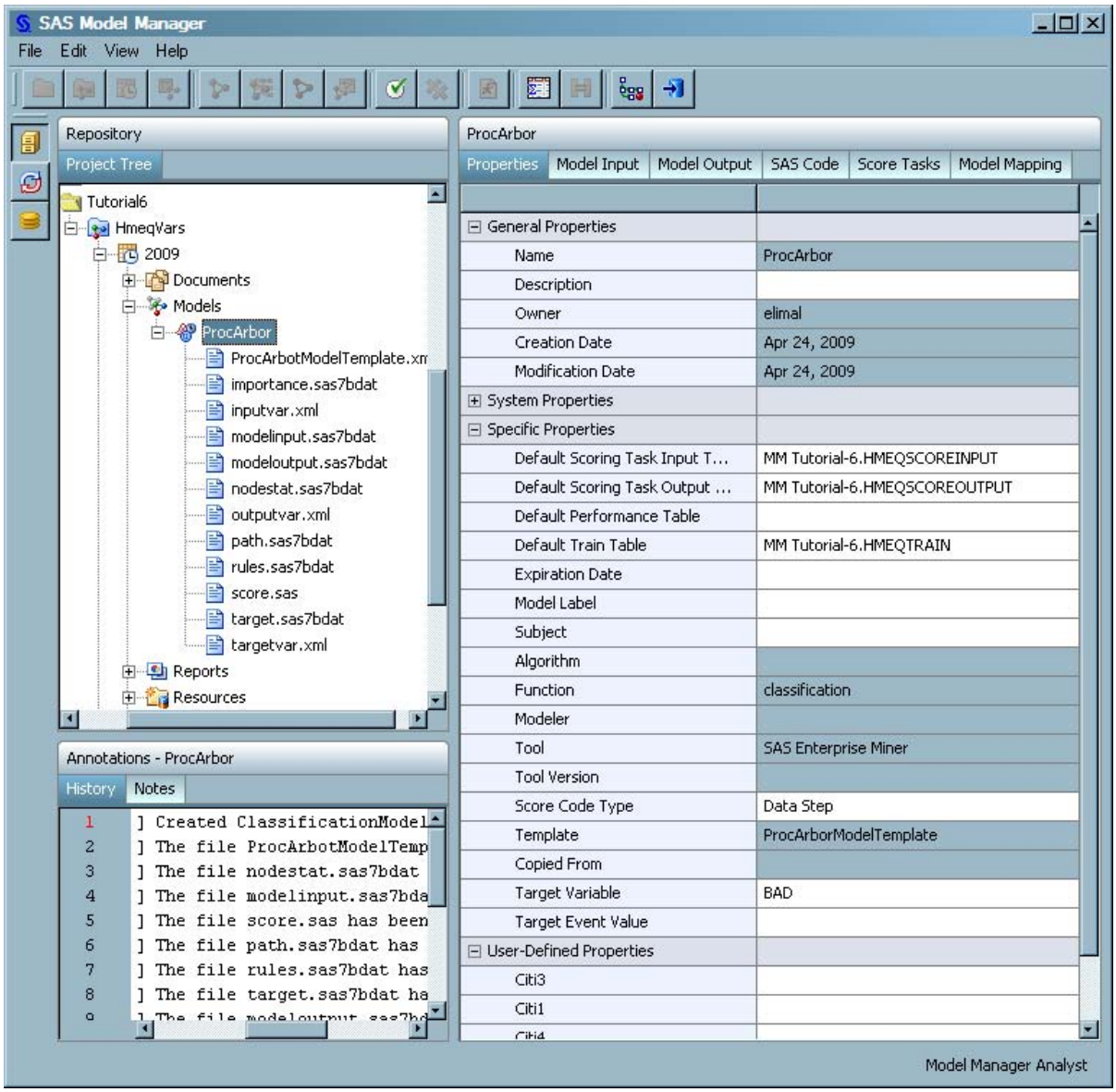


Verify Model Properties

Verify some model properties:

1. In the Project Tree, expand the **Models** folder and select **ProcArbor**.
2. In the Properties View, verify **BAD** as the value of the **Target Variable** property.
3. Verify that the **Score Code Type** property is **Data Step**.

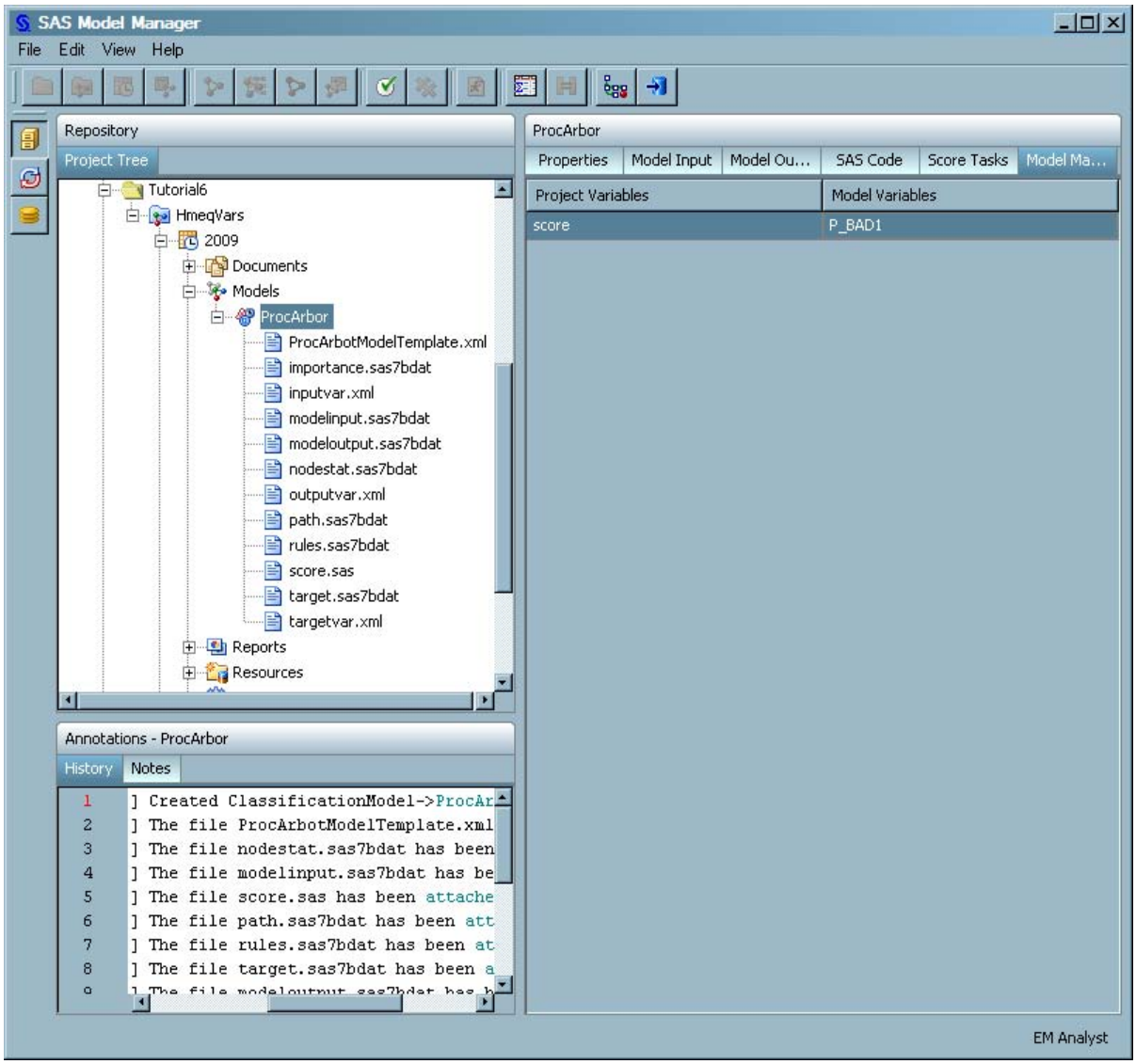
Here is the ProcArbor model in the SAS Model Manager window:



Map Model Output Variables to Project Output Variables

Because the project output variable name is not the same as the model output variable name, the output variables must be mapped. To map the variables, follow these steps:

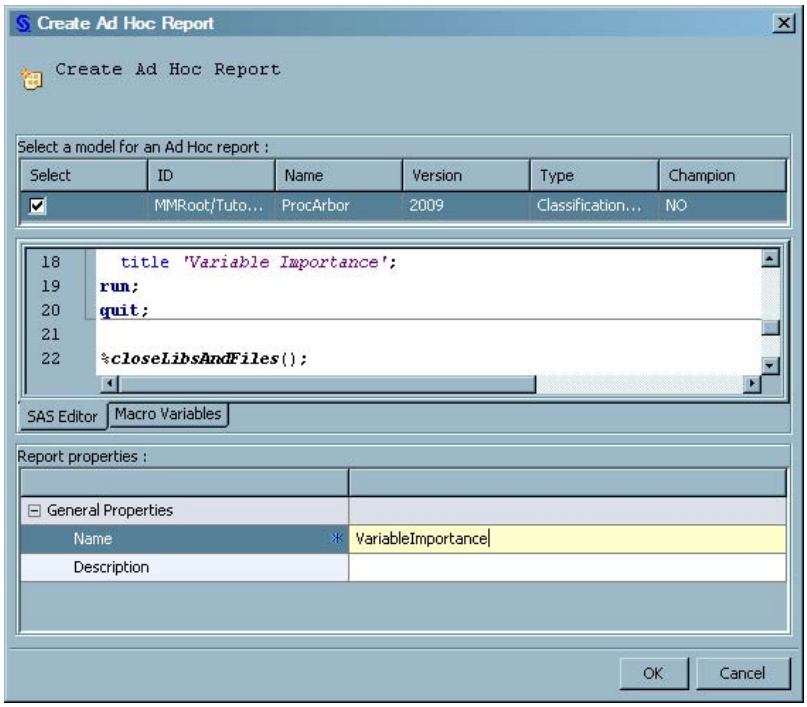
1. Select the model **ProcArbor**.
2. Click the **Model Mapping** tab to open the Model Mapping View.
3. In the **Model Variables** column for score, select **P_BAD1**.



Create an Ad-Hoc Variable Importance Report

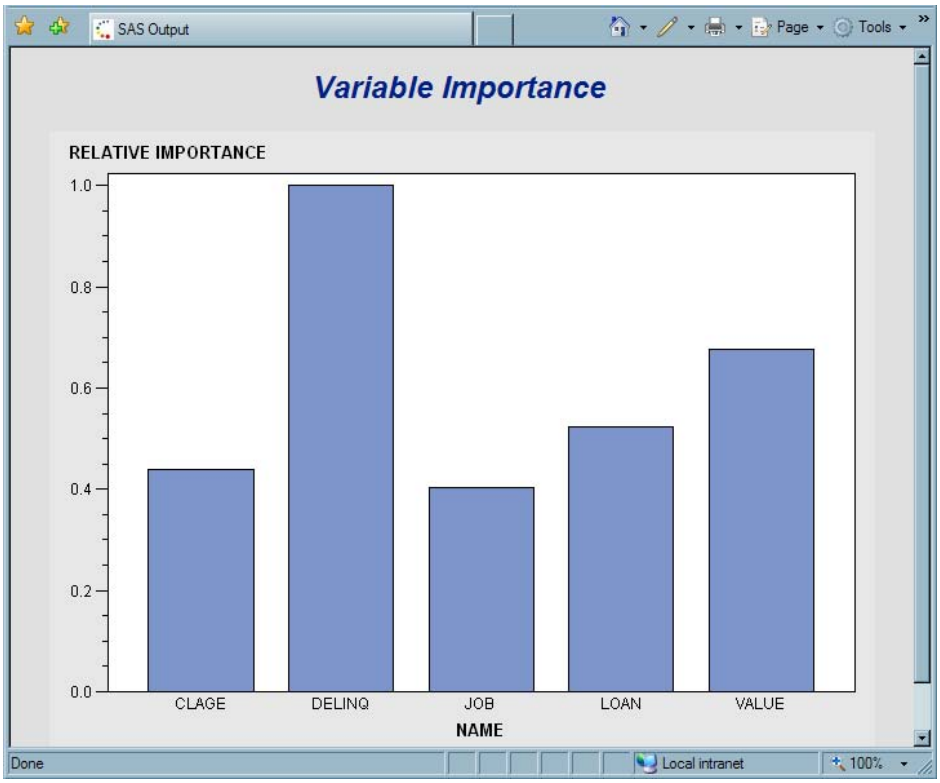
In this exercise, you create a report that is based on the model's PROC ARBORETUM importance data.

1. In the **`<drive>:Tutorial6/Samples`** folder, open the example report VarImportance.sas in a text editor and copy the code.
2. In the Project Tree, expand the **Tutorial6** folder and the project **HmeqVars**.
3. Under version **2009**, right-click the **Reports** folder and select **Reports ⇨ Create Ad Hoc Report**.
4. In the Create Ad Hoc Report dialog box, check the box for **ProcArbor** in the **Select a model for an Ad Hoc report** table.
5. In the SAS Editor, paste the code that you copied in Step 1.
6. In the **Name** field, enter **`VariableImportance`** and click **OK**.



7. View the report output.
   a. Expand the new report, **VariableImportance**.
   b. Right-click **ProcArbor.html** and select **Open Item**.
   c. If prompted, enter your user ID and password.

Here is the VariableImportance report:



Chapter 8

Tutorial 7: Creating Performance Monitoring Reports

Overview of Monitoring Reports

SAS Model Manager model monitoring reports enable you to monitor and evaluate model performance. Degrading model performance might be improved by tuning or refitting the model, or by using a new champion model.

To create monitoring reports, you create a performance task by using the Define Performance Task wizard. Then, you execute the performance task. The output from executing a performance task includes several charts, including Characteristic, Stability, Lift, ROC & Gini, and Kolmogorov-Smirnov charts. You can view these charts in SAS Model Manager or you can create monitoring reports in PDF, HTML, RTF, or Excel output formats.

Prepare the Tutorial Data Sets and Models

In this exercise you create a library in SAS Management Console for the data tables that are used in this tutorial.

Models Used in Tutorial 3

The exercises in this tutorial depend on some of the properties of the specific models that were added in Chapter 4, "Tutorial 3: Importing and Exporting Models," on page 55.

The Required Tutorial Files

The SAS data sets that are required for this tutorial are on your local computer after you extract them from the ZIP file SMM22Tutorial.zip. If you have not extracted the tutorial files, see "The Tutorial Files" on page 3.

This tutorial requires the following files and folders in the **<*drive*>\Tutorial7\Samples** folder:

- hmeq_2009Q1.sas7bdat
- hmeq_2008Q4.sas7bdat
- hmeq_2008Q3.sas7bdat
- hmeq_2008Q2.sas7bdat

Define a Data Library in SAS Management Console

Use SAS Management Console to define a data library and register the tables. Follow these steps:

1. Open SAS Management Console and log on to the SAS Metadata Server. Your user ID must be authorized to modify libraries in the metadata server.
2. Open the New Library Wizard to define the data library. Click **Environment Management** ⇨ **Data Library Manager** on the **Plug-ins** tab. Right-click **Libraries** and select **New Library** from the pop-up menu.
3. In the New Library Wizard, create a SAS library.
   a. Select **SAS BASE Library** and click **Next**.
   b. Specify **MM Tutorial-7** in the **Name** box.
   c. Ensure that the **Location** box specifies **/SharedData/ModelManager**, and click **Next**.
   d. (Optional) Depending on your configuration, if more than one server exists, then select a server.
   e. Specify **smm2tor7** for the libref and click **New**.
   f. Specify the server folder that you previously created, **<*drive*>\Tutorial7\Samples**, for the path specification and click **OK** twice.
   g. Click **Next** and **Finish**.

4. Register the data tables in the metadata repository.
   a. Right-click **MM Tutorial-7** under the **Libraries** node, select **Register Tables** from the pop-up menu, and click **Next**.
   b. If prompted, specify the user ID and password to the metadata server and click **OK**. Then click **Next**.
   c. Click **Select All Tables** and click **Browse**. The Select a Location dialog box appears.
   d. From the **Look in** box, select **Model Manager**. Click the **New folder** icon and enter `Tutorial7`. Double-click **Tutorial7**. Click **OK**.
   e. Click **Next**, and then click **Finish**.
5. Verify that table metadata was created. Select **MM Tutorial-7** under the **Libraries** node and examine the right pane.

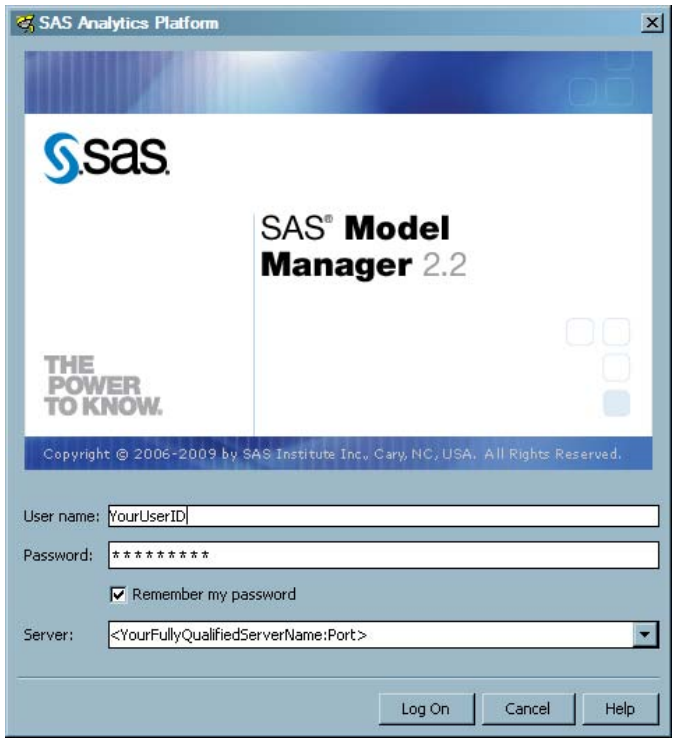
Add Input Data Sources

In this exercise you use SAS Model Manager to access the data tables that the performance task uses.

Start SAS Model Manager

To run SAS Model Manager, follow these steps:

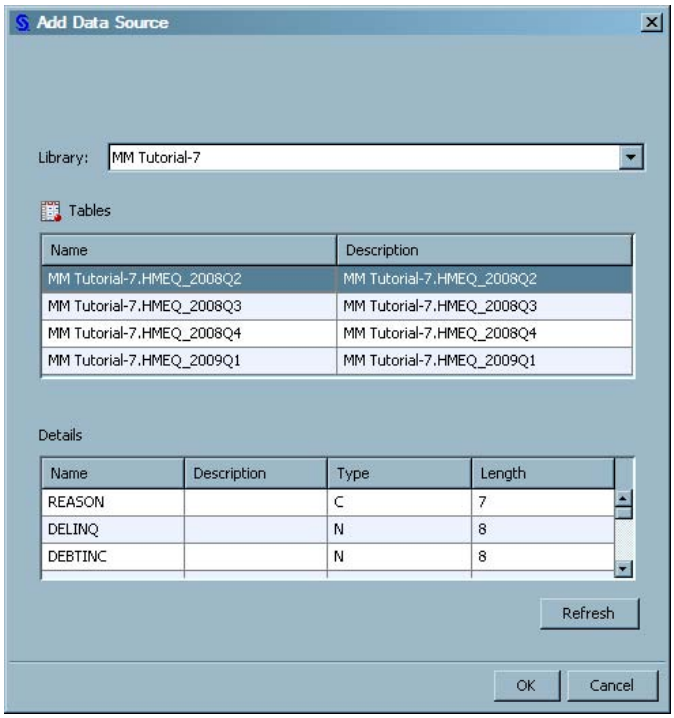
1. On your client machine, start the SAS Model Manager client.
2. Log on to SAS Model Manager. Ensure that your user ID is set up to use the tutorials.



Select Data Sources

To add data tables in the Data Source perspective that you use to create performance monitoring reports, follow these steps:

1. Log on to SAS Model Manager. Ensure that your user ID is a member of the **Model Manager Advanced Users** group.
2. Click the **Data Sources** perspective button  .
3. Right-click **Performance Tables** and select **Add Data Source**. The Add Data Source window opens.
4. Click the **Library** box and select **MM Tutorial-7**.
5. In the **Tables** box, select **MM Tutorial-7.HMEQ_2008Q2** and click **OK**.



6. Repeat the above steps to add the remaining tables as SAS Model Manager data sources and click **OK**:
   - **MM Tutorial-7.HMEQ_2008Q3**
   - **MM Tutorial-7.HMEQ_2008Q4**
   - **MM Tutorial-7.HMEQ_2009Q1**

Create the Champion Model Performance Data Sets

In this exercise, you run the Define Performance Task wizard to create a performance monitoring task for the champion model, Reg 1. The performance monitoring task uses the information that you supply in the Define Performance Task wizard to create SAS

programs. You then execute the SAS programs to create the performance monitoring data sets.

Ensure the Project Properties Are Set

The Define Performance Task wizard requires that three project properties be set before you can run the wizard. Click the **Loan** project. Ensure that the following project properties are set:

Project Property	Value
Training Target Variable	`bad`
Target Event Variable	`1`
Output Event Probability Variable	**score**

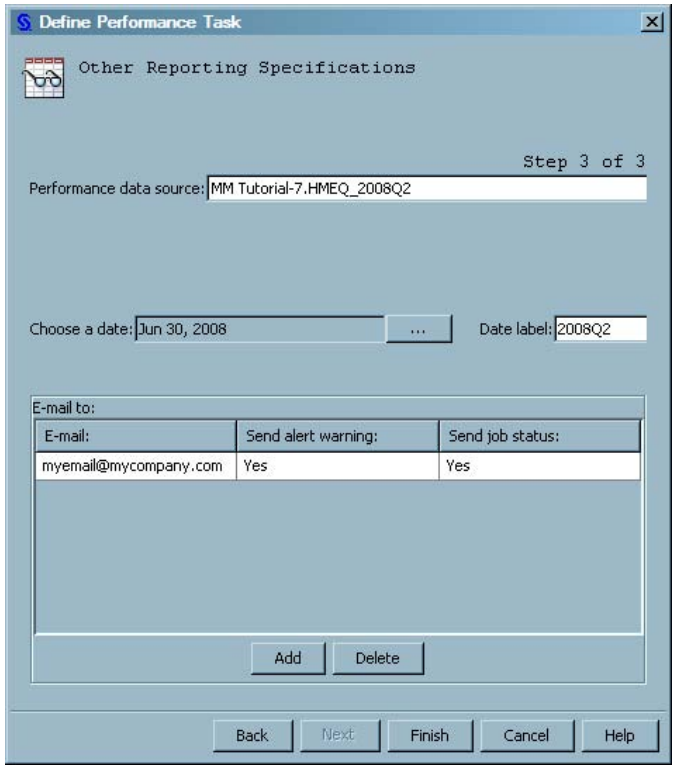
Run the Define Performance Task Wizard

To run the Define Performance Task wizard, follow these steps:

1. Expand the **Tutorial3** organizational folder, right-click **Loan**, and select **Define Performance Task** from the pop-up menu. The Define Performance Task wizard appears.
2. In the **Select Output Variable(s)** table, select the box for the **score** variable.
3. In the **Select Input Variable(s)** table, click **Select All**. Click **Next**.
4. On the **Warning and Alert Conditions** page, accept the default alert and warning conditions by clicking **Next**. The **Other Reporting Specifications** page appears.
5. Click the **Performance data source** box and select **MM Tutorial-7.HMEQ_2008Q2**.
6. Click the ellipsis button for **Choose a date** and select June 30, 2008 using the calendar. Click **OK**.

   Note: The date can be any date within the second quarter of 2008.
7. In the **Date label** box, enter `2008Q2`.

   Note: The label is used to identify the performance data in the performance monitoring charts. When you view the charts by using the Performance node in the Project Tree, SAS Model Manager uses the label **baseline** for the first set of performance data that is created for a champion model, and not the text that you entered in the **Date label** box.
8. Click **Add**. The Add Contact dialog box appears. Type your e-mail address, and click **OK**. Here is the Define Performance Task wizard.



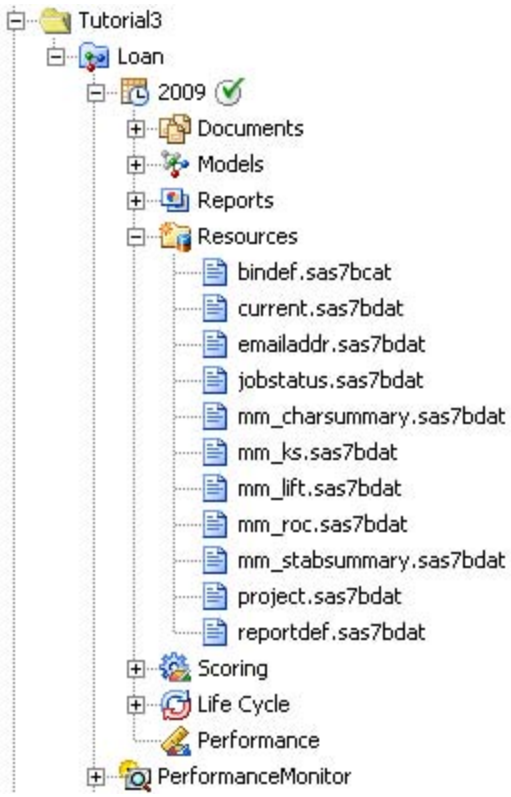
9. Click **Finish**. The wizard creates the SAS code that can be run to create the performance monitoring data sets. When the wizard has completed the task, an information message indicates whether the SAS code was saved successfully.

10. Execute the SAS program. Under the **Loan** project, right-click **PerformanceMonitor** and select **Execute**. SAS Model Manager executes the performance monitoring program. When the program execution is complete, an information message indicates whether the program ran successfully.

11. Expand **PerformanceMonitor**. Here you can see the SAS program that created the performance monitoring data sets and the resulting SAS log. Click both files to see the file contents in the **Content** pane.

12. Select **Resources** under the default version **2009**. The **Resources** node contains the data sets that are created by the Define Performance Task wizard as well as the data sets that are created by executing the performance task. When a performance task is executed the first time for a given champion model, the performance task creates the initial data sets that are used for plotting the model performance charts. In executing subsequent performance tasks that use new performance data for the given champion model, SAS Model Manager appends the resulting data sets to the existing data sets. All of the data in the model performance data sets for a given champion model is used to plot the model performance charts.

    Click on any file to see the contents of that file in the **Content** pane.



13. Select **Performance**. The Performance node displays the champion model performance data as a graph and as a data set.

    Note: To view at least one line segment in Characteristic and Stability graphs, SAS Model Manager requires performance data sets from three performance task executions, at a minimum.

14. Define a performance task and execute the SAS program for the remaining three Tutorial 7 performance data sources. Complete steps 1 through 10 for each performance data source.

    On the Define Performance Task wizard, page 1, select all input and output variables if they are not already selected.

    On page 2, use the default alert and warning conditions. No changes are necessary.

    On page 3, use these values for the **Performance data source**, **Choose a date**, and **Date label** boxes:

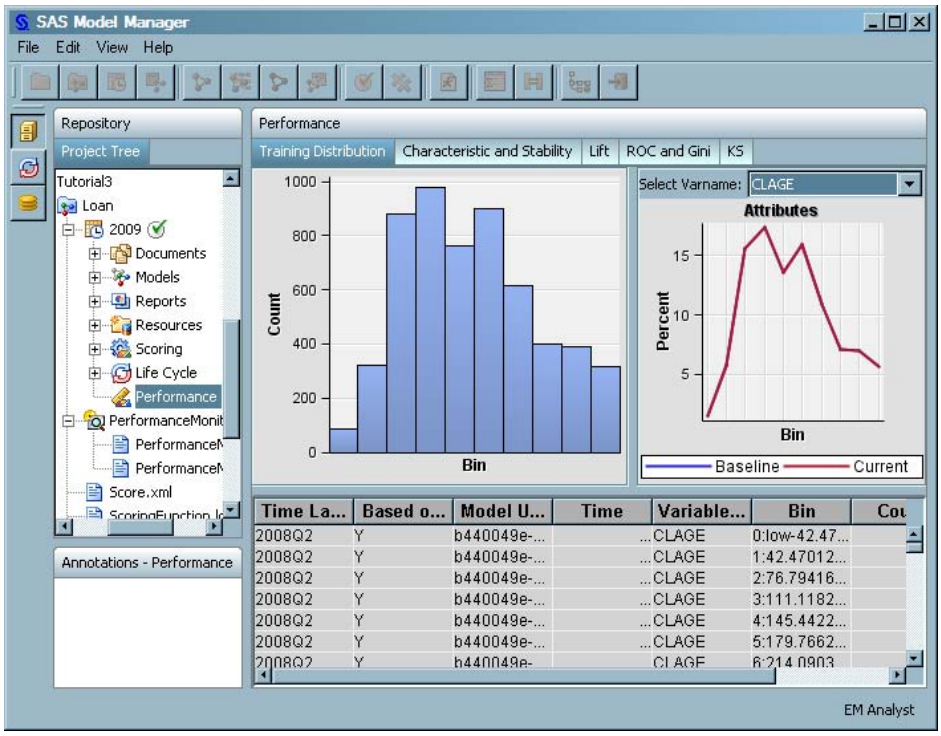
Performance data source	Choose a date	Date label
MM Tutorial-7.HMEQ2008Q3	September 30, 2008	2008Q3
MM Tutorial-7.HMEQ2008Q4	December 31, 2008	2008Q4
MM Tutorial-7.HMEQ2009Q1	March 31, 2009	2009Q1

View Performance Charts

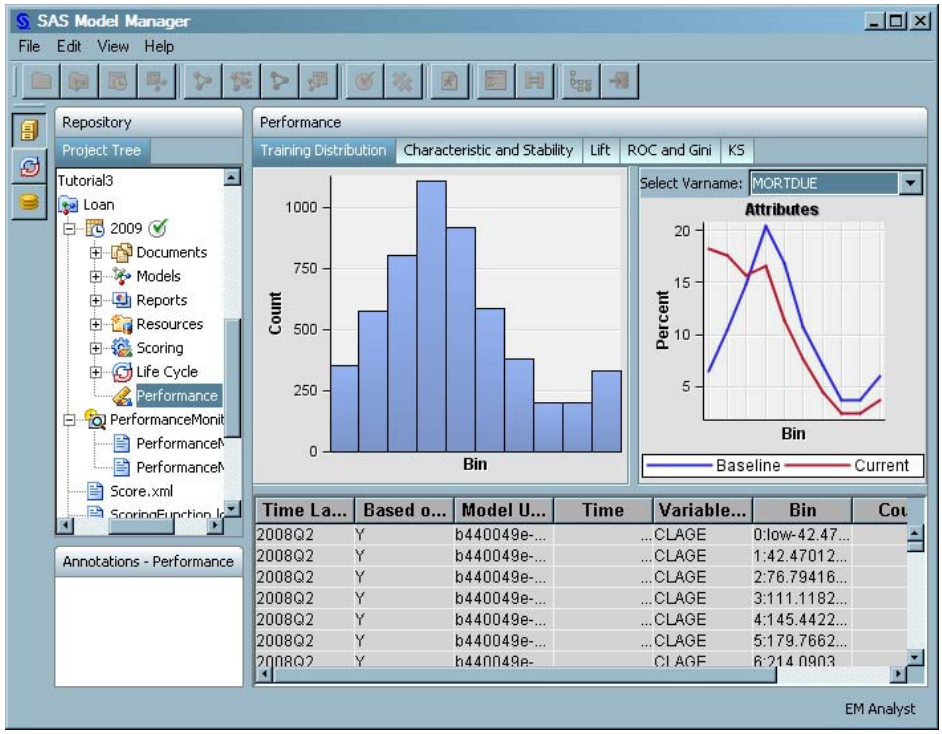
View the Training Distribution Chart

To demonstrate the Training Distribution chart features:

1. In the SAS Model Manager Project Tree, expand **Tutorial3**, expand **Loan**, and expand **2009**.
2. Select the **Performance** node to display the Performance charts.

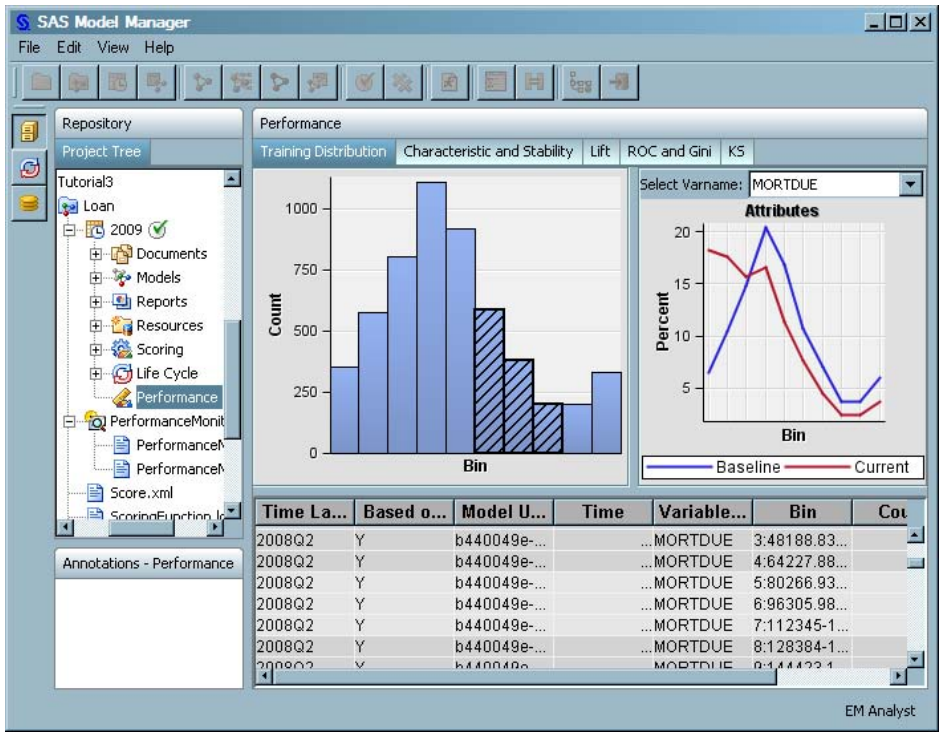


3. On the **Training Distribution** tab, click the **Select Varname** box and select **MORTDUE**. The training distribution data and charts display the data for the MORTDUE variable.



4. Click on a bar to highlight the corresponding entry in the table. If necessary, move the scroll bar to locate the highlighted entry in the table.

5. In the table, click the highlighted row and drag the mouse to select multiple rows. The associated bars in the chart are highlighted as you select the associated rows.



View the Characteristic and Stability Charts

To demonstrate the Characteristic and Stability chart features:

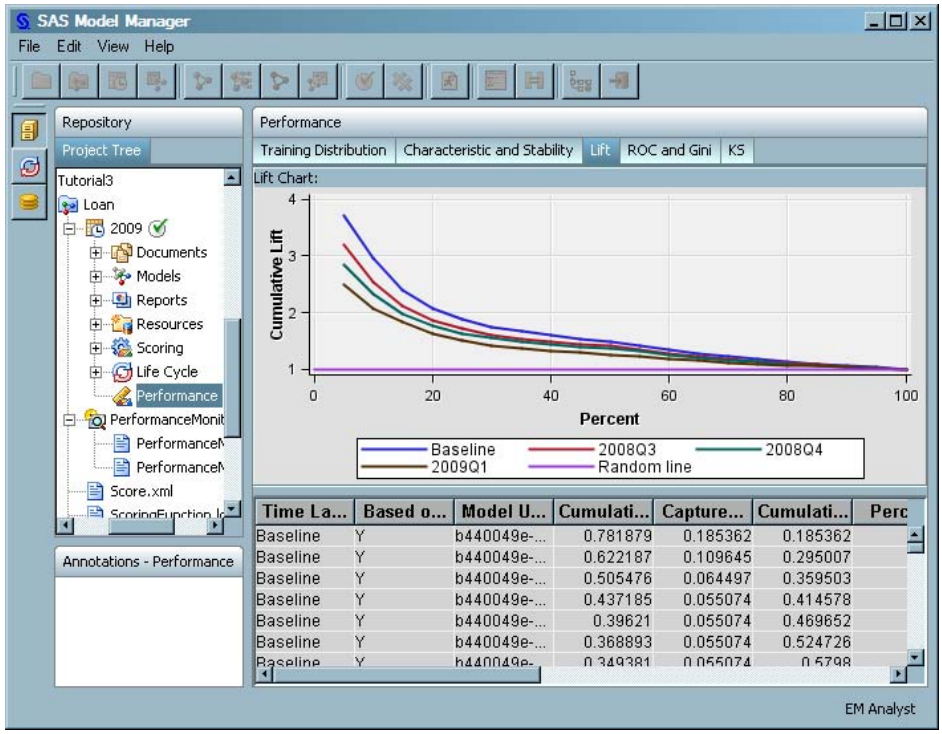
1. Select the **Characteristic & Stability** tab.

2. Select table entries to highlight the corresponding chart points.

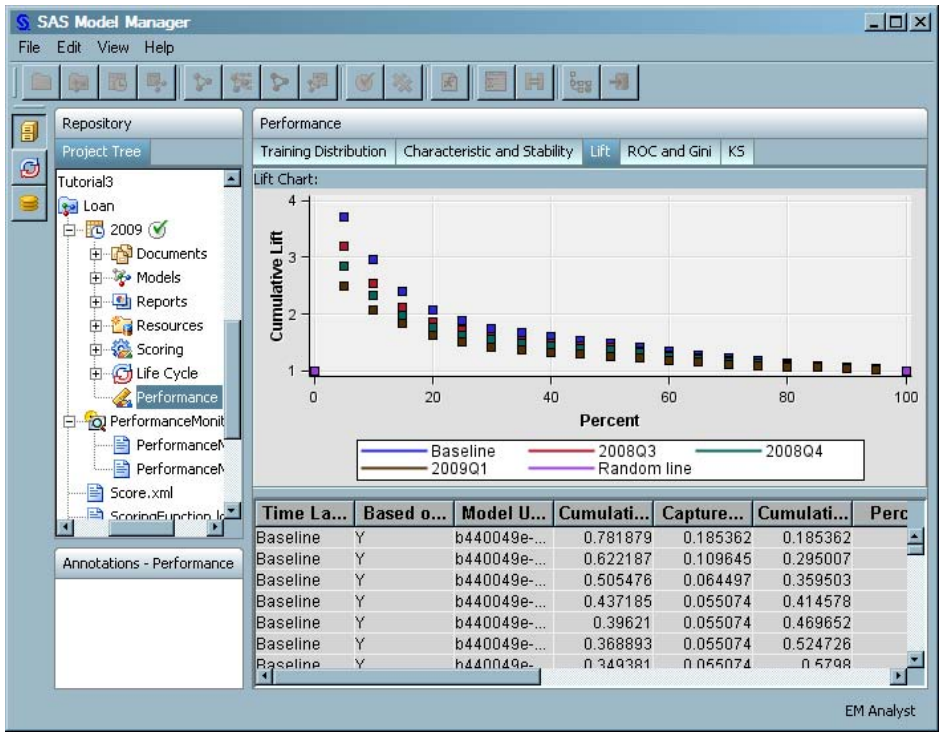
View the Lift Chart

To demonstrate the Lift chart features:

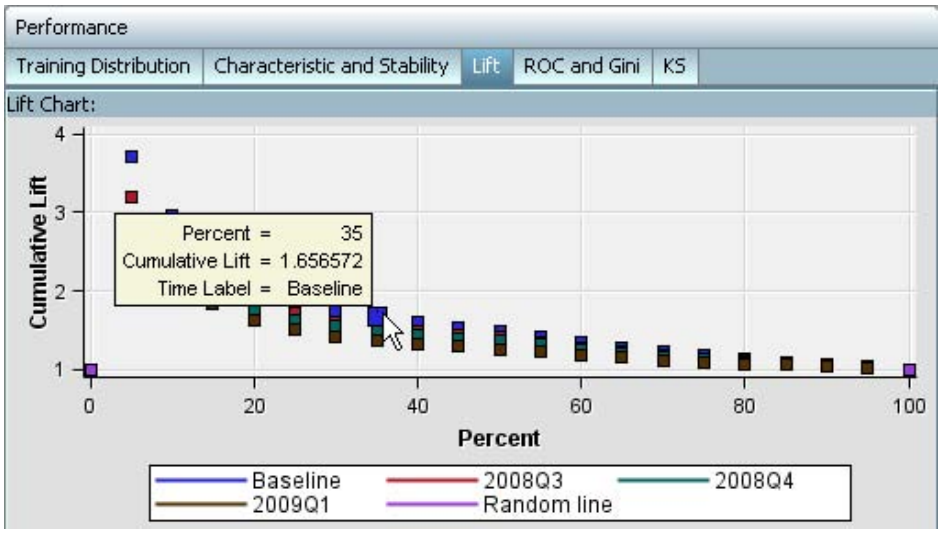
1. Select the **Lift** tab.



2. To change the chart to a plot chart, right-click the chart and select **Chart Type ⇨ Plot**



3. Move the pointer along one lift plot. You should see a pop-up box when the pointer is resting on a data point or is close to a data point.

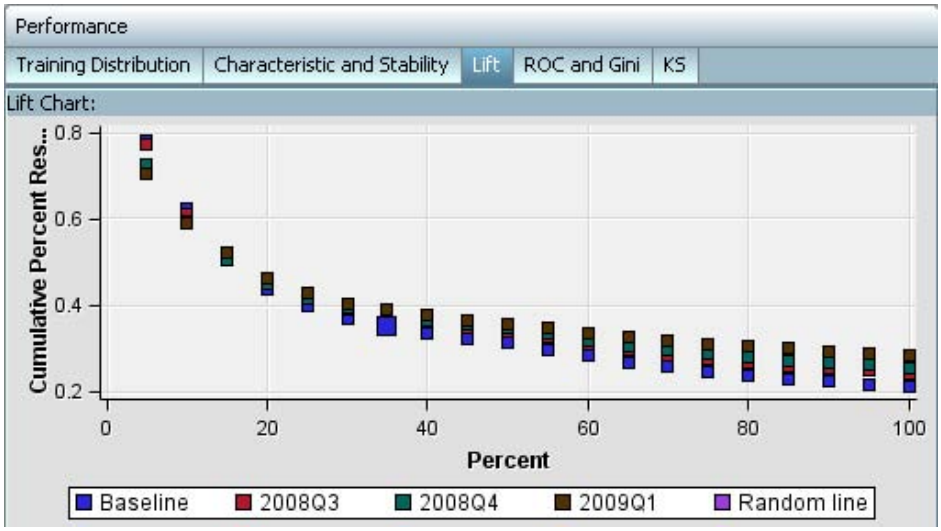


4. To chart the cumulative captured response:

   a. Right-click the chart area.

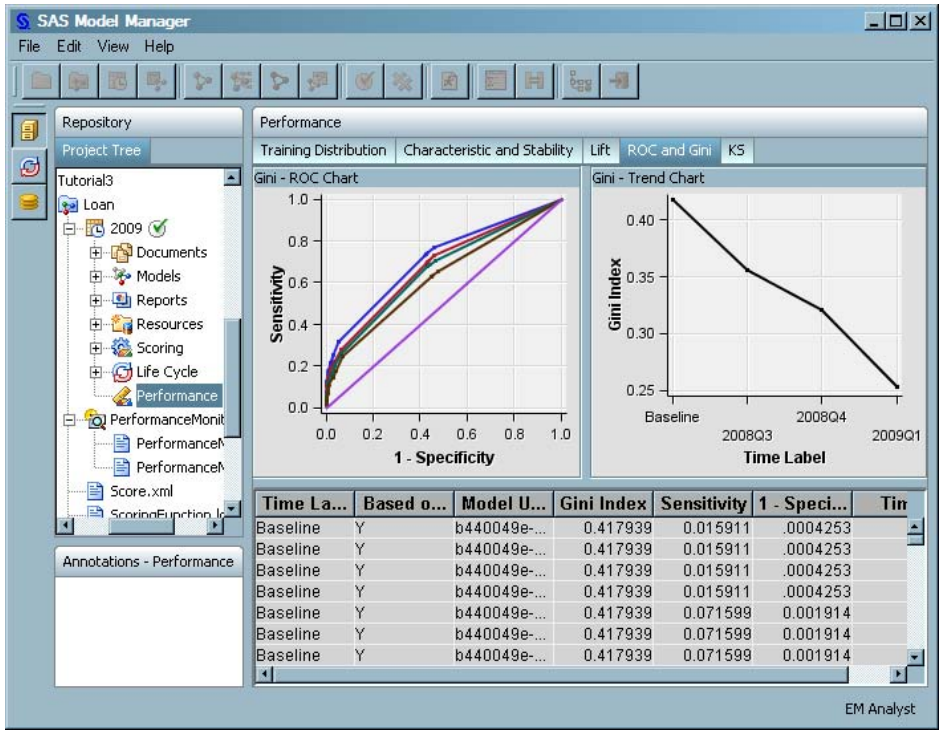
   b. Select **Data Options**.

5. Select **Role Y** for variable CuCapturedResp and click **OK**.



View the ROC & Gini Chart

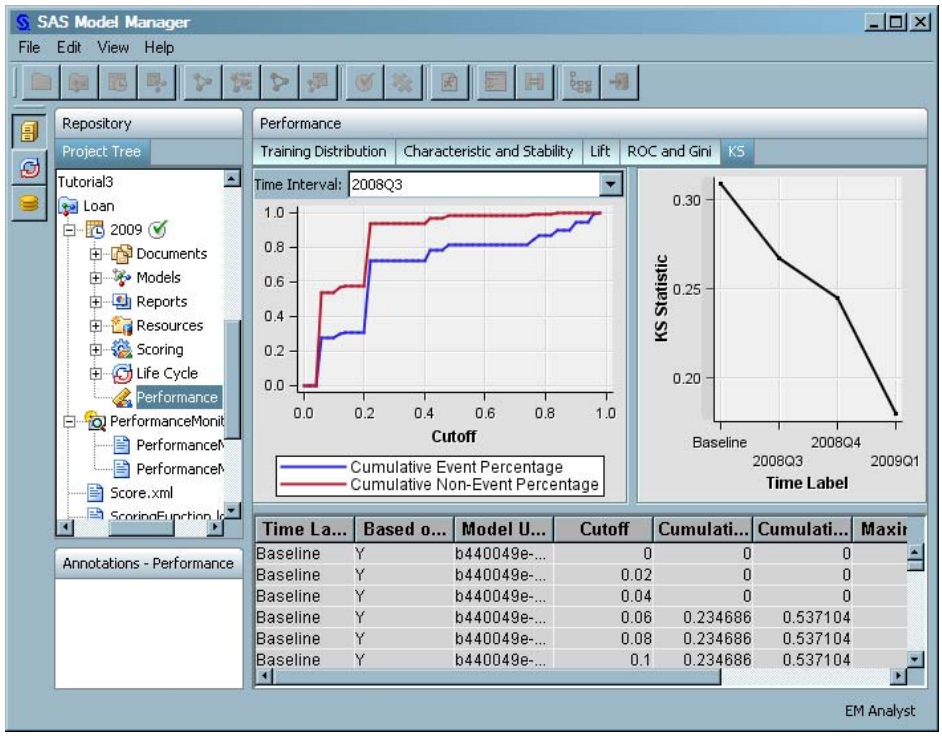
To view the ROC & Gini charts, select the **ROC & Gini** tab.



View the Kolmogorov-Smirnov (KS) Chart

To view the KS chart from the SAS Model Manager user interface, follow these steps:

1. Select the **KS** tab.

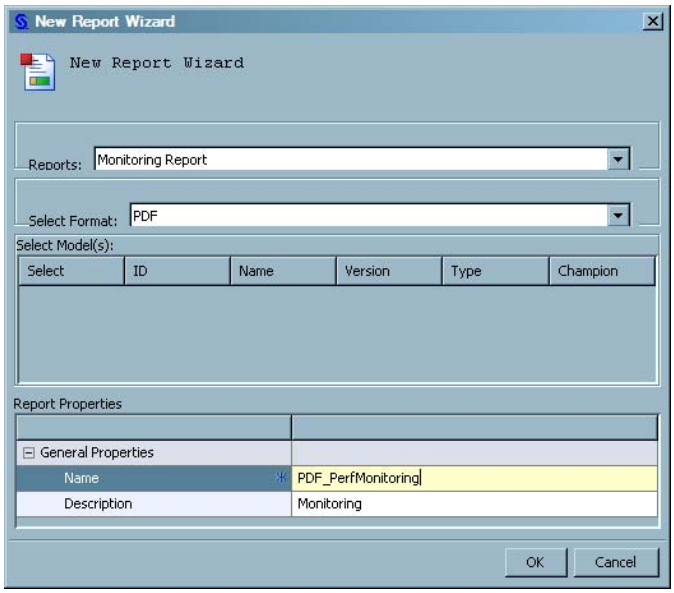


2. Select a different time point from the drop-down list of the **Time Interval** field.

Creating Output Formats for Performance Monitoring Reports

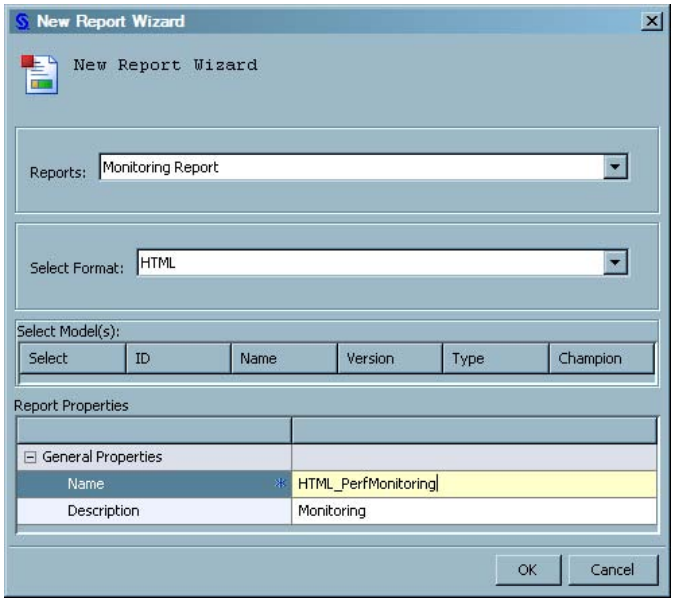
In this exercise you use the New Report Wizard to create the monitoring reports in PDF and HTML output formats.

1. In the Project Tree, expand **Tutorial3**, the **Loan** project, and the version **2009**.
2. Right-click **Reports** and select **Reports** ⇨ **New Report Wizard**.
   a. Click the **Reports** box and select **Monitoring Report**.
   b. In the **Name** box of the **General Properties**, enter **`PDF_PerfMonitoring`**.



   c. Click **OK**. An information message indicates whether the report creation was successful. Click **OK** to close the message box.
   d. View the PDF performance monitoring report. Expand the **Reports** folder. Right-click **PDF_PerfMonitoring** and select **Reports** ⇨ **View Report**. Scroll through the report or click a link in the table of contents to view various parts of the report.
3. Create the same report in HTML.

   Right-click **Reports** and select **Reports** ⇨ **New Report Wizard**.
   a. Click the **Reports** box and select **Monitoring Report**.
   b. Click the **Select Format** box and select **HTML**.
   c. In the **Name** box of the **General Properties**, enter **`HTML_PerfMonitoring`**.



d. Click **OK**. An information message indicates whether the report creation was successful. Click **OK** to close the message box.

e. View the HTML performance monitoring report. Expand the **Reports** folder. Right-click **HTML_PerfMonitoring** and select **Reports** ⇨ **View Report**. All charts and data appear on a single HTML page. Scroll through the report to view various parts of the report.

Customize Model Monitoring Reports

In this exercise, you add a customized title to the performance monitoring report by modifying the SAS code that SAS Model Manager created in order to format the performance monitoring data.

1. In the **Reports** folder, expand **HTML_PerfMonitoring** and click **taskCode.sas**. The report program appears in the Content pane.
2. Scroll to the bottom of the report code. Select and copy the following code:

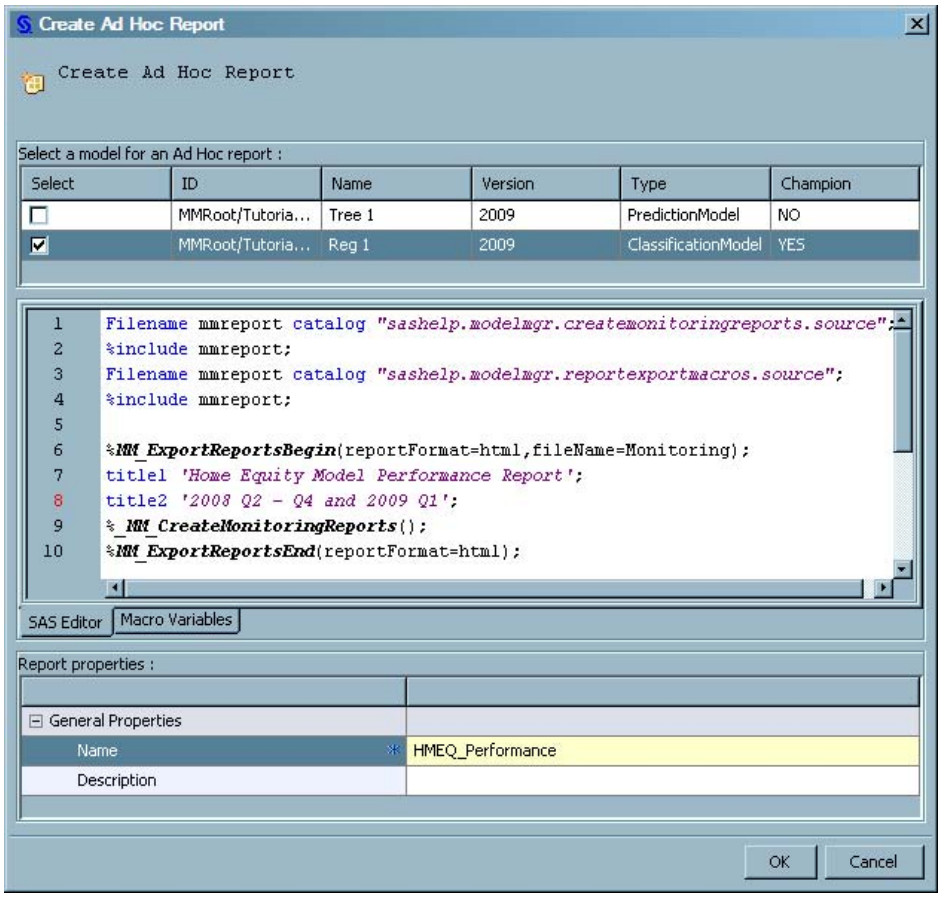
```
Filename mmreport catalog "sashelp.modelmgr.createmonitoringreports.source";
%include mmreport;
Filename mmreport catalog "sashelp.modelmgr.reportexportmacros.source";
%include mmreport;

%MM_ExportReportsBegin(fileName=Monitoring);
%_MM_CreateMonitoringReports();
%MM_ExportReportsEnd;
```

3. Right-click **Reports** and select **Reports** ⇨ **Create Ad Hoc Report**.
4. Select the box for the **Reg1** model.
5. In the SAS Editor, paste the code that you copied.
6. Modify these lines of code so that they match the following code:

```
%MM_ExportReportsBegin(reportFormat=html, fileName=Monitoring);
title1 'Home Equity Model Performance Report';
```

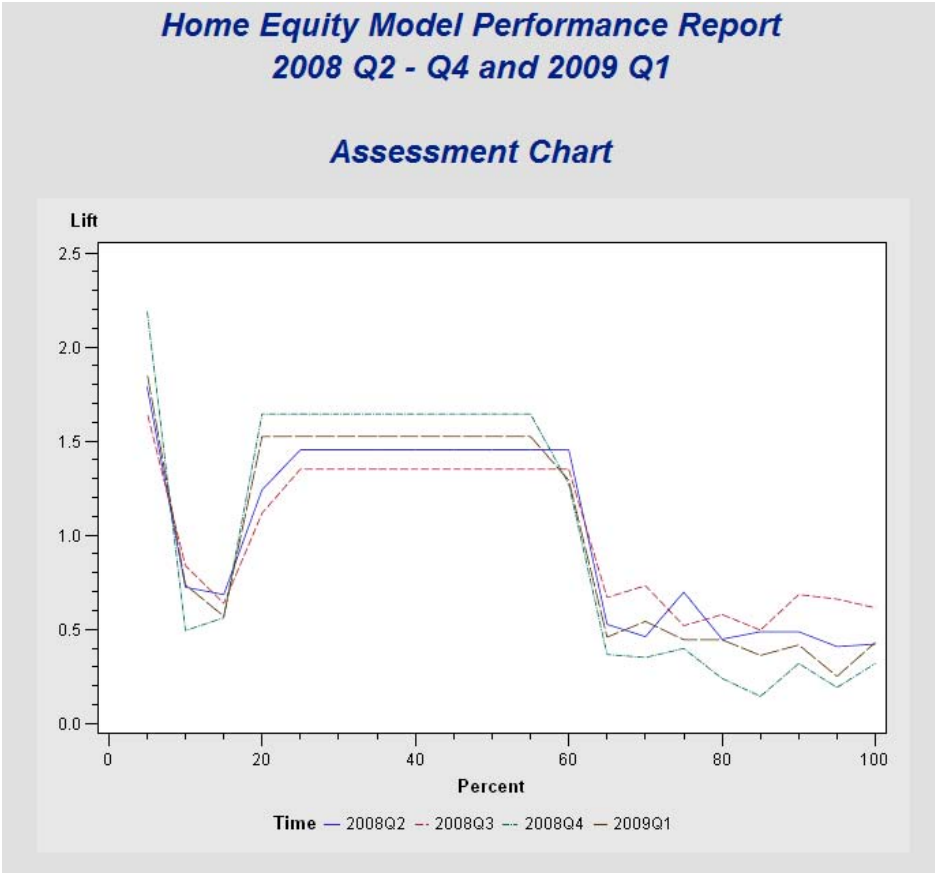
```
title2 '2008 Q2 - Q4 and 2009 Q1';
%_MM_CreateMonitoringReports();
%MM_ExportReportsEnd(reportFormat=html);
```

7. In the **Name** box, enter **`HMEQ_Performance`**.



8. Click **OK**. An information message indicates whether the report was successfully created. Click **OK** to close the message box.

9. In the **Reports** folder, expand **HMEQ_Performance**, right-click **Monitoring.html**, and select **Open Item**. Each chart has the new title. Here is the Lift chart:


Home Equity Model Performance Report
2008 Q2 - Q4 and 2009 Q1
Assessment Chart
Lift
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Percent
Time 2008Q2 2008Q3 2008Q4 2009Q1

Chapter 9

Tutorial 8: Scoring a SAS Model Manager Model Using SAS Data Integration Studio

Overview of Using Exported Models in SAS Data Integration Studio

The SAS Model Manager export feature enables you to register models and projects in SAS Metadata Repositories, making them available for other SAS products such as SAS Data Integration Studio and SAS Enterprise Guide. SAS Model Manager offers two export methods:

- Export a model
- Export a champion model

When you export a SAS Model Manager project champion model to the metadata repository, the result is a mining results object that contains the champion model of the project's default version. If the champion model in the project's default version is changed and the model is exported from the project level again to the same SAS metadata folder, the mining results object in the metadata repository is updated with the new champion model.

To illustrate an application that can use an exported SAS Model Manager project champion model, this tutorial uses SAS Data Integration Studio to connect metadata objects (including a mining results object) to create a scoring job. The tutorial also covers reexporting the project champion model with new content and updating the scoring job.

Prepare Tutorial Data Sets and Models

In this exercise you create a library in SAS Management Console for the data tables that are used in this tutorial.

Models Used in Tutorial 3

The exercises in this tutorial depend on some of the properties of the specific models that were added in Tutorial 3. Although it should be possible to substitute other projects, versions, or models in place of those mentioned in the following actions, this tutorial is written to follow Chapter 4, "Tutorial 3: Importing and Exporting Models," on page 55.

The Required Tutorial Files

The SAS data sets that are required for this tutorial are on your local computer after you extract them from the ZIP file SMM22Tutorial.zip. If you have not extracted the tutorial files, see "The Tutorial Files" on page 3.

This tutorial requires the following files and folders in the **<*drive*>\Tutorial8\Samples** folder:

- score_input.sas7bdat
- score_output.sas7bdat

Define a Data Library in SAS Management Console

Use SAS Management Console to define a data library and register the tables. Follow these steps:

1. Open SAS Management Console and log on to the SAS Metadata Server. Your user ID must be authorized to modify libraries in the metadata server.
2. Open the New Library Wizard to define the data library. Click **Environment Management** ⇨ **Data Library Manager** on the **Plug-ins** tab. Right-click **Libraries** and select **New Library** from the pop-up menu.
3. In the New Library Wizard, create a SAS library.
   a. Select **SAS BASE Library** and click **Next**.
   b. Specify **MM Tutorial-8** in the **Name** box.
   c. Ensure that the **Location** box specifies **/SharedData/ModelManager**, and click **Next**.
   d. (Optional) Depending on your configuration, if more than one server exists, then select a server.
   e. Specify **smm2tor8** for the libref and click **New**.
   f. Specify the server folder that you previously created, **<*drive*>\Tutorial8\Samples**, for the path specification and click **OK** twice.
   g. Click **Next** and **Finish**.

4. Register the data tables in the metadata repository.
   a. Right-click **MM Tutorial-8** under the **Libraries** node, select **Register Tables** from the pop-up menu, and click **Next**.
   b. If prompted, specify the user ID and password to the metadata server and click **OK**. Then click **Next**.
   c. Click **Select All Tables** and click **Browse**. The Select a Location dialog box appears.
   d. From the **Look in** box, select **Model Manager**. Click the **New folder** icon and enter `Tutorial8`. Double-click **Tutorial8**. Click **OK**.
   e. Click **Next**, and then click **Finish**.
5. Verify that table metadata was created. Select **MM Tutorial-8** under the **Libraries** node and examine the right pane.

Export a Project Champion Model from SAS Model Manager

In this exercise, you export a project champion model from SAS Model Manager in order for that model to be accessed and scored by SAS Data Integration Studio. When you export a project, you export the champion model from the project's default version.

Note: This task requires that you use a user ID that is a member of the SAS Model Manager Advanced User group or the SAS Model Manager Administrator group.

Note: If you create user-defined properties at the project level, these properties are exported with the champion model. User-defined properties might be helpful for scoring applications that search mining result objects for specific name value pairs. Each user-defined project property is stored in the SAS metadata repository as an Extension metadata object, which is a name-value pair.

1. Click the Project perspective button  . Expand **Tutorial3**.
2. Right-click **Loan** and select **Export Project Champion Model**. The Metadata Repository dialog box appears.
3. Double-click **Shared Data**. Double-click **Model Manager** and click **Tutorial3**. An information message indicates whether the champion model was successfully exported. Click **OK** to close the message box.

Score a Model Using a SAS Data Integration Studio Job

In this exercise, you create a SAS Data Integration Studio scoring job by using the Loan mining result from the SAS Metadata Repository. After you create the job, you run the job and view the output.

Open the SAS Data Integration Studio Desktop

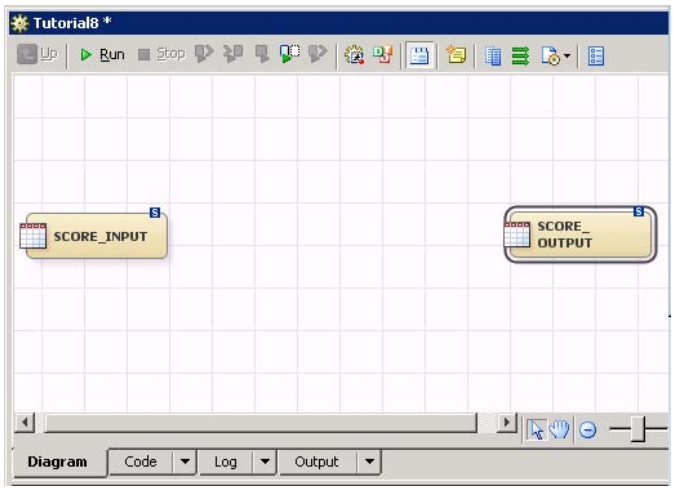
To log on to SAS Data Integration Studio:

1. Launch SAS Data Integration Studio.
2. If prompted, create a SAS Metadata Profile for the SAS Metadata server.
3. Log on with the profile for SAS Metadata server.

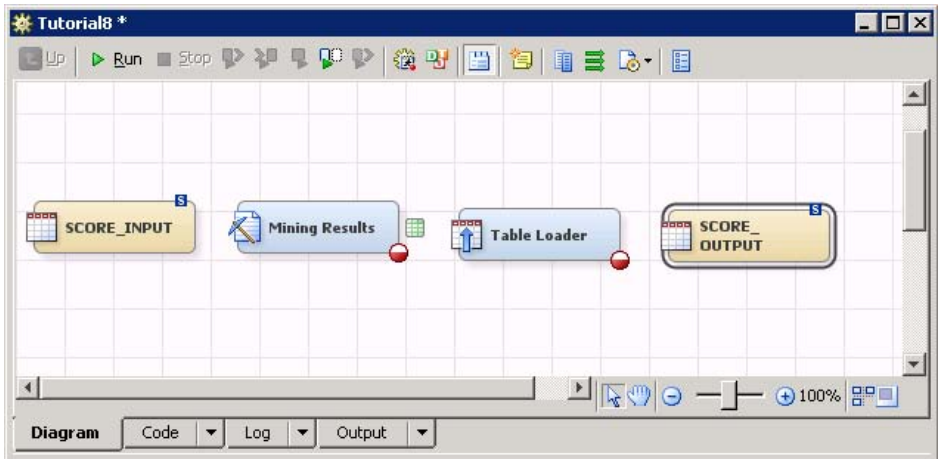
Create a New Job

To create a new job, follow these steps:

1. Use the New Job Wizard to add the job:
   a. From the SAS Data Integration Studio desktop, select **My Folder**. Then select **File** ⇨ **New** ⇨ **Job**. The New Job dialog box appears.
   b. In the **Name** box, enter `Tutorial8` and click **OK**.
   c. Click the **Inventory** tab, expand **Table**, and find the tables **SCORE_INPUT** and **SCORE_OUTPUT**.
   d. Click and drag **SCORE_INPUT** to the **Diagram** tab. Click and drag **SCORE_OUTPUT** to the **Diagram** tab. Position the **SCORE_INPUT** node farthest to the left. Position the **SCORE_OUTPUT** node farthest to the right. These nodes are the beginning and ending nodes in the diagram. Leave enough space between them for two additional diagram nodes.



   e. Click the **Transformations** tab and expand **Access**. Select and drag **Table Loader** to the **Diagram** tab. Place **Table Loader** node before the **SCORE_OUTPUT** node.
   f. From the **Transformations** tab, expand **Data**. Select and drag **Mining Results** to the **Diagram** tab. Place the **Mining Results** node between the **SCORE_INPUT**node and the **Table Loader** node. Here is the **Diagram** tab:



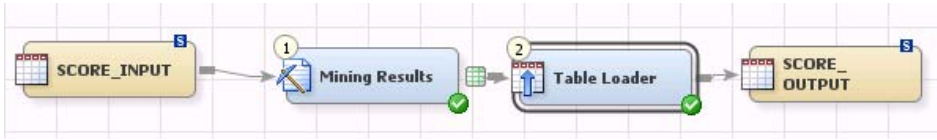
g. Double-click the **Mining Results** node. The Mining Results Properties window opens. Click the **Mining Results** tab, expand **Mining results**, and select **Loan**. The UUID in the **Key** box is the UUID of the **Loan** project in SAS Model Manager.

2. Click the **Target Table Columns** tab. Expand **OutputTable**, select **score**, and click . Click **OK**.

3. Drag the output handle from the **SCORE_INPUT** node to the **Mining Results** node. The half-filled circle on the **Mining Result** node is changed to a check mark to indicate that the node requirements have been met.

4. Drag the output handle from the **Mining Results** node to the **Table Loader** node.

5. Drag the output handle from the **Table Loader** node to the **SCORE_OUTPUT** node. The half-filled circle on the **Table Loader** node is changed to a check mark to indicate that the node requirements have been met. Here is the diagram:

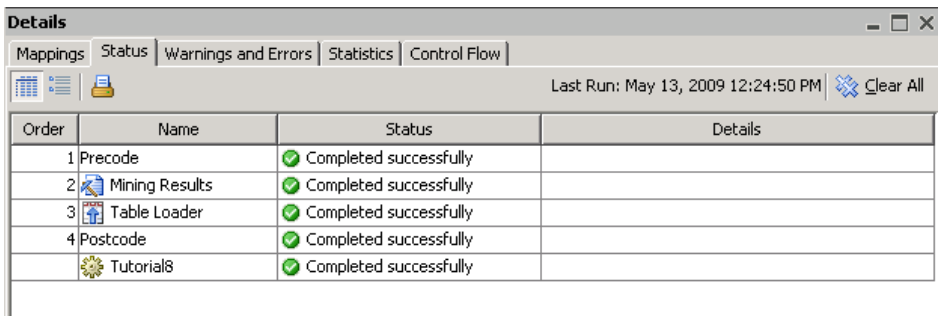


6. Save the job. Click **File** ⇨ **Save** .

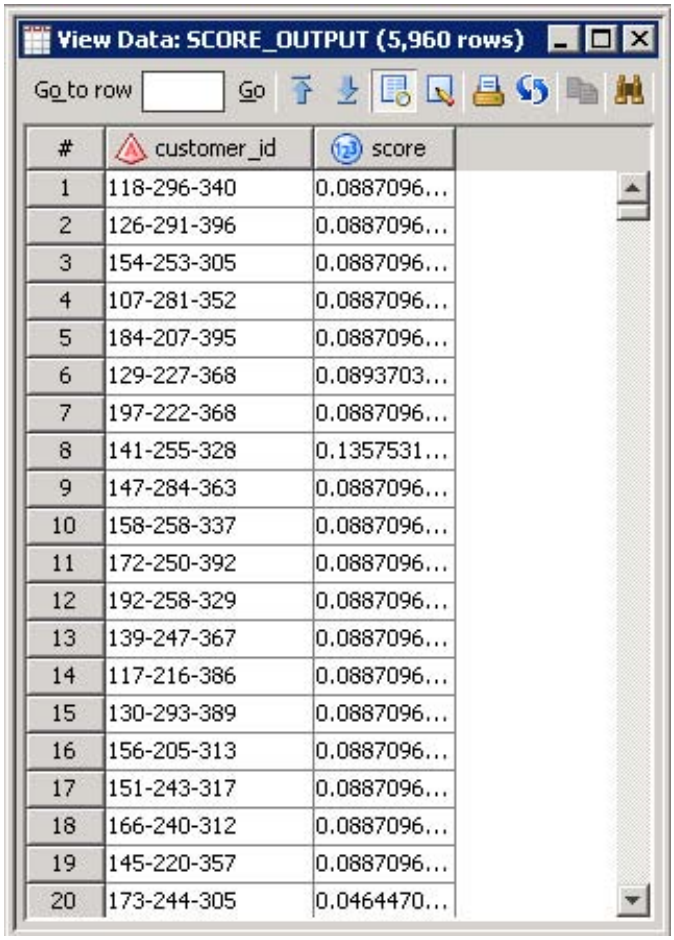
Run the SAS Data Integration Studio Scoring Job

To run the job and view the output, follow these steps:

1. On the **Diagram** tab, select Run. The Tutorial8 job runs. Here is the job status:



2. To view the output, right-click the **SCORE_OUTPUT** node and select **Open**. Here is the output:



View Data: SCORE_OUTPUT (5,960 rows)

Go to row Go

#	customer_id	score
1	118-296-340	0.0887096...
2	126-291-396	0.0887096...
3	154-253-305	0.0887096...
4	107-281-352	0.0887096...
5	184-207-395	0.0887096...
6	129-227-368	0.0893703...
7	197-222-368	0.0887096...
8	141-255-328	0.1357531...
9	147-284-363	0.0887096...
10	158-258-337	0.0887096...
11	172-250-392	0.0887096...
12	192-258-329	0.0887096...
13	139-247-367	0.0887096...
14	117-216-386	0.0887096...
15	130-293-389	0.0887096...
16	156-205-313	0.0887096...
17	151-243-317	0.0887096...
18	166-240-312	0.0887096...
19	145-220-357	0.0887096...
20	173-244-305	0.0464470...

Verify the Model Code Used in the Job

To verify that you have used the correct model, view the model code that was used in the SAS Data Integration Studio job.

Click the **Code** tab and scroll down through the lines until you find the following comment block:.

```
*-------------------------------------------------------------*;
* Macro variable identifying the scored data set;
*%let EM_SCORE_OUTPUT=;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Input Data Source;
* TYPE: SAMPLE;
* NODE: Ids;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Regression;
* TYPE: MODEL;
* NODE: Reg;
*-------------------------------------------------------------*;
**********************************;
*** begin scoring code for regression;
**********************************;
```

The **NODE** value that is associated with **TYPE: MODEL** is the model name. In this case, the model name is **Reg**.

Declare and Export a New Champion Model in SAS Model Manager

In this exercise, you declare a different model as the champion model after an initial project champion model has been exported to the SAS Metadata Repository. You then export the new project champion model to the metadata repository.

To declare and export a new champion model, do the following:

1. In the SAS Model Manager Project Tree, expand the **Tutorial3** folder, the **Loan** project, the **2009** version and the **Models** folder.
2. Right-click the **Tree1** model and select **Set Champion Model**.
3. When prompted to confirm the change, select **Yes**.
4. Right-click the **Loan** project and select **Export Project Champion Model**. The Metadata Repository window opens.
5. Double-click **Shared Data**, double-click **Model Manager**, select **Tutorial3**, and click **OK**.

   An information message indicates whether the champion model was successfully exported. Click **OK** to close the message box.

For more information about this task, see the *SAS Model Manager 2.2: User's Guide*.

Update the Job to Use the Latest Champion Model

This exercise demonstrates the steps to update the SAS Data Integration Studio job after you change the champion model in SAS Model Manager. After you export the **Loan** project from SAS Model Manager, SAS Data Integration Studio recognizes a new mining results object.

To update the job, follow these steps:

1. Close and reopen Tutorial8.
   a. Click the **Folders** tab and select **Tutorial8**.
   b. Select **File** ⇨ **Close**.
   c. Double-click **Tutorial8** to reopen the job. When the job reopens, it uses the new score code from the Mining Result object and regenerates the code that is associated with the job.
2. In the **Tutorial8** diagram, right-click the **Mining Results** node and select **Properties**. The Mining Results Properties window opens. Click the **Mining Results** tab. The **Loan** mining result is highlighted. The **Algorithm** box shows that the model is a **DecisionTree** model.
3. Click the **Model Attributes** tab. A message box might appear that warns of potential table changes if you change the mining result. Click **Yes**.

4. Click **View Source Code**. Scroll to the top of the window. Compare the text in the comment tags to the Tree 1 model code in SAS Model Manager. They are the same, as shown in the displays below.

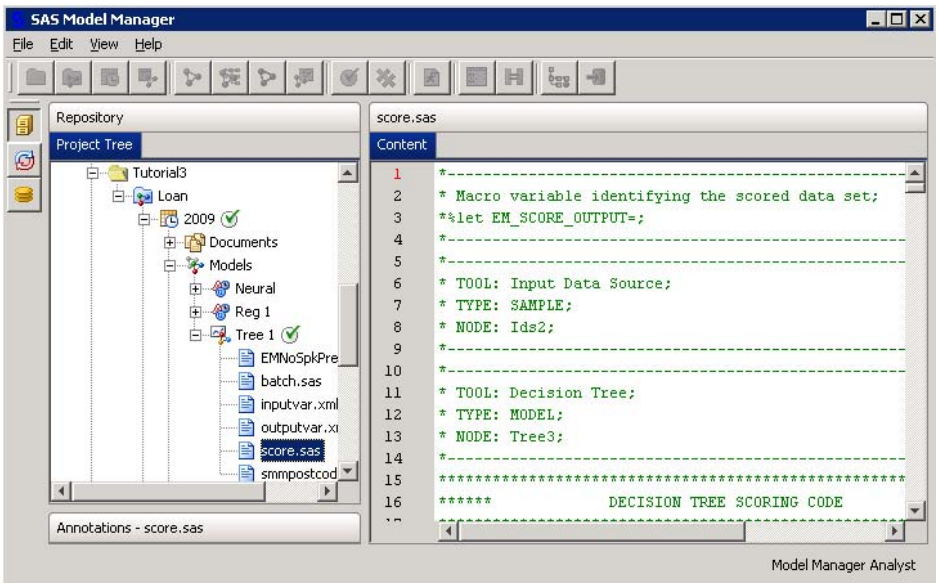
   To view the model code in SAS Model Manager, follow these steps:

   a. Log on to SAS Model Manager and expand the following Project Tree nodes:

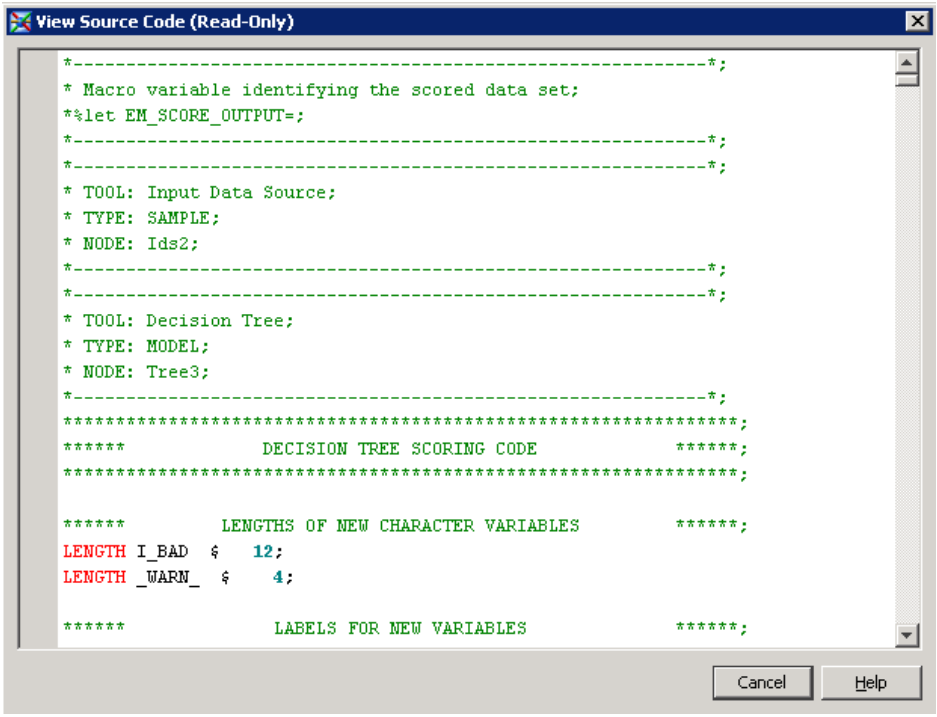
      - **Tutorial3** folder
      - **Loan** project
      - **2009** version
      - **Models** folder
      - **Tree 1** model

   b. In the **Tree 1** model, select **score.sas**. The model code appears in the **Content** view.

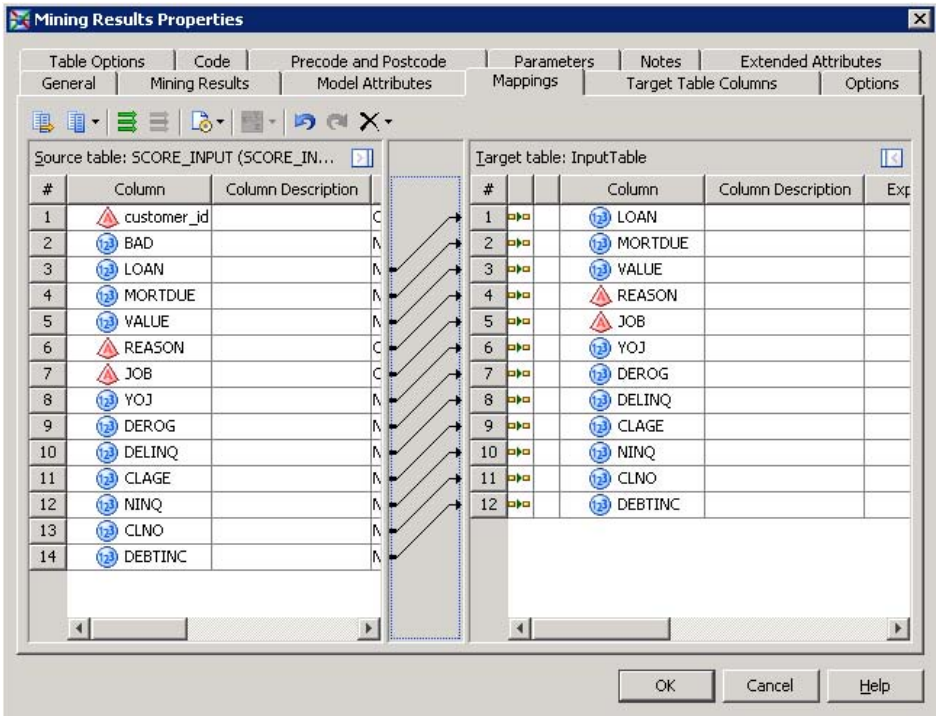
   Here is the Tree 1 score code in SAS Model Manager.



   Here is the Tree 1 score code in SAS Data Integration Studio.

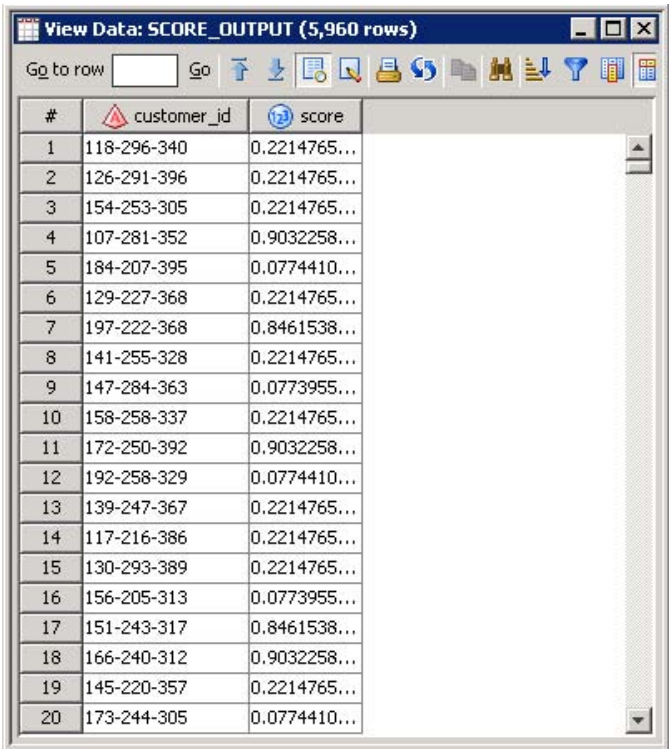


5. Click **Cancel** to close the View Source Code window.
6. Identify the variables to be used in the transform output. Click the **Target Table Columns** tab:
   a. From the **Available** list, select **score** and click  to move the **score** variable to the **Selected** list.
   b. From the **Available** list, expand **SCORE_INPUT**, select **customer_id**, and click  to move the **customer_id** variable to the **Selected** list.
7. Click the **Mappings** tab. Right-click the space between the two lists of variables and select **Map All**. Here is the **Mappings** tab:



Click **OK**.

8. Double-click the **Table Loader** node and click the **Mappings** tab.
9. Right-click the space between the two lists of variables and select **Map All**. Click **OK**.
10. On the **Diagram** tab, select Run. The **Tutorial8** job runs.
11. To view the output, right-click the SCORE_OUTPUT node and select **Open**. Here is a partial view of the output:



View Data: SCORE_OUTPUT (5,960 rows)

Go to row Go

#	customer_id	score
1	118-296-340	0.2214765...
2	126-291-396	0.2214765...
3	154-253-305	0.2214765...
4	107-281-352	0.9032258...
5	184-207-395	0.0774410...
6	129-227-368	0.2214765...
7	197-222-368	0.8461538...
8	141-255-328	0.2214765...
9	147-284-363	0.0773955...
10	158-258-337	0.2214765...
11	172-250-392	0.9032258...
12	192-258-329	0.0774410...
13	139-247-367	0.2214765...
14	117-216-386	0.2214765...
15	130-293-389	0.2214765...
16	156-205-313	0.0773955...
17	151-243-317	0.8461538...
18	166-240-312	0.9032258...
19	145-220-357	0.2214765...
20	173-244-305	0.0774410...

Chapter 10

Tutorial 9: Using the Java Scoring API

Overview of Using the Java Scoring API

SAS Model Manager In-Database Scoring for Teradata (Java Scoring API) works with SAS Model Manager. When you publish a Teradata scoring function for a project, SAS Model Manager exports the project's champion model to the SAS Metadata Repository, and the SAS Scoring Accelerator creates a scoring function inside Teradata. The scoring functions are deployed inside Teradata based on the project's champion model score code. The Java Scoring API enables you to access a scoring model that is stored as a function in the Teradata EDW. As the application developer, you are responsible for creating or maintaining the code that uses the Java Scoring API to submit a scoring request, and parses the scoring result that is returned.

This tutorial shows you how to use the Java Scoring API with Teradata. It contains examples and step-by-step directions for preparing to use the Java Scoring API and shows you how to modify the associated Java code to produce your desired results.

This tutorial demonstrates the following tasks:

- interact with the **hmeqid** Teradata table.
- select a record to score against by using the **Loan** project champion model that was created in Tutorial 3 and then used in Tutorial 5.
- perform what-if scenarios by locally altering the data and resubmitting the score request.

Prerequisites

The exercises in this tutorial depend on the preparation steps and some of the properties of the specific projects that were added in Chapter 6, "Tutorial 5: Publishing Teradata Scoring Functions," on page 83.

Before proceeding with this tutorial, complete the following tasks:

1. Download and install Grails 1.1. Follow the installation instructions on the Grails Web site (**`http://www.grails.org/installation`**).

   Windows Specifics
   For the Windows environment, download the **Binary Zip** file and define the GRAILS_HOME and JAVA_HOME environment variables as system variables.

2. After setting the environment variables, open a new command window (DOS shell) to use for the rest of this tutorial.

3. (optional) If you are behind a corporate firewall you might need to specify your HTTP proxy. To set your HTTP proxy run the following command and complete the requested information when prompted:

   ```
   grails set-proxy
   ```

   Note: You will be asked to supply the name and port of your gateway. You will also be prompted to specify the HTTP user name and password. Contact your system administrator for more information.

4. To create the scoring application for this tutorial, follow these steps:

   a. Create a new directory called **SMMtutorial9** (for example, **`C:\SMMtutorial9`** in a Windows environment).

   b. In the command window, navigate to the **SMMtutorial9** directory that you created in the previous step.

   c. Run the following command to create the directory structure:

      ```
      grails create-app scoreTutorial
      ```

   d. Navigate to the **`scoreTutorial`** directory in the command window to verify that it has been created.

Prepare to Use the Java Scoring API

In this exercise you configure your environment to use the Scoring Tutorial example application (Java Scoring API). To perform a quick setup for this tutorial, use the zip file that contains the files that must be made available to the Scoring Tutorial example application. The zip file, scoreTutorial.zip, replaces the contents of the **`scoreTutorial\grails-app`** directory. To prepare the Scoring Tutorial example application for use follow these steps:

1. Locate the scoreTutorial.zip file in the **Tutorial9** folder that was created when you extracted the contents of the SMM22Tutorial.zip. For more information, see SAS Model Manager Tutorials on page 3.

2. Unzip that file into the ***SMMtutorial9*\scoreTutorial** directory and click **Yes to All**, to replace existing files.

3. Three jar files must be added to the ***SMMtutorial9*\scoreTutorial\lib** directory in order for the Java Scoring API to access a scoring function that was published to Teradata by the SAS Model Manager Client. Two of the jar files are the Teradata JDBC driver jars and the third is the SAS jar that contains the Java Scoring API.

   To obtain the jars and place them in the ***SMMtutorial9*\scoreTutoriallib** directory, follow these steps:

   a. Ask your system administrator to give you the **sas.modelmanager.td.jar** file. The file is located on the middle-tier server where the SASModelManagerInDatabaseScoringForTeradata SAS application is installed.

      Here is an example of the directory path on a Windows server: **C:\Program Files\SAS\SASVersionedJarRepository\9.2\eclipse\plugins\sas.modelmanager.td_***.

   b. To download the Teradata JDBC driver from the Teradata Web Site (http://www.teradata.com), select **Support & Downloads ⇨ Downloads ⇨ Teradata JDBC Driver**, and select the version of the JDBC driver that supports Teradata version 12.

   c. Place the **sas.modelmanager.td.jar** and Teradata JDBC driver jars (**terajdbc4.jar** and **tdgssconfig.jar**) in the ***SMMtutorial9*\scoreTutorial\lib** directory.

4. Obtain the connection information for the Teradata database and enter that information into the Scoring Tutorial application.

   Ask the Teradata database administrator for the following information:

   - host name (server host name, where the Teradata EDW resides)
   - database name
   - user ID
   - password

5. Edit the **DataSource.groovy** file and replace the existing values with the information that you obtained in the previous step. The file is located in the **scoreTutorial\grails-app\conf** directory.

   Here is an example of the **DataSource.groovy** file. The database connection information that you specify in this file enables the application to connect to the Teradata database and enables you to work with the contents of the **hmeqid** table. The variable values in italics are user-supplied information.

```
dataSource {
    pooled = false
    driverClassName = "com.teradata.jdbc.TeraDriver"
    dialect = "org.hibernate.dialect.TeradataDialect"
    username = "UserID"
    password = "Password"
}

hibernate {
    cache.use_second_level_cache=true
    cache.use_query_cache=true
    cache.provider_class='com.opensymphony.oscache.hibernate.OSCacheProvider'
```

```
}

// environment specific settings
environments {
    development {
        dataSource {
            url = "jdbc:teradata://HostName/DATABASE=DatabaseName"
        }
    }
        test {
          dataSource {
            dbCreate = "update"
            url = "jdbc:hsqldb:mem:testDb"
          }
        }
        production {
          dataSource {
            dbCreate = "update"
            url = "jdbc:hsqldb:file:prodDb;shutdown=true"
          }
        }
}
```

6. Edit the **ScoreService.groovy** file and replace the existing values with the connection information that you obtained. The file is located in the ***SMMtutorial9*\scoreTutorial\grails-app\services** directory.

   The ScoreService.groovy file contains the **ScoreService** class and the **runScoring()** method. The content of this file is similar to what you must implement in order to use the Java Scoring API from within your own application. As with the **DataSource.groovy** file, you must update the **HostName**, **UserID**, **Password**, and **DatabaseName** variable values in the **runScoring()** method. This step is necessary in order for the **DataSource.groovy** file to pass the variables into the constructor for the **Scoring** class and establish a connection to the database.

   The ScoreService.groovy file also requires a value for the **projectUUID** variable. You can obtain the correct value to use from one of the following locations:

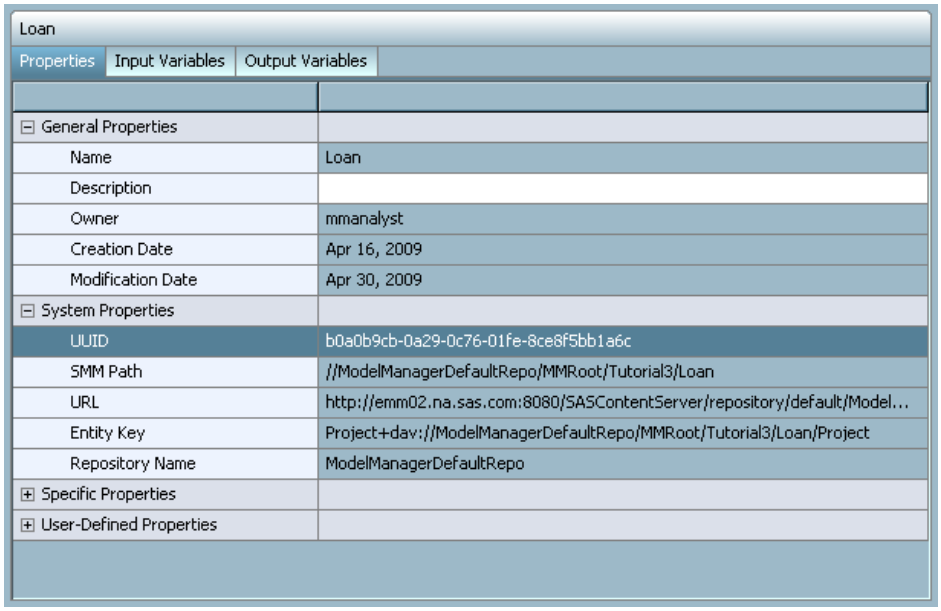
   - The **project_uuid** column in the **project_metadata** table, located in the same database that contains the **hmeqid** table.
   - The SAS Model Manager **UUID** project system property from the **Loan** project.

   Here is a partial example of the **ScoreService.groovy** file. The variable values in italics are user-supplied information.

```
class ScoreService {

    boolean transactional = false

    def String runScoring(GrailsParameterMap params) {

        if (params == null || params.isEmpty())
            return ""
            def projectUUID = "ProjectUUID"
        def host = "HostName"
        def username = "UserID"
        def password = "Password"
        def database = "Database"
```

To retrieve the **ProjectUUID** value from SAS Model Manager, follow these steps:

1. Open SAS Model Manager and select the **Loan** project that you created in the Tutorial 3 folder.
2. On the **Properties** tab, click the plus sign to expand the **System Properties**.



3. Copy the **UUID** property value that you will use for the **ProjectUUID** string value.

Using the Java Scoring API

In this exercise, the Scoring Tutorial example application uses SAS Model Manager Java Scoring API functionality.

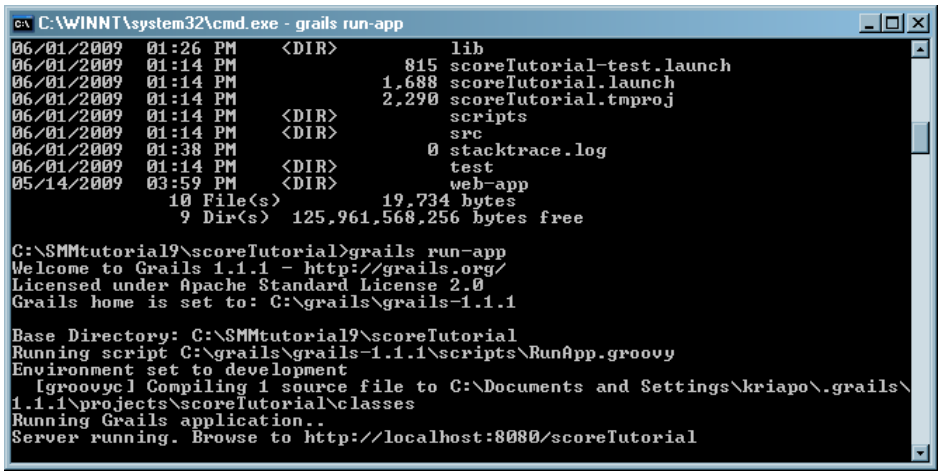
Start the SAS Scoring Tutorial Application

To run the SAS Scoring Tutorial application, follow these steps:

1. Navigate to the **`scoreTutorial`** directory, and then run the following command to compile and run the SAS Scoring Tutorial application (using the Jetty application server):

   ```
   grails run-app
   ```

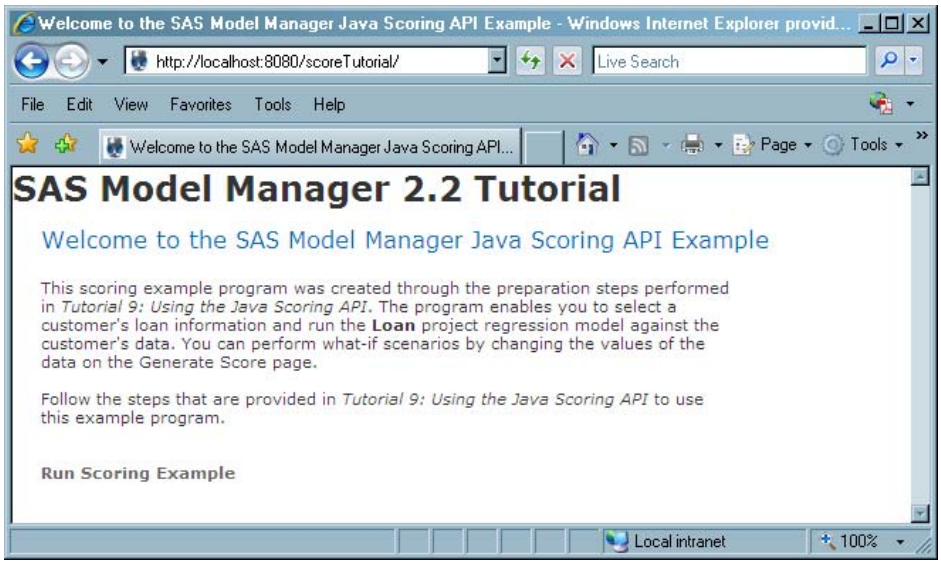
   Note: Keep the command window open after you run the command.



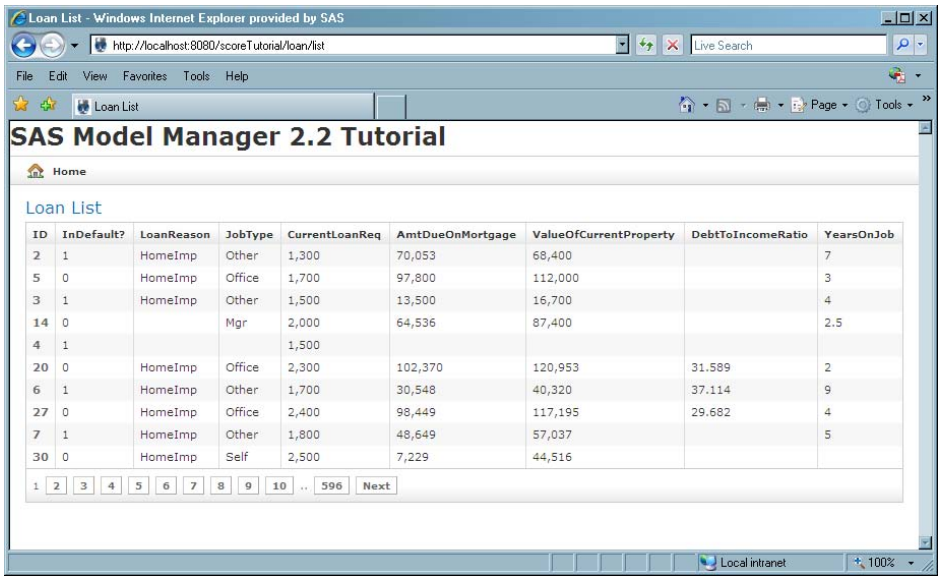
2. (optional) By default, the application attempts to run the server on port 8080. If this port is already in use, then you can use the following command to start the application server on a different port:

   ```
   grails -Dserver.port=8090 run-app
   ```

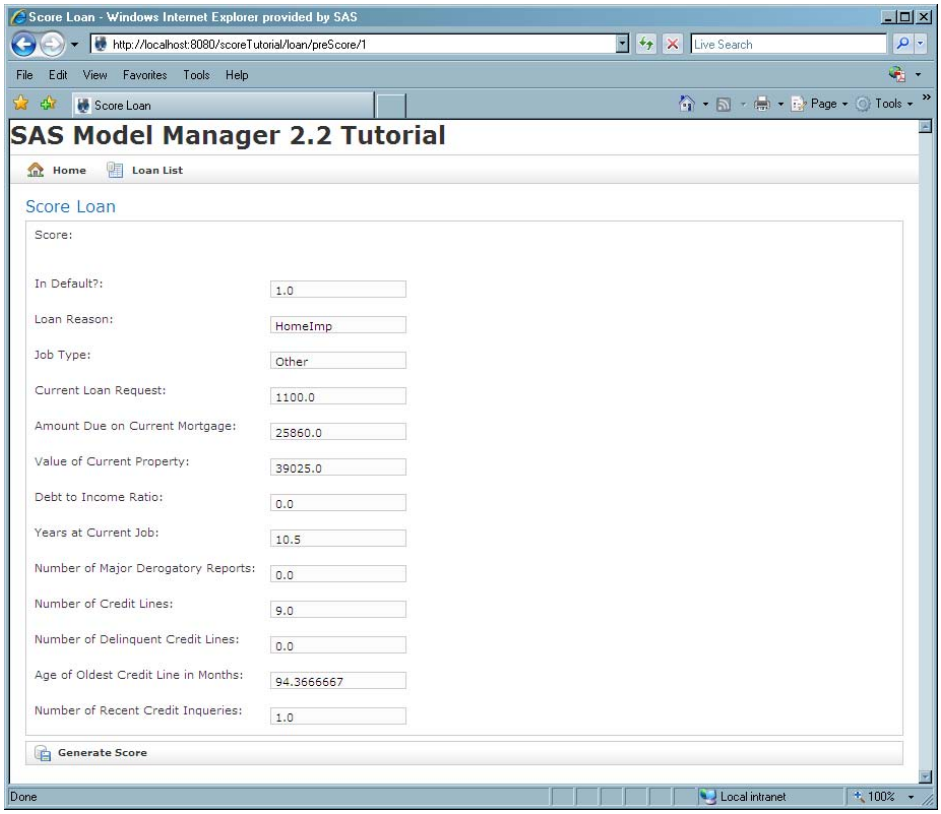
3. Access the application by using the URL that is provided after the compile and run process of the SAS Scoring Tutorial application is complete. The default URL is **`http://localhost:8080/scoreTutorial`**.



4. Click **Run Scoring Example** to display the Loan List page. To sort the list, click one of the column headings.

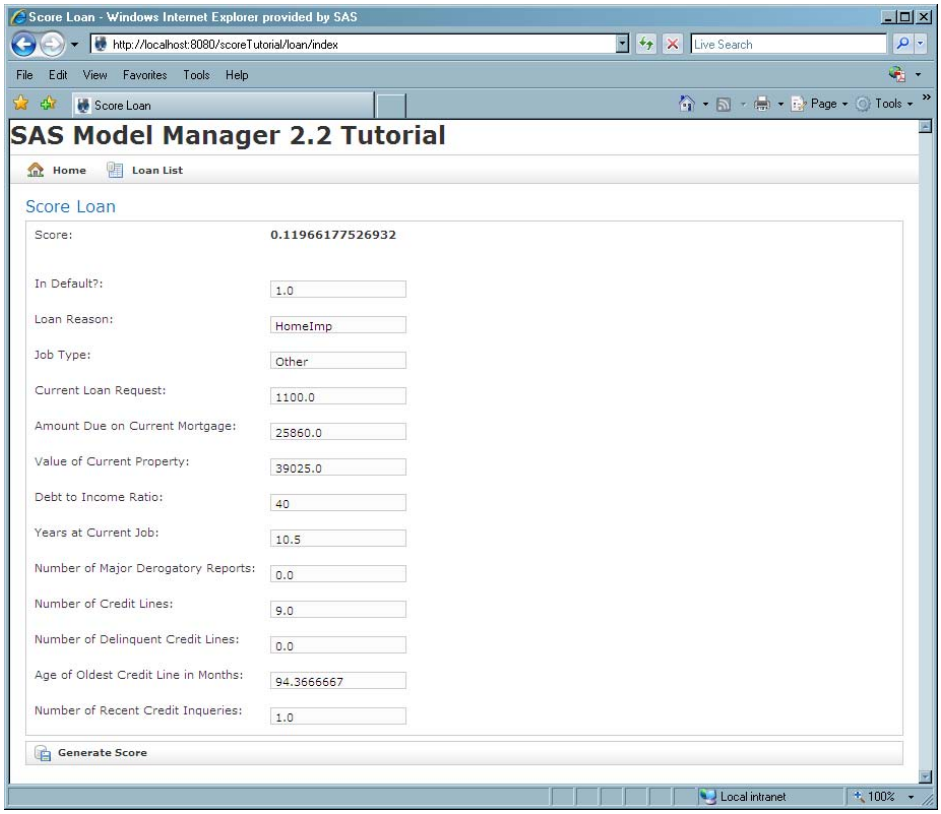


5. Click **ID** to sort the columns, and then click the number in the **ID** column for the customer loan record that you want to score.



6. Click **Generate Score** to use the data in the database to generate a score for the selected customer loan record. Examine the **Score** value.

7. Change one or more of the values and then click **Generate Score**. For example, change the value of **Debt to Income Ratio** from 0 to 40.



8. Click **Loan List** to return to the list of customer loan records or click **Home** to return to view the Scoring Tutorial example overview and directions.

For more information about the **Scoring** and **Results** classes that are used by the Java Scoring API, see "Java Scoring API Classes" on page 148.

Another example is included in the installation directory of the SAS Model Manager In-Database Scoring for Teradata product. The example is a command-line Java program that is named ScoreAPI.java. For more information about the Java Scoring API, see the SAS Model Manager In-Database Scoring for Teradata appendix in the *SAS Model Manager: User's Guide*.

Java Scoring API Classes

About the Java Classes

The Java Scoring API consists of two public classes, the **Scoring** class and the **Results** class. Both of these classes are used in the **`ScoreService.groovy`** file.

Scoring Class

The Scoring class provides the following:

1. The ability to establish a JDBC Connection. If you have an existing connection use the following Java string:

   ```
   Scoring scoringTask = new Scoring(existingJDBCConnection, userId);
   ```

   If you want the Java Scoring API to establish the connection and release it when the scoring task is complete, use the following Java string, which is used in the **`ScoreService.groovy`** file:

```
Scoring scoringTask = new Scoring(fullyQualifiedHostName, userId, password,
databaseName);
```

2. The ability to enable and disable logging of scoring actions. By default logging is enabled and the log is written to the scoring_log table in the Teradata database. The following Java string is used to disable logging of scoring actions:

   ```
   scoringTask.setWriteScoringLog(false);
   ```

3. The ability to perform a scoring request by calling the score() method. The score() method takes three Java string parameters, the **projectId**, the **whereClause**, and the **dynamicPredictors**. Here is an example of a score() method:

   ```
   scoringTask.score(projectId, whereClause, dynamicPredictors);
   ```

   projectId
   The value of the *projectId* variable is associated with the project whose champion model was published with the scoring function to Teradata. This value was obtained when you performed the preparation steps to use the Java Scoring API. To retrieve the value again, obtain it from one of the following locations:

   - The **project_uuid** column in the **project_metadata** table located in the same database which contains the **hmeqid** table.
   - The SAS Model Manager **UUID** project system property from the **Loan** project.

   whereClause
   The *whereClause* argument is an SQL WHERE statement that returns a single record from the table that is being scored against (for example, **hmeqid**). Because this table contains a unique key column, **custid**, the WHERE clause in this tutorial is extremely simple. However, the WHERE clause can be as complex as needed in order to return a single record (confined only by your database's WHERE clause syntax).

   dynamicPredictors
   The *dynamicPredictors* argument specifies predictors to override some or all of the column values for the customer record that you are retrieving. The ScoreService.runScoring() method determines which column values have been modified by the user of the application, and they are included in the *dynamicPredictors* string. These values override, but do not overwrite, the values that are currently in the record that was selected for scoring. Either an empty or a null string value can be passed in as this argument if no predictors must be overridden.

Results Class

The returned result from the score() method call is a Java string. Depending on what the model is set up to return, the string can be a single value, a single name=value pair, or a semicolon-separated series of name=value pairs. You can use the Results class to parse these various combinations and provide a simple interface to pull the values out.

The runScoring() method passes the string that was returned from the score() method into the constructor for the Results class. In the code example, the string that is returned by the runScoring() method is a single name=value pair where the name is **`score`** so the **`results.get("score")`** method obtains the score value and returns that to the caller.

The result can also be accessed by the **`results.get(0)`** method.

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.

SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.

SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

support.sas.com/publishing

SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

support.sas.com/spn

sas | THE POWER TO KNOW®