Paper SAS1921-2015

SAS® Model Manager: An Easy Way for Deploying SAS® Analytical Models to Databases and Hadoop

Jifa Wei and Kristen Aponte, SAS Institute Inc.

ABSTRACT

SAS® Model Manager provides an easy way to deploy analytical models into various relational databases or into Hadoop using either scoring functions or the SAS® Embedded Process publish methods. This paper gives a brief introduction of both the SAS Model Manager publishing functionality and the SAS® Scoring Accelerator. It describes the major differences between using scoring functions and the SAS Embedded Process publish methods to publish a model. The paper also explains how to perform in-database processing of a published model by using SAS applications as well as SQL code outside of SAS. In addition to Hadoop, SAS also supports these databases: Teradata, Oracle, Netezza, DB2, and SAP HANA. Examples are provided for publishing a model to a Teradata database and to Hadoop. After reading this paper, you should feel comfortable using a published model in your business environment.

INTRODUCTION

Using SAS Model Manager, you can publish models to a database or to Hadoop, so that you can perform in-database scoring using SQL code or other SAS applications. Only models that are set as the project champion or as a challenger and are associated with the DATA step score code type can be published. The SAS Scoring Accelerator and SAS/ACCESS® interface to the data management system are used by SAS Model Manager to publish the models and translate them into scoring functions or scoring files for in-database processing. There are two methods by which published models are processed inside the database: scoring functions and the SAS Embedded Process.

Scoring functions are similar to user-defined functions in a database. The SAS Scoring Accelerator is used to convert scoring models into scoring functions. The functions can then be used in SQL statements in the same way as other database functions. The scoring functions publish method is supported for DB2, Greenplum, Netezza, and Teradata.

The SAS Embedded Process runs inside the database to read and write data from the database and to execute the DATA step 2 (DS2) code. After the model scoring files are published to a database or to Hadoop, they are used by the SAS Embedded Process to run the scoring model. The advantage of using the SAS Embedded Process is that a single function or a stored procedure is used instead of multiple user-defined functions. The SAS Embedded Process publish method is supported for Aster, DB2, Greenplum, Hadoop, Netezza, Oracle, SAP HANA, and Teradata.

The examples in this paper demonstrate how you can publish models and then score them from within a database or Hadoop.

PUBLISHING ANALYTICAL MODELS FROM SAS® MODEL MANAGER

SAS Model Manager enables you to organize models using folders and projects. You can use a folder to group related projects. A project enables you to set up a requirement for your business problem. The project is a contract between a modeler and a business department that has a model requirement.

The business requirement for a model project includes the following:

- **A list of variables that can be used by a modeler to build a model.** In SAS Model Manager, these variables are called project input variables.
- **A list of variables that a model produces to solve your business problem.** In SAS Model Manager, these variables are called project output variables.
- **A modeling project type.** A type can be Classification, Prediction, Analytical, or Segmentation.

SAS Model Manager enables users to import models built by SAS® Enterprise Miner™, SAS/STAT® software, SAS® High-Performance Analytics, or a model built by a third-party tool, such as R. It also can import Predictive Modeling Markup Language (PMML) models. You can have more than one model in a project. However, you must set one model as the project champion model before you can publish a model project to a database or to Hadoop. You can also publish one or more challenger models. The champion model and the challenger models must all satisfy the project requirements. The project requirement demands that the champion and challenger models use only a subset of project input variables and produce a superset of project output variables. SAS Model Manager enables you to map a model's output variable to a project's output variable if they are equivalent.

One of the major attributes for a model is its score code type. The source code type can be **SAS DATA step**, **SAS Program**, or **PMML**. The score code for DATA step model contains only SAS DATA step language elements without DATA steps, procedure steps, or SAS macro calls. The score code for a SAS Program model can contain any valid SAS statement. SAS Model Manager can publish only models with the DATA step score code type to a database or Hadoop.

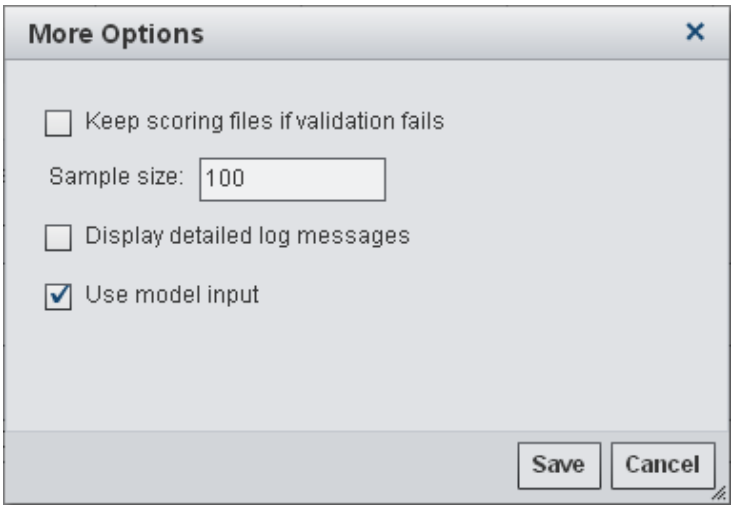
When SAS Model Manager publishes a model to a database or to Hadoop, it uses both the SAS Scoring Accelerator and the corresponding SAS/ACCESS interface. SAS Model Manager enhances a user's experience of using the SAS Scoring Accelerator by providing a graphical user interface (GUI). Furthermore, SAS Model Manager validates a published model to make sure that it produces the same results that are produced for the same model in a SAS environment.

WHAT HAPPENS DURING THE PUBLISHING PROCESS?

The following steps occur when you publish a model project in SAS Model Manager.

1. SAS Model Manager generates the score.sas and score.xml files that are required by SAS Scoring Accelerator.
2. SAS Model Manager then generates the SAS code that invokes the SAS Scoring Accelerator.
3. If you select **Validate scoring results** in the **Publish Models** window, you must specify a model train table if the field is empty. If you set the Default train table project property in SAS Model Manager, the field is populated. SAS Model Manager randomly selects the number of observations from the train table that you specified in the **More Options** window to produce a sample SAS data set.
4. SAS Model Manager runs the model's score code against the sample data set you produced in Step 3 to produce the resulting SAS data set.
5. SAS Model Manager copies the sample data set produced in Step 3 to create a table in the corresponding database or the Hadoop environment using the SAS/ACCESS interface.
6. SAS Model Manager generates the corresponding SQL code for the target database. The SQL code uses either the newly published database scoring functions or the published model scoring files through the SAS Embedded Process to produce a new table. The new table is based on the table produced in Step 5. SAS Model Manager submits the SQL statement using the SQL procedure. For Hadoop, SAS Model Manager submits a corresponding SAS macro to run the published model in Hadoop.
7. SAS Model Manager moves the resulting table produced in Step 6 back to the SAS environment using SAS/ACCESS.
8. Using PROC COMPARE, SAS Model Manager compares the two SAS data sets that resulted from Steps 4 and 7. The COMPARE procedure determines whether the published model has the same behavior as the original model in the SAS environment.
9. SAS Model Manager deletes the table produced in Step 6 from the target database or Hadoop.

Display 1 shows the **More Options** window provided by SAS Model Manager.



Display 1. SAS Model Manager Publish Options

- The **Keep scoring files if validation fails** option is applicable only if you select **Validate scoring results** in the **Publish Models** window. If you select this option, SAS Model Manager does not remove published scoring functions or model files from the target database or Hadoop even if the validation fails.
- **Sample size** is the number of observations that SAS Model Manager uses to randomly select from the train table. The default is 100.
- The **Display detailed log messages** option offers two functionalities. The first is to enable the MPRINT SAS option for generated SAS code. The second is to preserve temporary files that are produced by both SAS Model Manager and SAS Scoring Accelerator for the publish operation.

  If you have a value for the **Directory for temporary scoring files** setting in SAS® Management Console, the temporary files are placed in the specified directory. Otherwise, all temporary files are placed in the **`SASUSER`** directory.

  You can find the location of the **`SASUSER`** directory by running PROC OPTIONS in SAS. If you do not select the **Display detailed log messages** option, then the temporary files produced by SAS Model Manager and SAS Scoring Accelerator during the publish operation are placed in the **`SAS WORK`** directory. These files are no longer available after the publish operation has completed.
- The **Use model input** option is applicable when you publish a model to a database using the scoring function publish method. When this option is enabled, the champion or a challenger model's input variables are used to generate the score.xml file. Otherwise, SAS Model Manager uses the containing project's input variables to generate the score.xml file. The **Use model input** option is not applicable to Hadoop.

PUBLISHING MODELS TO A DATABASE

SAS Model Manager provides two methods for publishing a model to a database. The first is publishing a model as scoring functions. The other method is publishing a model through the SAS Embedded Process. This paper uses a sample project named SGFProject to demonstrate all the publishing functionality. The SGFProject project contains two output variables: EM_PROBABILITY and classification. It has one champion model named Regression.

Publish Models Using the Scoring Function Publish Method

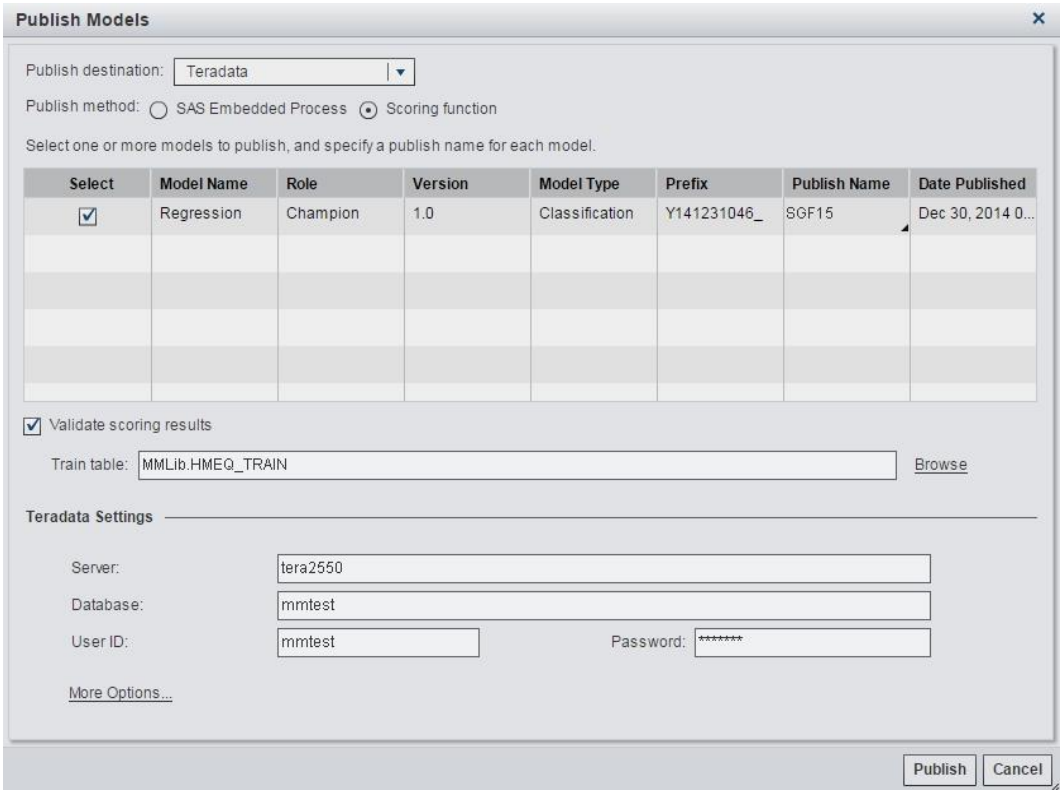
When you publish the model project SGFProject to a database as scoring functions, the publish operation produces four user-defined functions in the database. The publishing operation produces an additional user-defined function in Teradata for each additional project output variable.

Display 2 shows the settings when this project is published to a Teradata database using the scoring function publish method. You can modify the value of the **Publish Name**. This example uses `SGF15` for the publish name. SAS Model Manager appends one line of code to the end of the model's score code. It uses the following format:

```
SGF15= 'EM_PROBABILITY=' || strip(EM_PROBABILITY)|| ';'
||'classification=' || '"' || strip(classification) ||'"';
```

The variable SGF15 contains values for all of the project's output variables.

To avoid a potential overwrite in the target database, SAS Model Manager dynamically generates the prefix to make sure that it is unique. The value is in the format **`Y`***`yymmddnnn`*, where *nnn* is a counter. Whenever you publish a model to Teradata using the scoring function publish method, the main logic for the published model is captured in the Teradata user-defined function **`Y`***`yymmddnnn`***`_scorestub`**.



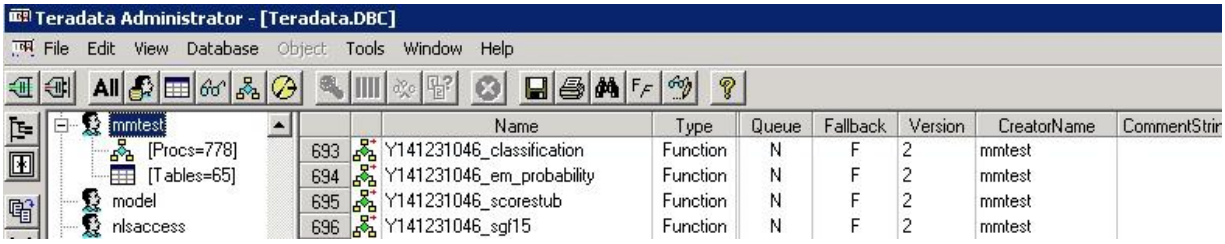
Display 2. Publish a Model Project As a Scoring Function to Teradata

The four user-defined functions are created in the mmtest Teradata database as illustrated in Display 3. Among the four user-defined functions, two of them are for the project output variables. The **`Y141231046_em_probability`** function is for the EM_PROBABILITY output variable, and the **`Y141231046_classification`** function is for classification output variable. The **`Y141231046_sgf15`** function is for the model's publish name and is used for returning values for both project output variables in the following format as a VARCHAR.

```
em_probability=0.9;classification="YES";
```

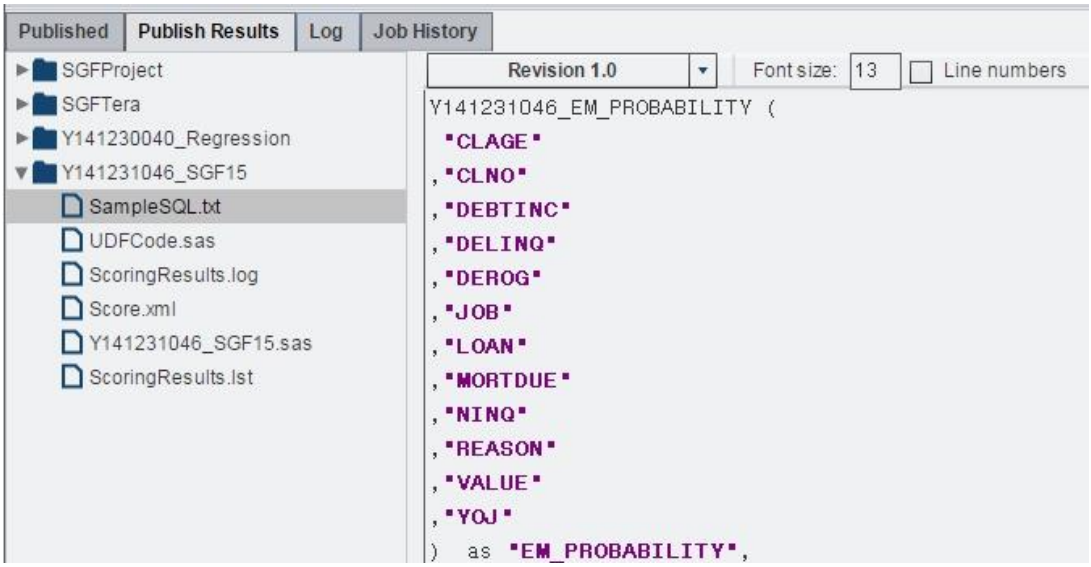
The **`Y141231046_sgf15`** function is particularly useful if you need to return both values because you need to call only one function to get the values for both project output variables.

The **`Y141231046_scorestub`** function contains the main logic of the model's score code and is used by the other three user-defined functions.



Display 3. Published Scoring Functions Displayed in Teradata Administrator

After a model project is successfully published to a database, SAS Model Manager creates a results folder to capture the resulting files, as illustrated in Display 4.



Display 4. Publish Results Folder in SAS Model Manager

There are six files in this folder. Table 1 describes the functionality of each file.

Filename	Description
SampleSQL.txt	Contains sample SQL code that you can use to invoke the newly published user-defined functions.
UDFCode.sas	Contains the SAS code that was submitted by SAS Model Manager for the publish operation.
ScoringResults.log	Contains the SAS log for the publish operation.
Score.xml	Contains the model project input variables and project output variables. It is generated by SAS Model Manager at run time and is required by SAS Scoring Accelerator.
Y141231046_SGF15.sas	Generated by SAS Model Manager at run time and contains the SAS score code (score.sas) that is required by SAS Scoring Accelerator.
ScoringResults.lst	Contains the SAS listing for this publish operation.

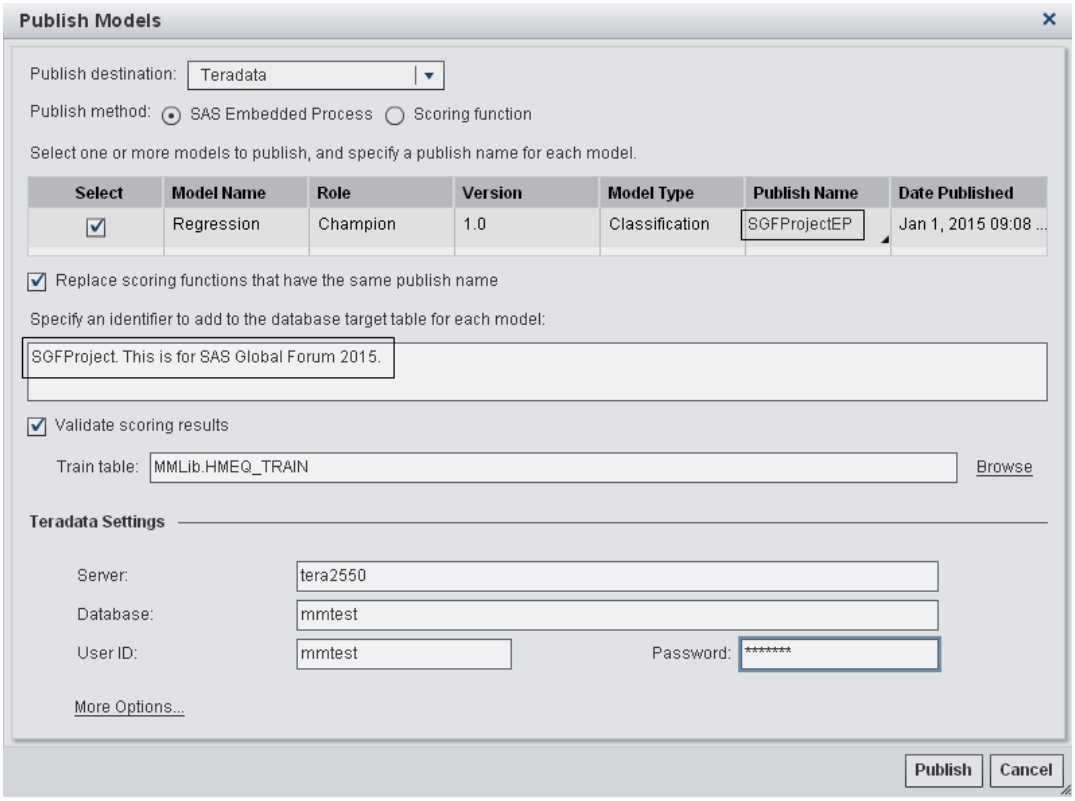
Table 1. Publish Results Files

Publishing Models Using the SAS Embedded Process Publish Method

Publishing a SAS Model Manager model project using the SAS Embedded Process publish method is a simple operation compared to using the scoring function publish method. When you publish a model project to a database using the SAS Embedded Process, the SAS Scoring Accelerator converts the

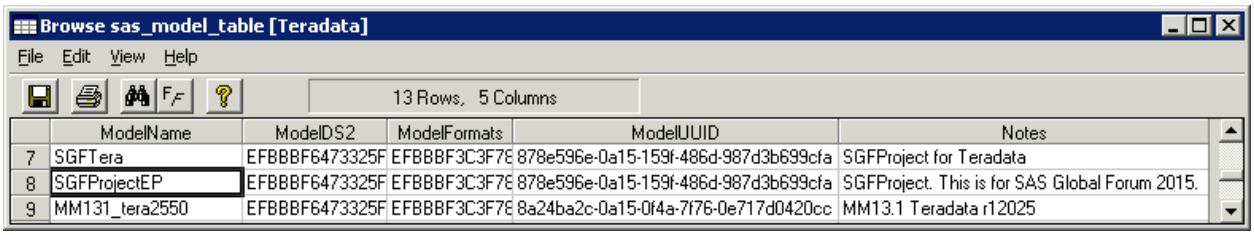
model's DATA step code to DS2 code. It then moves the DS2 code to a table (sas_model_table) in a database on a Teradata server. Display 5 shows the user interface for the operation.

You can modify the value of the publish name and also enter a value in the **Specify an identifier to add to the database target table for each model** field. The value that you enter is saved to the table sas_model_table.



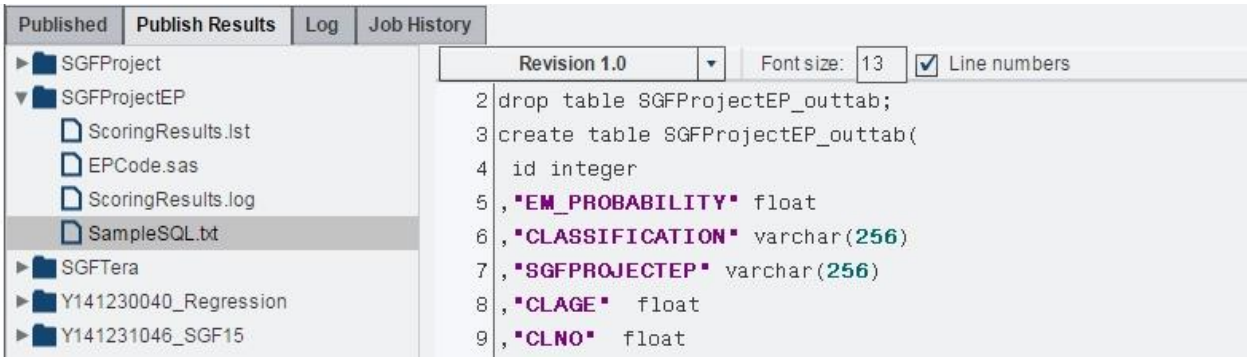
Display 5. Publish a Model Project to Teradata Using the SAS Embedded Process

Display 6 shows the sas_model_table contents for the corresponding results of the publish operation that is shown in Display 5. The ModelName column contains the value that you specified in the Publish Name column in SAS Model Manager. The ModelDS2 column contains the corresponding model's DS2 code. The ModelUUID column contains the corresponding model's universally unique identifier (UUID). The Notes column contains the text that you specified for the identifier in the text area in Display 5. The Notes column can be very helpful for users who read sas_model_table without access to SAS Model Manager.



Display 6. Sas_model_table Contents

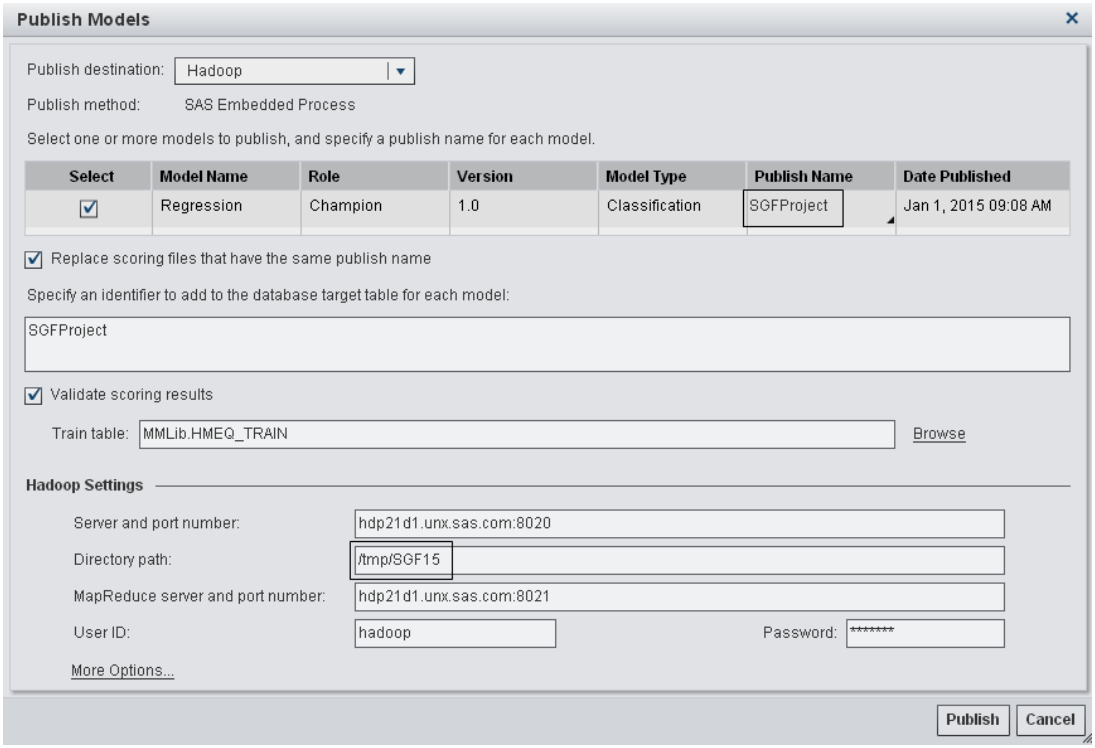
After you successfully publish a model to a database using the SAS Embedded Process, SAS Model Manager also creates a folder in the model repository in which to store the corresponding publish results files. Display 7 shows the list of these files. SampleSQL.txt contains sample SQL code that you can use in your application. **`EPCode.sas`** is SAS code submitted by SAS Model Manager for this publish operation.



Display 7. Published Results Folder in SAS Model Manager

PUBLISHING MODELS TO HADOOP

Publishing a model to Hadoop is similar to publishing a model to a database using the SAS Embedded Process. Instead of moving a model's DS2 code to a table, it moves the model's DS2 code to a Hadoop Distributed File System (HDFS) directory. Display 8 shows the SAS Model Manager publish user interface for Hadoop. You need to specify an HDFS directory for this publish operation. You can also modify the value of the Publish Name.



Display 8. Publish a Model to Hadoop

After a model is published to Hadoop, a list of files is created in an HDFS directory on a Hadoop server. The list of files includes the model's DS2 code (SGFProject.ds2), the model's DATA step code (score.sas), score.xml, and notes.txt. The notes.txt file contains both the text that you entered in the identifier text area and the corresponding model's UUID. Display 9 shows the list of files in HDFS.

Contents of directory /tmp/SGF15/ds2/SGFProject

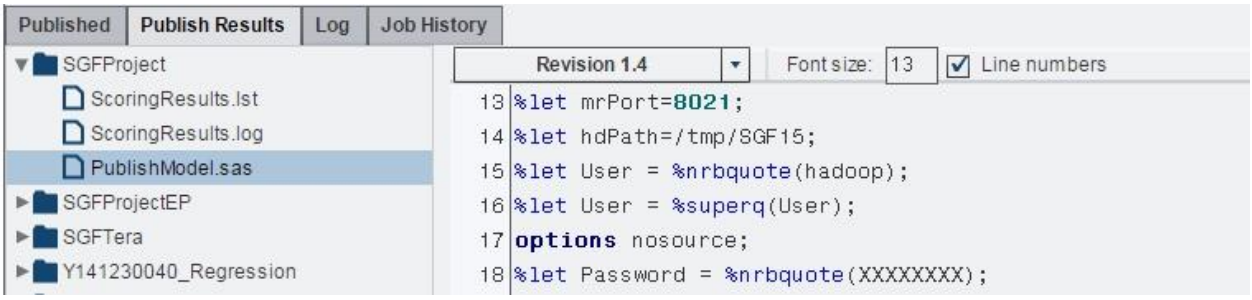
Goto : /tmp/SGF15/ds2/SGFProj go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
SGFProject.ds2	file	4.38 KB	1	128 MB	2014-12-26 14:01	rw-r--r--	hadoop	hdfs
SGFProject_ufmt.xml	file	6.24 KB	1	128 MB	2014-12-26 14:01	rw-r--r--	hadoop	hdfs
notes.txt	file	77 B	1	128 MB	2014-12-26 14:01	rw-r--r--	hadoop	hdfs
score.sas	file	7.16 KB	1	128 MB	2014-12-26 14:01	rw-r--r--	hadoop	hdfs
score.xml	file	1.88 KB	1	128 MB	2014-12-26 14:01	rw-r--r--	hadoop	hdfs

Display 9. List of Files in HDFS

A results folder is also created in SAS Model Manager to capture the resulting SAS files for this publish operation. Display 10 shows the results. It does not have a SampleSQL.txt file because Hadoop is not a database. A later section of this paper describes how you can score a published model in Hadoop.



Display 10. Published Results Folder in SAS Model Manager

USING A PUBLISHED MODEL FOR SCORING

After a model is published, you have various ways to execute the published model from within the database or Hadoop. You do not need to move your data from the database or Hadoop to your SAS environment. This paper uses Teradata and Hadoop to demonstrate how this can be done.

SCORING A PUBLISHED MODEL IN A DATABASE

As already described, you can publish a model to Teradata by using either a scoring function publish method or the SAS Embedded Process publish method. The two methods work differently in a database. The scoring function method turns a model into a collection of user-defined functions in the Teradata database. The number of user-defined functions created for the model is determined by the containing model project's output variables.

On the other hand, the SAS Embedded Process publishing method moves the model's DS2 code into a table (sas_model_table) within the database. The SAS Embedded Process executes the model's DS2 code. The SAS Embedded Process is a stored procedure in the Teradata database.

Scoring a Published Model Using a Scoring Function

The way that a user-defined function in Teradata works is similar to the way that a native SQL function works. You can call each individual function as you would call any other native SQL function.

For example, if you want to create a table that contains an ID and probability column based on a table, you can submit the following SQL code. This example creates a table called sgf_score_udf using an input table called hmeq_score_input:

```
insert into sgf_score_udf(ID, "probability")

select smmid,
Y141231046_EM_PROBABILITY (
"CLAGE"
,"CLNO"
,"DEBTINC"
,"DELINQ"
,"DEROG"
,"JOB"
,"LOAN"
,"MORTDUE"
,"NINQ"
,"REASON"
,"VALUE"
,"YOJ"
)  as "probability"
from mmtest.hmeq_score_input;
```

You can submit the above SQL code using your favorite SQL engine. After the code is executed, sgf_score_udf contains the same number of records as the hmeq_score_input table with the ID and probability columns. The hmeq_score_input column should contain all the variables specified as arguments of the function **`Y141231046_EM_PROBABILITY`**.

Scoring a Published Model Using the SAS Embedded Process

In contrast to scoring a model using scoring functions, scoring a model using the SAS Embedded Process is quite straightforward. To execute the model's score code, you specify the following three elements and call the SAS Embedded Process:

- scoring input table
- model name
- scoring output table

This example code highlights the three elements in highlighted in gray:

```
call SAS_SYSFNLIB.SAS_SCORE_EP
(
   'INQUERY=select "smmid",
"CLAGE","CLNO","DEBTINC","DELINQ","DEROG","JOB","LOAN","MORTDUE","NINQ","RE
ASON","VALUE","YOJ"
      from "mmtest"."hmeq_score_input" ',
   'MODELTABLE="mmtest"."sas_model_table"',
   'MODELNAME=SGFProjectEP',
   'OUTTABLE="mmtest"."sgf_score_ep"',
   'OUTKEY=smmid',
   'OPTIONS='
);
```

You can submit the above SQL code using your favorite SQL engine. After you run the code, the scoring output table contains all of the variables in the scoring input table plus the model's output variables.

The major difference between these two methods is that a scoring function produces only one value, and the SAS Embedded Process scoring produces all of the model's output variables. Running the SAS Embedded Process within a database has the same effect as running a model's SAS code in a SAS environment.

SCORING A PUBLISHED MODEL IN HADOOP

Scoring a published model in Hadoop is similar to scoring a published model using the SAS Embedded Process in a database because both methods use SAS Embedded Process technology. To score a published model in Hadoop, you must run the SAS macro %INDHD_RUN_MODEL in SAS. However, the scoring action still occurs in Hadoop. The SAS macro provides an easy way to invoke the SAS Embedded Process in Hadoop.

You also need to specify a score input file, a score output file, and a model's DS2 file when you score a published model in Hadoop. This is the same as when you score a published model using the SAS Embedded Process in a database. Make sure that the score input file is present in an HDFS location.

This sample code scores a published model in Hadoop:

```
%let indconn=%str(HADOOP_CFG=C:\Users\sasdemo\merged_hdp21d1.xml
USER=hadoop PASSWORD=xxxxxx);

%indhd_run_model (inmetaname=/tmp/SGF15/meta/score_in.sashdmd
   , outdatadir=/tmp/SGF15/temp/score_out10
   , outmetadir=/tmp/SGF15/meta/score_out10.sashdmd
   , scorepgm=/tmp/SGF15/ds2/SGFProject/SGFProject.ds2
   , formatfile=/tmp/SGF15/ds2/SGFProject/SGFProject_ufmt.xml
   , forceoverwrite=true
   , keep=em_classification em_eventProbability
   , trace=YES);
```

Instead of specifying a score input file HDFS location, you need to specify the HDFS full path of the input metadata file, also called the .sashdmd file. There are three ways to generate a SASHDMD file, depending on whether you have the score input file in Hive.

The File Is Not in a Hive Library

The following sample generates score_in_hmeq.sashdmd in the **/tmp/SGF15/meta** directory:

```
libname hdlib hadoop user="hadoop" pw=xxxxxx server="hdp21d1.unx.sas.com"
   HDFS_TEMPDIR="/tmp/SGF15/temp"
   HDFS_DATADIR="/tmp/SGF15/temp"
   HDFS_METADIR="/tmp/SGF15/meta"
   config='C:\sasdemo\hadoop\hdp21d1\merged_hdp21d1.xml'
   DBCREATE_TABLE_EXTERNAL=NO;

proc hdmd
   name=hdlib.score_in_hmeq
   format=delimited
   data_file='score_in';
   column customer_id char(20);
   column BAD double;
   column LOAN double;
   column MORTDUE double;
   column VALUE double;
   column REASON char(7);
   column JOB char(7);
   column YOJ  double;
   column DEROG double;
```

```
   column DELINQ double;
   column CLAGE  double;
   column NINQ double;
   column CLNO double;
   column DEBTINC double;
run;
```

The File Is in a Hive Library

The following sample generates hmeq_test.sashdmd in the **/tmp/SGF15/meta** directory:

```
libname hdlib hadoop user="hadoop" pw=xxxxxxx server="hdp21d1.unx.sas.com"
   HDFS_TEMPDIR="/tmp/SGF15/temp"
   HDFS_DATADIR="/tmp/SGF15/temp"
   HDFS_METADIR="/tmp/SGF15/meta"
   config='C:\sasdemo\hadoop\hdp21d1\merged_hdp21d1.xml'
   DBCREATE_TABLE_EXTERNAL=NO;

libname hive hadoop server="hdp21d1.unx.sas.com"
   user=hadoop
   database=hpsumm
   subprotocol=hive2;

proc hdmd
   name=hdlib.hmeq_test
   from=hive.hmeq_test;
run;
```

The File Is Created with the SAS/ACCESS Hadoop Engine

The following sample generates hmeq_train.sashdmd in the **/tmp/SGF15/meta** directory:

```
libname hdlib hadoop user="hadoop" pw=xxxxxx server="hdp21d1.unx.sas.com"
   HDFS_TEMPDIR="/tmp/SGF15/temp"
   HDFS_DATADIR="/tmp/SGF15/temp"
   HDFS_METADIR="/tmp/SGF15/meta"
   config='C:\sasdemo\hadoop\hdp21d1\merged_hdp21d1.xml'
   DBCREATE_TABLE_EXTERNAL=NO;

libname local base "C:\sasdemo\SGFPaper\sample";

data hdlib.hmeq_train;
   set local.hmeq_train;
run;
```

CONCLUSION

This paper describes how to use SAS Model Manager and SAS Scoring Accelerator to publish a model to a database and to Hadoop. It provides detailed publishing results for publishing a model to a database and to Hadoop. It also shows the differences between publishing a model using scoring functions and using SAS Embedded Process to a database. This paper provides sample code for using a published model in a database and Hadoop.

RECOMMENDED READING

- *SAS® 9.4 In-Database Products: User's Guide.* Available at http://support.sas.com/documentation/cdl/en/indbug/67366/PDF/default/indbug.pdf
- *SAS® 9.4 In-Database Products: Administrator's Guide.* Available at http://support.sas.com/documentation/cdl/en/indbag/67365/PDF/default/indbag.pdf

- *SAS® Model Manager 13.1: User's Guide.* Available at http://support.sas.com/documentation/cdl/en/mdlmgrug/67022/PDF/default/mdlmgrug.pdf
- *SAS® In-Database Processing with Teradata: An Overview of Foundation Technology.* Available at http://support.sas.com/resources/papers/teradata08.pdf
- SAS/ACCESS® 9.4 for Relational Databases Reference. Available at http://support.sas.com/documentation/cdl/en/acreldb/67589/PDF/default/acreldb.pdf
- SAS® DS2 Language Reference. Available at http://support.sas.com/documentation/cdl/en/ds2ref/67313/PDF/default/ds2ref.pdf

ACKNOWLEDGMENTS

The authors express sincere gratitude to the SAS Model Manager and SAS Scoring Accelerator developers, testers, technical support, and also to our customers.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Jifa Wei
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Email: jifa.wei@sas.com

Kristen Aponte
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Email: kristen.aponte@sas.com