



THE
POWER
TO KNOW.

SAS[®] Business Rules Manager 2.1 Administrator's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2013. *SAS® Business Rules Manager 2.1: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Business Rules Manager 2.1: Administrator's Guide

Copyright © 2013, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a) and DFAR 227.7202-4 and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

July 2014

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit support.sas.com/bookstore or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.



SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

Contents

Chapter 1 • General Overview	1
About SAS Business Rules Manager	1
Viewing Help and Documentation	2
Chapter 2 • Architecture for System Administrators	3
Architecture of the SAS Intelligence Platform	3
The SAS Intelligence Platform and SAS Business Rules Manager	4
Process for Publishing Rule Flows	6
Chapter 3 • Installation and Configuration	7
Pre-installation Tasks for SAS Business Rules Manager	8
Pre-installation: Database Installation and Preparation Tasks for Oracle	10
Installing SAS Business Rules Manager	13
Post-installation Tasks for SAS Business Rules Manager	18
Directories for Business Rules Metadata and XML Files	22
Deleting Business Rules XML Content	22
Chapter 4 • Users, Groups, Roles, and Capabilities	23
Security Administration Tasks for SAS Business Rules Manager	23
Administering SAS Identities for Users	24
Groups and Group Membership	24
Roles and Capabilities	25
Administering Group and Role Membership	28
Modifying System Permissions for SAS Business Rules Manager Users	29
Chapter 5 • Import And Export Macro Reference	31
Introduction to the Import and Export Macros	31
Dictionary	32
Index	57

Chapter 1

General Overview

About SAS Business Rules Manager	1
Viewing Help and Documentation	2

About SAS Business Rules Manager

Decision management systems can transform the way businesses make decisions. They enable businesses to use the information they already have to make better decisions—decisions that are based on predictive analytics rather than on past history. Decision management systems automate the process of making decisions, particularly day-to-day operational decisions. They improve the speed, efficiency, and accuracy of routine business processes, in part by reducing the need for human intervention.

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

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

vocabulary management

A business vocabulary identifies the objects and actors in your business domain. It defines the entities and terms that are the building blocks of business rules. SAS Business Rules Manager enables you to easily create and edit entities and terms. For individual terms, you can create a list of allowable values, which makes creating rules even easier.

business rule authoring

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

rule set management

A rule set is a logical collection of rules. A single rule set can have many rules. For example, you might have a rule set that determines a customer's asset balance and another rule set that determines a customer's debt level. SAS Business Rules

Manager displays rules sets in decision tables. Each row of a decision table defines the conditions and actions for one rule. By using SAS Business Rules Manager, you can easily create new rule sets, reorder the rules in a rule set, add new rules to existing rule sets, and more.

rule flow authoring and publishing

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

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

Viewing Help and Documentation

SAS Business Rules Manager provides the following types of Help and documentation:

how-to Help

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

embedded Help

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

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

SAS Business Rules Manager: User's Guide

This document provides detailed information about the concepts and tasks that are related to using SAS Business Rules Manager. This document is available at <http://support.sas.com/documentation/solutions/brm>.

The user ID and password for this site are available from your SAS consultant.

SAS Business Rules Manager: Administrator's Guide

This document contains information about the administration tasks that are required to set up and configure the SAS Business Rules Manager and is also available at <http://support.sas.com/documentation/solutions/brm>.

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

Chapter 2

Architecture for System Administrators

Architecture of the SAS Intelligence Platform	3
The SAS Intelligence Platform and SAS Business Rules Manager	4
Process for Publishing Rule Flows	6

Architecture of the SAS Intelligence Platform

The SAS Intelligence Platform is designed to efficiently access large amounts of data, while simultaneously providing timely intelligence to a large number of users. The Platform uses an n -tier architecture that enables you to distribute functionality across computer resources, so that each type of work is performed by the resources that are most suitable for the job.

A *tier* in the SAS architecture represents a conceptual category of software components that perform similar types of computing tasks and that require similar types of resources. Different tiers do not necessarily represent separate computers or groups of computers. More than one computer can be used for a specified tier as well.

You can modify the SAS architecture to meet the demands of your workload. For a large company, the architecture can easily consist of many computers with different operating environments. For prototyping, demonstrations, or very small enterprises, the components for all of the tiers can be installed on a single computer.

The architecture of the SAS Intelligence Platform consists of the following tiers.

Client	The client tier provides users with access to intelligence data and to functionality through web-based interfaces.
Middle	The middle tier provides an environment in which the SAS Customer Intelligence web application can execute. The middle tier applications rely on servers on the SAS server tier to perform SAS processing, including data query and analysis.

SAS Server	<p>The SAS server tier includes the SAS Metadata Server and several compute servers that execute SAS analytical and reporting processes for distributed clients.</p> <p>The compute servers include the SAS OLAP Server, the SAS Workspace Server, the SAS Pooled Workspace Server, and the SAS Stored Process Server. The SAS object spawner process runs on server machines and handles requests for SAS Workspace Servers, SAS Pooled Workspace Servers, and SAS Stored Process Servers.</p>
Data	<p>The data tier stores your enterprise data. All of your existing data assets can be used, whether your data is stored in relational database management systems, SAS tables, or enterprise resource planning system (ERP) tables.</p>

For more information about the SAS Intelligence Platform, see *SAS Intelligence Platform: Overview* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

The SAS Intelligence Platform and SAS Business Rules Manager

The SAS Intelligence Platform architecture is a comprehensive, end-to-end infrastructure for creating, managing, and distributing enterprise intelligence. This architecture consists of the following tiers:

client tier

provides users with desktop access to data and functionality through easy-to-use interfaces. With SAS Business Rules Manager, users author rule sets and rule flows through the SAS Business Rules Manager Flex client.

middle tier

provides web-based interfaces for creating reports and distributing information, while passing analysis and processing requests to the SAS servers. The middle tier provides SAS Business Rules Manager services.

server tier

provides SAS servers that process data and handle client requests. For SAS Business Rules Manager, the server tier provides the SAS Business Rules Engine.

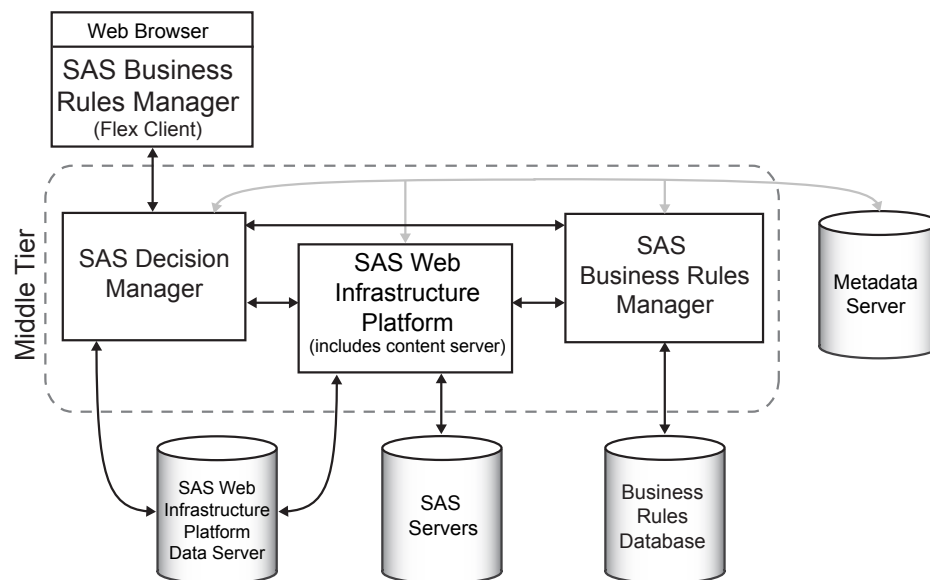
data tier

stores your data. The business rules database contains all of the data that the user enters through the SAS Business Rules Manager Flex client application.

Note: SAS Business Rules Manager does not support middle-tier clustering. As a result, SAS Business Rules Manager runs on the master node in the cluster. If the master node is not available, SAS Business Rules Manager is not available (even if other SAS applications are available on other nodes in the cluster).

The following figure shows how SAS Business Rules Manager is deployed on the SAS Intelligence Platform.

Figure 2.1 SAS Intelligence Platform Architecture and SAS Business Rules Manager



SAS Business Rules Manager

the Flex client application for SAS Business Rules Manager. Through this client application, users author vocabularies, rule sets, and rule flows.

SAS Decision Manager

manages requests to SAS Business Rules Manager and provides general SAS Decision Manager functions such as the administration and setup functions.

SAS Web Infrastructure Platform

provides common SAS infrastructure services that SAS Business Rules Manager uses to authenticate users and to access services within the SAS platform.

SAS Business Rules Manager on the middle tier

manages communication with the business rules database and initiates the process of saving rule flows to the content repository.

SAS Web Infrastructure Platform Data Server

serves as transactional storage for SAS middle-tier software and some SAS solutions software.

SAS Servers

SAS application servers that execute SAS code that is submitted from the middle-tier applications.

Business Rules database

contains all of the data that users enter through the SAS Business Rules Manager client.

metadata server

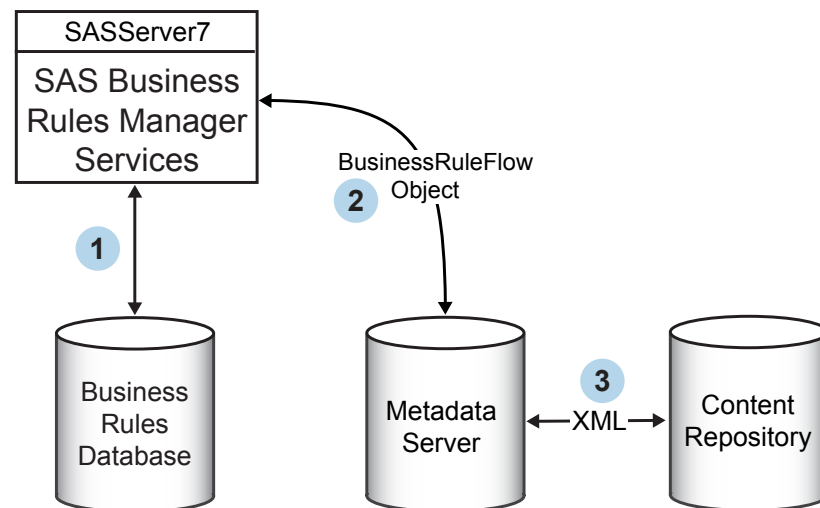
contains the BusinessRuleFlow public metadata objects that are created when a rule flow is published. These objects are used by the integrated SAS applications to execute rule flows.

Process for Publishing Rule Flows

When a user publishes a rule flow, SAS Business Rules Manager creates an XML file and a BusinessRuleFlow metadata object. The XML file is stored in the content repository, and the metadata object is stored on the metadata server.

The following figure illustrates the process of publishing rule flows.

Figure 2.2 Process for Publishing Rule Flows



- 1 SAS Business Rules Manager reads the rule flow data in the business rules database.
- 2 SAS Business Rules Manager generates the BusinessRuleFlow public metadata object and the XML file. SAS Business Rules Manager stores the metadata object on the metadata server.
- 3 The metadata object stores the XML file in the content repository.

Chapter 3

Installation and Configuration

Pre-installation Tasks for SAS Business Rules Manager	8
About the Pre-installation Tasks	8
Verify Operating System Requirements	8
Determine the Database to Use	8
Determine Time Zone Requirements	8
Install the Prerequisite Software	9
Create Standard User Accounts	9
Obtain a Deployment Plan and a SAS Installation Data File	9
Download Your Software and Create a SAS Software Depot	10
Pre-installation: Database Installation and Preparation Tasks for Oracle	10
About the Database Pre-installation Tasks	10
Install the Oracle Database Server	10
Install JDBC Drivers	10
Install a Database Client Application	11
Determine Required Database Information	11
Specify the Required Database Privileges	12
Test the Connection to Your Database	12
Installing SAS Business Rules Manager	13
About the SAS Deployment Wizard	13
Migrating from Previous Versions of SAS to SAS 9.4	13
Single-Machine versus Multiple-Machine Installations	14
Products Installed with SAS Business Rules Manager	14
Running the SAS Deployment Wizard	14
Post-installation Tasks for SAS Business Rules Manager	18
Overview of Post-installation Tasks	18
(Optional) Create Oracle Database Synonyms	18
Follow Instructions in Instructions.html	19
(Optional) Create Database Tables	19
Set Up the UUID Generator Daemon	19
Modify Log File Settings	19
Create a Business Rules Manager Administrator Group and User	21
Create a Default BRM Business Context	21
Directories for Business Rules Metadata and XML Files	22
Deleting Business Rules XML Content	22

Pre-installation Tasks for SAS Business Rules Manager

About the Pre-installation Tasks

Before you begin to install SAS Business Rules Manager, be sure to review the Pre-Installation Checklist that is provided with your deployment plan. This checklist provides a detailed list of the pre-installation requirements. It also enables you to record important information that you need when you are installing the software.

Perform the following pre-installation tasks before you install SAS Business Rules Manager:

1. Verify that your system meets the minimum requirements. (See page 8.)
2. Determine the database that you want to use for your business rules database. (See page 8.)
3. Determine whether you need to synchronize the time zones specified in all of your operating environments. (See page 8.)
4. Install the prerequisite software. (See page 9.)
5. Create user accounts in the operating system. (See page 9.)
6. Obtain a deployment plan and installation data file. (See page 9.)
7. Download your software and create a software depot. (See page 10.)

The following topics provide details about each step.

Verify Operating System Requirements

Ensure that you meet the minimum requirements that are described in *System Requirements--SAS Business Rules Manager* at <http://support.sas.com/documentation/installcenter/en/ikbrlstofrsr/66624/HTML/default/index.html>. System requirements are unique for each operating system. They include software requirements, hardware requirements, space requirements, specific product requirements, and graphics hardware and software compatibility.

Determine the Database to Use

Note: Support for the SAS Web Infrastructure Platform Data Server as a business rules database is experimental.

You can use either Oracle or the SAS Web Infrastructure Platform Data Server for your business rules database. The SAS Web Infrastructure Platform Data Server is based on PostgreSQL 9.1.9. For more information, see “SAS Web Infrastructure Platform Data Server” in Chapter 2 of *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Determine Time Zone Requirements

If you will be deploying rule flows using SAS Real-Time Decision Manager, and those rule flows use terms of type Date or Datetime, then all of your operating environments

(middle tier, server tier, SAS Federation Server, and client) must be set to the same time zone. If the time zones do not match across all environments, it is recommended that you do not use Date or Datetime data types.

Install the Prerequisite Software

Before you install SAS Business Rules Manager, install the following prerequisite software:

- Oracle, if you are using Oracle for your business rules database. See [“Pre-installation: Database Installation and Preparation Tasks for Oracle”](#) on page 10 for instructions.
- Adobe Flash Player version 10.1.0 or later. This software is required on each client machine that accesses SAS Business Rules Manager.

Create Standard User Accounts

As a pre-installation task, you must create the following user accounts in the operating system:

- an account for the user who will install and configure the SAS software.
- an account to run the spawned SAS servers.
- an account for scheduling jobs. (Specify the user ID for this account at the **SAS Decision Manager: Scheduling User ID** prompt.)

You should also create a SAS Server Users group on Windows and a sas group on UNIX.

For important details about setting up these users and groups, see the pre-installation checklist for your deployment. Also see Chapter 2, “Setting Up Users, Groups, and Ports,” in *SAS Intelligence Platform: Installation and Configuration Guide*.

Obtain a Deployment Plan and a SAS Installation Data File

Before you can install your SAS software, you must obtain a deployment plan. The deployment plan is an XML file that specifies the software that you will install and configure on each machine in your environment. The plan serves as input to the SAS Deployment Wizard. A deployment plan can be a custom plan for your specific software installation, or it can be a standard, predefined plan that describes a common configuration. For more information, see “About Deployment Plans” in Chapter 6 of *SAS Intelligence Platform: Installation and Configuration Guide*. See also [“Migrating from Previous Versions of SAS to SAS 9.4”](#) on page 13.

You must also obtain a SAS Installation Data (SID) file. The SID file contains license (SETINIT) information that is required to install SAS.

Note: When you create a deployment plan that includes SAS Business Rules Manager, you will see SAS Decision Manager components in your plan even if you have not licensed SAS Decision Manager or included the SAS Decision Manager offering in your plan.

Download Your Software and Create a SAS Software Depot

Use the SAS Download Manager to download the software that is listed in your SAS Software Order. The SAS Download Manager creates a SAS Software Depot from which you install your software. For more information, see Chapter 3, “Creating a SAS Software Depot,” in *SAS Intelligence Platform: Installation and Configuration Guide*. You can then use the SAS Deployment Wizard to install your software.

Pre-installation: Database Installation and Preparation Tasks for Oracle

About the Database Pre-installation Tasks

If you are using Oracle for your business rules database, perform the following steps before you install SAS Business Rules Manager:

1. [Install the Oracle database server. \(See page 10.\)](#)
2. [Install the JDBC drivers. \(See page 10.\)](#)
3. [Install a database client application. \(See page 11.\)](#)
4. [Determine the required database information. \(See page 11.\)](#)
5. [Specify the required database privileges. \(See page 12.\)](#)
6. [Test the connection to your database. \(See page 12.\)](#)

The following topics provide details about each step.

Install the Oracle Database Server

If you are using Oracle for your business rules database, then you must install an Oracle database server. You must install this third-party software before you install SAS Business Rules Manager.

For more information, see *System Requirements--SAS Business Rules Manager* at <http://support.sas.com/documentation/installcenter/en/ikbrlstofrsr/66624/HTML/default/index.html>.

Install JDBC Drivers

You must download the following JDBC drivers and place them in a separate directory without any other files to ensure proper installation and configuration of SAS Business Rules Manager.

- Oracle Database 10g: SAS Business Rules Manager uses the ojdbc6.jar file. You can download a copy of the Oracle driver from http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html. Select the latest Oracle Database 10g 10.2.x driver. The JDBC driver version must match the database version.
- Oracle Database 11g: SAS Business Rules Manager uses the ojdbc6.jar file. You can download a copy of the Oracle driver from <http://www.oracle.com/>

technology/software/tech/java/sqlj_jdbc/index.html. Select the latest Oracle 11.2x driver. The JDBC driver version must match the database version.

For more information about supported database drivers, see *System Requirements--SAS Business Rules Manager* at <http://support.sas.com/documentation/installcenter/en/ikbrlstofrsr/66624/HTML/default/index.html>.

Install a Database Client Application

As part of the post-installation process, SAS Business Rules Manager requires that you run a database script to prepare and initialize your database. The `Instructions.html` file (created by the SAS Deployment Wizard) provides more information about this script. To run the script, you must have installed a database client application, and you must have made the application available by setting the `PATH` environment variable.

Determine Required Database Information

During the installation and configuration of SAS Business Rules Manager, the SAS Deployment Wizard requires information about the database that SAS Business Rules Manager uses. The following table provides a list of information that you need in order to complete the steps in the SAS Deployment Wizard.

Table 3.1 SAS Deployment Wizard Information

Property	Description
Database Type	Specifies the database vendor to use with SAS Business Rules Manager. SAS Business Rules Manager supports only Oracle.
User Name	Specifies the user name for the database that is used with your SAS Business Rules Manager installation.
Password	Specifies a valid password for the user name associated with the database account.
Port	Specifies the port that is used by the database. The default port for Oracle is 1521.
Host Name	Specifies the host name of the machine where the database is installed.

Property	Description
Database Name	<p>Specifies the database name. For Oracle databases, the Net Service Name and the Service Name fields that are configured in the <code>tnsnames.ora</code> file must be the same. You must use this value for the Database Name field in the SAS Deployment Wizard. For example, if you had the following entry in the <code>tnsnames.ora</code> file, you would enter <code>monitordb</code> in the Database Name field in the SAS Deployment Wizard:</p> <pre>monitordb = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (COMMUNITY = TCP_COMM) (PROTOCOL = TCP) (HOST = hostname.your.company.com) (PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = monitordb)))</pre> <p>The Net Service Name and Service Name in this example are the same.</p>
DBMS JAR File	<p>Specifies the location of the database vendor's JDBC JAR file. This file must be available on the middle tier and on any machine on which you are deploying SAS Business Rules Manager in order for configuration to take place.</p> <p>For more information about the JDBC drivers that are supported by SAS Business Rules Manager, see <i>System Requirements--SAS Business Rules Manager</i> at http://support.sas.com/documentation/installcenter/en/ikbrlstofrsr/66624/HTML/default/index.html.</p>

Specify the Required Database Privileges

Ensure that the users of your rules database have the required database privileges. The required privileges for Oracle databases are:

- CONNECT
- CREATE SESSION
- RESOURCE
- CREATE TABLE
- CREATE VIEW
- CREATE SEQUENCE
- CREATE TRIGGER

Test the Connection to Your Database

Execute a command from the terminal to verify that your database is set up. For example, to use an Oracle database, you can execute the following command using SQL*Plus:

```
sqlplus USER/PASSWORD@ORACLE_SID
```

You must be able to execute this command from any directory. If you are able to execute a database command such as this only from the database installation directory, then verify that the PATH variable is set up correctly. The database client application must be installed and available on the PATH.

Installing SAS Business Rules Manager

About the SAS Deployment Wizard

You use the SAS Deployment Wizard to install and configure the SAS software and related products that are included in your deployment plan file. When you execute the SAS Deployment Wizard, you select the deployment type that you are performing. You can install and configure the software in a single execution of the wizard, or you can install and configure the software in two separate executions. The latter approach gives you the opportunity to test the SAS license before the configuration step.

The SAS Deployment Wizard prompts you to perform a variety of tasks, including the following items:

- specify the software order, the deployment plan, and the SAS software products that you are installing and configuring
- specify the directory paths for third-party products that you have installed, such as web application server software and the Java Development Kit
- specify host machine information
- specify information about user accounts that were created in the pre-installation phase
- for multiple-machine configurations, install the server-tier, middle-tier, and client-tier software on the appropriate machines

For more information, see Chapter 6, “Installing and Configuring Your SAS Software,” in *SAS Intelligence Platform: Installation and Configuration Guide*.

Note: The SAS Deployment Wizard prompts you for SAS Decision Manager settings even if you have not licensed SAS Decision Manager or included the SAS Decision Manager offering in your plan.

Migrating from Previous Versions of SAS to SAS 9.4

When you are migrating from a previous version of SAS to SAS 9.4, you cannot use the migration provided by the SAS Deployment Wizard and deploy SAS Business Rules Manager at the same time. To perform the migration available in the SAS Deployment Wizard, create a deployment plan without including SAS Business Rules Manager in the offerings, run the SAS Deployment Wizard, and refer to that plan from the SAS Deployment Wizard. After your migration is complete, edit the deployment plan, add the SAS Business Rules Manager offering, and run the SAS Deployment Wizard again, but do not select **Migration**.

Single-Machine versus Multiple-Machine Installations

You can install SAS Business Rules Manager on one or on several machines. This choice is determined when you order SAS Business Rules Manager and is detailed in the deployment plan XML file.

For multiple-machine installations, you must first install SAS Business Rules Manager on the server-tier machine. You can then install SAS Business Rules Manager on other additional machines that are part of a middle tier in your configuration. For guidelines on installing SAS on multiple machines, see “Installation Order Rules for Multiple Machine Deployments” in Chapter 6 of *SAS Intelligence Platform: Installation and Configuration Guide*.

The server tier consists of a set of SAS servers that are installed as a part of the SAS Intelligence Platform. The server tier contains the code generation macro that is necessary for executing rules and integrating SAS Business Rules Manager with other SAS products.

Products Installed with SAS Business Rules Manager

Your deployment plan for SAS Business Rules Manager includes additional SAS products that support and complement SAS Business Rules Manager functionality. The SAS Deployment Wizard prompts you to install and configure each of the following products:

- Base SAS
- SAS/ACCESS Interface to Oracle or SAS/ACCESS Interface to PostgreSQL
- SAS Decision Manager Common Components
- SAS Integration Technologies
- SAS Management Console
- SAS Metadata Server
- SAS Visual Analytics Report Services (Optional)

Running the SAS Deployment Wizard

About Running the SAS Deployment Wizard

To run the SAS Deployment Wizard, follow the instructions in “Install and Configure SAS Interactively” in Chapter 6 of *SAS Intelligence Platform: Installation and Configuration Guide*.

Note: You can run the wizard on operating systems that do not use a windowing environment. For more information, see *SAS Deployment Wizard and SAS Deployment Manager: User's Guide* at <http://support.sas.com/documentation/installcenter/en/ikdeploywizug/66034/PDF/default/user.pdf>.

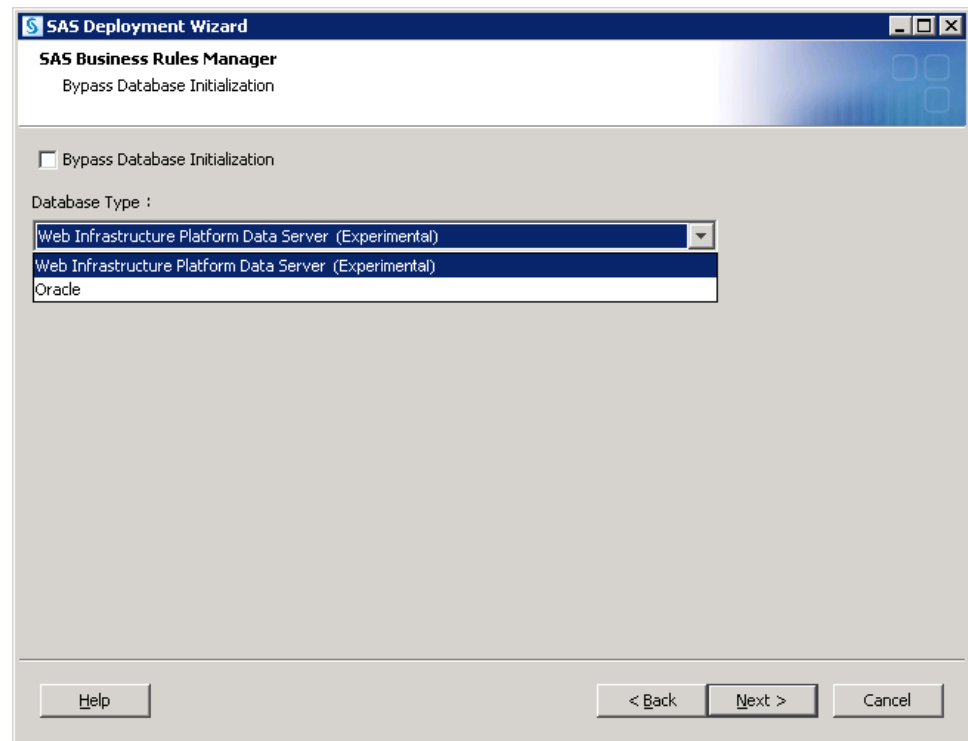
The type and number of configuration-related pages that you see depend on the prompt level that you choose, the SAS tier that you are currently deploying, and the contents of your SAS 9.4 custom order. The following topics provide information for prompts that are specific to SAS Business Rules Manager. For additional information about any of the SAS Deployment Wizard prompts, see the online Help for the wizard page in question.

Initialize the Database

In the database initialization dialog box, select **Bypass Database Initialization** if you have already created the data and tables within the database for SAS Business Rules Manager. If your database is empty, do not select this check box. SAS Business Rules Manager creates the tables, sequences, and constraints that are specified in the database schema.

Select either **Web Infrastructure Platform Data Server** or **Oracle** for your rules database type.

Note: The SAS Web Infrastructure Platform Data Server is experimental.



Specify Oracle Connection Options

Specify the connection options for the Oracle database. You see this dialog box only if you are using Oracle for the rules database. See also [“Determine Required Database Information” on page 11](#).

SAS Deployment Wizard
SAS Business Rules Manager
 Specify the transactional database connection options.

Database Host Name:

Select JDBC driver file for Oracle database:

Database Name:

Service Name (SID):

Connection Port:

Database Host Name

the fully qualified name of the host machine on which the Oracle database is installed.

Select JDBC driver file for Oracle database

the location of the directory that contains the database vendor's JDBC JAR files. You must have this file available on the middle tier (for multi-tier deployments) for configuration to take place.

Database Name

the database name. You can refer to the Oracle SID located in the tnsnames.ora file.

Service Name (SID)

the service name should be the same as the database name.

Connection Port

the port on which the database is listening.

Specify Database User ID and Password

Specify the user ID and password that will be used to connect to the rules database. If you are using Oracle for the rules database, you must also specify the schema. (The following dialog box is from an Oracle installation.)

SAS Deployment Wizard
SAS Business Rules Manager
Specify the Connection Options.

Schema:

User ID:

Password:

Confirm Password:

Schema

the name of the schema that was created for SAS Business Rules Manager data. You need to enter the schema only if you are using Oracle.

User ID

the ID of the database user whose credentials will be used to access SAS Business Rules Manager data on the server. For Oracle databases, the user ID is often the same as the schema name.

Password

the password of the database user whose credentials will be used to access SAS Business Rules Manager data on the server.

Select a SAS Web Infrastructure Platform Instance

If you are installing SAS Business Rules Manager as an add-on product and have already installed a SAS Web Infrastructure Platform, the SAS Deployment Wizard asks you to select a platform instance. Select **Web Infrastructure Platform 9.4** for the platform instance.

Select the SAS Application Server

If you are installing SAS Business Rules Manager as an add-on product and have already defined other SAS application servers, the SAS Deployment Wizard asks you to select which application server you want to use. Select an application server other than SASMETA.

Post-installation Tasks for SAS Business Rules Manager

Overview of Post-installation Tasks

After you install SAS Business Rules Manager, complete the following post-installation tasks:

1. (Optional) Run a database script to create synonyms in the database. (See page 18.)
2. Follow the instructions in the `Instructions.html` file. (See page 19.)
3. (Optional) Create database tables. (See page 19.)
4. Set up the UUID generator daemon. (See page 19.)
5. (Optional) Modify log file settings. (See page 19.)
6. Create the Business Rules Manager Administrator group and an administrator user ID. (See page 21.)
7. Create a default business context. (See page 21.)

The following topics provide details about each task.

(Optional) Create Oracle Database Synonyms

If you do not want to use the default schema for the rules database, there are two SQL scripts that you can run to create synonyms for the SAS Business Rules Manager rules database tables. These scripts are in `!SAS_CONFIG\Applications\SASBusinessRulesManager\2.1\ddl\Oracle\optional`. Use your preferred Oracle tool to run these scripts. One of these scripts, `brm_oracle_grant_priv_synonym.sql`, uses substitution variables. If your Oracle tool does not support substitution variables, then you need to manually replace the variable with its value, as described in Step 2.

To run these scripts, follow these steps:

1. In the script named `brm_oracle_grant_priv_synonym.sql`, find the following line:


```
def usernm='YOUR_USER_NAME';
```

Replace `YOUR_USER_NAME` with the user ID that you are using to access the SAS Business Rules Manager database.

If your Oracle tool supports substitution variables, skip to Step 3. If not, continue with Step 2.
2. If your Oracle tool does not support substitution variables, in the script named `brm_oracle_grant_priv_synonym.sql`, replace all occurrences of `&usern` with the user ID that you are using to access the SAS Business Rules Manager database.
3. Run the script named `brm_oracle_create_synonym.sql`, using your preferred Oracle tool. This script does not require substitution variables.

4. Run `brm_oracle_grant_priv_synonym.sql`, using your preferred Oracle tool.

Follow Instructions in Instructions.html

At the end of the installation process for SAS Business Rules Manager, the SAS Deployment Wizard produces an HTML document named `Instructions.html`. If your server tier and middle tier are hosted on separate machines, there is an `Instructions.html` file for each machine. Follow the instructions that are provided in the HTML documents.

(Optional) Create Database Tables

If you selected **Bypass Database Initialization** but do not have an existing database instance, run the following SQL scripts if they have not already been run. Run these scripts on your business rules database with a compatible database client for your installation. These scripts are located in `SAS-config-dir/Lev1/Applications/SASBusinessRulesManager/2.1/ddl/Oracle`.

1. `brm_oracle_create_table.sql`
2. `brm_oracle_create_constraint.sql`
3. `brm_oracle_create_sequence.sql`
4. `brm_oracle_create_view.sql`
5. `brm_oracle_required_inserts.sql`
6. `edm_oracle_create_table.sql`
7. `edm_oracle_create_sequence.sql`
8. `edm_oracle_create_constraint.sql`
9. `edm_required_inserts.sql`

Set Up the UUID Generator Daemon

The SAS Business Rules Engine uses the `UUIDGEN` function to create unique identifiers for rule-fired records. If you are executing rules in an operating environment other than Windows, you need to set up the object spawner to be a UUID Generator Daemon (`UUIDGEND`). For information, see “Universal Unique Identifiers and the Object Spawner” in Chapter 39 of *SAS Language Reference: Concepts*.

In addition, you should specify the `UUIDGENHOST` system option for any jobs that run code that was generated by the SAS Business Rules Engine. For more information, see “`UUIDGENHOST`= System Option” in *SAS System Options: Reference*.

Modify Log File Settings

Log4j Configuration File

SAS Business Rules Manager uses `log4j` to perform logging. As SAS Business Rules Manager begins to run, the `log4j` configuration file for the web application is read from `SAS-config-dir\Web\Common\LogConfig\SASBusinessRulesManagerWeb-log4j.xml`. This file is a standard `log4j` configuration file.

You should not change the existing categories or root logger in the configuration file unless you are instructed to do so by SAS Technical Support.

Logging Priority Levels

You can change the logging priority levels in the log configuration file if needed.

Table 3.2 Logging Priority Levels

Priority	Description
DEBUG	The most verbose logging level. This level displays information that is most useful for debugging an application. SAS Business Rules Manager should run under this priority only for capturing additional log information. This priority level is not an acceptable priority level for the day-to-day operation of SAS Business Rules Manager.
INFO	Verbose logging level. This level displays messages that highlight the progress of an application. SAS Business Rules Manager should run under this priority only for capturing additional log information. This priority level is not an acceptable priority level for the day-to-day operation of SAS Business Rules Manager.
WARN	Restrictive logging. This level displays information about potentially harmful situations and is an acceptable priority for the day-to-day operation of SAS Business Rules Manager.
ERROR	The most restrictive logging level. This level displays error events and is an acceptable priority for the day-to-day operation of SAS Business Rules Manager.

Log Files

By default, SAS Business Rules Manager writes the log files to ***SAS-config-dir*** \Web\Logs\SASServer7_1\SASBusinessRulesManagerWeb2.1.1.log. You can change the location of this log file in the configuration file. Changes to the configuration file take effect when the middle-tier application server is restarted. See “Administering Logging for SAS Web Applications” in Chapter 8 of *SAS Intelligence Platform: Middle-Tier Administration Guide* for more information about this configuration file.

SAS Business Rules Manager creates a new log file each day. For information about logging configurations, see “Modifying Your Server Logging Configurations” in Chapter 10 of *SAS Intelligence Platform: System Administration Guide*.

Turning on SQL Logging

To turn on SQL logging and write SQL parameter values for each query to the log file, add the following categories to the log4j.xml configuration file:

```
<category additivity="false" name="org.hibernate.type">
  <priority value="TRACE"/>
  <appender-ref ref="SAS_FILE"/>
</category>

<category additivity="false" name="org.hibernate.SQL">
  <priority value="DEBUG"/>
  <appender-ref ref="SAS_FILE"/>
```

</category>

Create a Business Rules Manager Administrator Group and User

The following instructions are high-level instructions for the tasks that you need to do for SAS Business Rules Manager in SAS Management Console. For details about using SAS Management Console to modify users and groups, see *SAS Management Console: Guide to Users and Permissions* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

To create an administrator group and user in SAS Management Console:

1. Select the **User Manager** plug-in.
2. Add the Decision Manager: Usage role to the Business Rules Manager Users group.


Note: The Business Rules Manager Users group is created when you install SAS Business Rules Manager. See “[Predefined Groups in SAS Business Rules Manager](#)” on page 25 for more information.

3. Create a new group named Business Rules Manager Administrators, and make this new group a member of the Business Rules Manager Users group.
4. Add the Decision Manager Common: Administrator role to the Business Rules Manager Administrators group.
5. Create a new user, and add the new user to the Business Rules Manager Administrators group. Use this user ID to log on to SAS Business Rules Manager and create a business context. See “[Create a Default BRM Business Context](#)” on page 21.

Create a Default BRM Business Context

For SAS Business Rules Manager, a business context is an additional level of authentication that controls access to SAS Business Rules Manager. It is recommended that you create one default business context for all users so that users are not required to select a business context each time they log on.

To create a default business context:

1. Log on to SAS Business Rules Manager using the administrator ID that was created using the steps described in “[Create a Business Rules Manager Administrator Group and User](#)” on page 21. The URL is `http://hostName:portNumber/SASDecisionManager`. You should see a **Setup** workspace and an **Administration** workspace.
2. Select the **Setup** workspace.
3. Click . The Business Context Properties dialog box appears.
4. Specify **BRM** as the name of the business context.
5. Specify **Standard BRM Business Context** as the description.
6. Select **Products** ⇒ **SAS Business Rules Manager** ⇒ **Business Rules Manager 2.1** as the root folder.
7. Click the **Users** tab.
8. Add the **Business Rules Manager Users** group to the **Selected** list.
9. Click **OK**.

Although it is possible to create additional business contexts, SAS Business Rules Manager does not filter data according to business contexts, so creating additional contexts does not provide any additional benefits.

Directories for Business Rules Metadata and XML Files

SAS Business Rules Manager creates two directories for business rules metadata: **Products** and **/System**. These directories are on the content server.

SAS Business Rules Manager creates a location for published XML files, **Sasdav/Products**. The BusinessRuleFlow metadata object does not delete the XML documents stored in this location in order to ensure that an audit trail is maintained. For more information, see [“Deleting Business Rules XML Content” on page 22](#).

For content that was published with an earlier release of SAS Business Rules Manager, the current release of SAS Business Rules Manager continues to use the original publish location. Do not delete directories created by earlier releases of SAS Business Rules Manager.

Deleting Business Rules XML Content

Before you delete any XML content from **Sasdav/Products**, you should do the following:

1. Back up all versions of the content. The easiest way to back up the content is to use SAS Management Console to export the BusinessRuleFlow object that refers to the content.
2. Ensure that there are not any BusinessRuleFlow objects that refer to the content.

Chapter 4

Users, Groups, Roles, and Capabilities

Security Administration Tasks for SAS Business Rules Manager	23
Administering SAS Identities for Users	24
Overview of SAS Identities	24
Creating SAS Identities	24
Groups and Group Membership	24
About Groups	24
Predefined Groups in SAS Business Rules Manager	25
Roles and Capabilities	25
About Roles and Capabilities	25
Predefined Roles for SAS Business Rules Manager	26
Viewing Roles and Capabilities in SAS Management Console	26
Administering Group and Role Membership	28
Adding a User to a Role	28
Creating New Roles	28
Modifying Roles	29
Modifying System Permissions for SAS Business Rules Manager Users	29

Security Administration Tasks for SAS Business Rules Manager

Security administration for SAS Business Rules Manager consists of the following tasks:

- administering SAS identities for your users by adding account information to the SAS Metadata Server
- administering groups of users in order to simplify the management of roles
- administering roles, which provide users with access to specific application features

The information in this document is a brief introduction to the concepts of users, SAS identities, groups, roles, and capabilities. For complete information about security administration, see *SAS Management Console: Guide to Users and Permissions* and *SAS Intelligence Platform: Security Administration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

Administering SAS Identities for Users

Overview of SAS Identities

For each SAS Business Rules Manager user, you must create an individual SAS identity on the SAS Metadata Server. The SAS identity is a copy of the ID with which the user logs on to SAS applications. Based on this identity, the system can determine who can access which application and can audit individual actions in the metadata layer. The SAS identity consists of a name and the user ID for the user's external account. This ID can be any type of account that is known to the metadata server's host, such as an LDAP account, Active Directory account, host account, or other type of account. When entering user IDs for Windows accounts, be sure to qualify the ID (for example, `WIN\myID` or `myID@mycompany.com`).

Creating SAS Identities

To create SAS identities for your users, manually enter the information for each user through the User Manager plug-in in SAS Management Console. If you have a large number of users, then you can extract user and group information from one or more enterprise identity sources. You can then use SAS bulk-load macros to create the identity metadata from the extracted information.

For more information about creating and managing identities, see *SAS Management Console: Guide to Users and Permissions*. For information about the SAS bulk-load macros, see Appendix 2, "User Import Macros," in *SAS Intelligence Platform: Security Administration Guide*. These documents are available at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

Groups and Group Membership

About Groups

A group is a set of users. Groups enable you to grant multiple users membership in a role or permissions to metadata, thus simplifying security administration. You can create as many groups as are needed in order to manage your installation.

TIP A group's membership can include other groups as well as individual users. This enables you to create a nested group structure.

Predefined Groups in SAS Business Rules Manager

Table 4.1 Predefined Groups In SAS Business Rules Manager

Group	Description
Public	This group includes everyone who can access the metadata server, either directly or through a trust relationship. If a user is able to log on to a client application but does not have an individual SAS identity, the user is assumed to be in the public group. Because this group has implicit membership, you cannot explicitly add or remove users from this group.
SAS Users	This group includes everyone who can access the metadata server, either directly or through a trust relationship. If a user is able to log on to a client application but does not have an individual SAS identity, the user is assumed to be in the public group. Because this group has implicit membership, you cannot explicitly add or remove users from this group.
SAS Administrators	This is a standard group for metadata administrators. In a standard configuration, members are granted broad access and administrative capabilities, but are not unrestricted.
Decision Manager Common Administrator	In your initial installation, this group is a member of the Decision Manager Common: Administrator role. You can add users to this group in order to enable them to manage administration resources such as business contexts, user sessions, and environment settings.
Business Rules Manager Users	This group is created during the installation process. It is associated with the necessary identity to allow users to access the business rules database during rule flow execution. It is the only group that is granted permission, by default, to publish business rules content to the SAS Content Server. Without configuration changes, membership in this group is required for business rules users who do not have administrator permission.
SAS System Services	This group enables members to export files on the Folders tab of SAS Management Console.

Roles and Capabilities

About Roles and Capabilities

A role manages the availability of application features such as menu items and plug-ins. An application feature that is under role-based management is called a capability.

Certain actions are available only to users or groups that have a particular role. Any user or group who is a member of a role has all of that role's capabilities.

Roles can contribute to one another. A role automatically includes all of the capabilities of a role that contributes to it.

Roles differ from permissions. In general, roles do not affect access to metadata or data.

Predefined Roles for SAS Business Rules Manager

Your installation includes several predefined roles for administrators and users of SAS Business Rules Manager. Depending on what software you have installed, you might have other predefined roles.

Note: The ability to access and update metadata is subject to permissions that are placed on that metadata. These roles do not affect permissions.

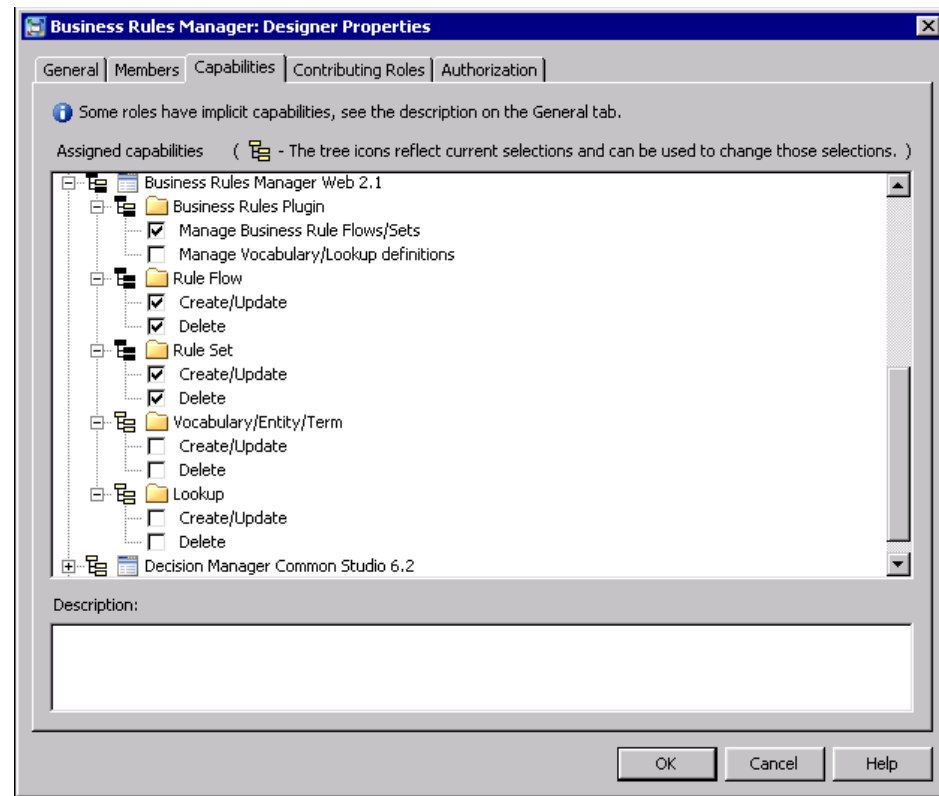
Table 4.2 Predefined Roles For SAS Business Rules Manager

Role	Capabilities
Decision Manager: Usage	This role provides the implicit capability to log on to SAS Decision Manager applications. This role contributes this capability to the other roles.
Decision Manager Common: Administrator	Users in this role can manage the categories in the Administration workspace: business contexts, user sessions, locks, and environment settings.
Business Rules Manager: All Capabilities	Users in this role can view and edit all business rules content, including vocabularies, entities, terms, lookup tables, rule sets, and rule flows.
Business Rules Manager: Designer	Users in this role can create, edit, and delete rule sets and rule flows in the Designer workspace.
Business Rules Manager: Designer Read-Only	Users in this role can view rule sets and rule flows in the Designer workspace.
Business Rules Manager: Definitions	Users in this role can create, edit, and delete vocabularies, entities, terms, and lookup tables in the Definitions workspace.
Business Rules Manager: Definitions Read-Only	Users in this role can view vocabularies, entities, terms, and lookup tables.

Viewing Roles and Capabilities in SAS Management Console

To view details about a role, open the User Manager plug-in in SAS Management Console, right-click the role, and select **Properties**. You can then view tabs that display the role's members, capabilities, and contributing roles.

For example, the following display shows the capabilities for the Business Rules Manager: Designer role. These capabilities correspond to the description of this role in [Table 4.2 on page 26](#).



Note: Some roles have implicit capabilities that are not specified on the **Capabilities** tab.

Two of the SAS Business Rules Manager capabilities control access to categories in the application:

Manage Business Rule Flows/Sets

gives users access to the Business Rules category in the Designer workspace. If the user is not assigned to a role with this capability, the Business Rules category does not appear when that user logs on to SAS Business Rules Manager.




Manage Vocabulary/Lookup definitions

gives users access to Vocabularies and Lookups in the Data category in the Definitions workspace. If the user is not assigned a role with this capability, those selections do not appear when that user logs on to SAS Business Rules Manager.

The Create/Update and Delete capabilities control access to specific object types. You can combine the category capabilities with the object capabilities to control access at whatever level is needed. For example, if you want a user to be able to view and edit rule flows only, the user should have only the following capabilities:

- Manage Business Rule Flows/Sets for the Business Rules Plugin
- Create/Update and Delete capabilities for Rule Flow objects

The following table describes the icons used in the Properties dialog box.

Icon	Description
	None of the capabilities in this category have been specified for this role.
	Some of the capabilities in this category have been specified for this role, either explicitly or through a contributing role.
	All of the capabilities in this category have been specified for this role, either explicitly or through a contributing role.

Shaded check boxes indicate capabilities that come from contributing roles.

Administering Group and Role Membership

To administer group and role membership, use the User Manager plug-in in SAS Management Console.

Adding a User to a Role

In most cases, the best way to place a user in a role is to add the user to a group that belongs to the role. You can also add users directly to roles.

To place a user in one of the predefined roles, you can add the user to one of the predefined groups. For example, to add a user to the Decision Manager Common: Administration role, add the user to the Decision Manager Common Administrator group. The user also gets the capabilities of the contributing role, Decision Manager: Usage.

Note: There is no reason to add a user directly to the Decision Manager: Usage role. This role enables a user to log on, but does not provide any other useful functionality.

For more information, see *SAS Management Console: Guide to Users and Permissions* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

Creating New Roles

The predefined roles might be sufficient for many sites. Other sites might need to make application features available to users on either a broader or more granular basis than the predefined roles allow.

You can use combinations of capabilities to create a new role. However, you can use only the capabilities that already appear in User Manager. You cannot create new capabilities.

For detailed information about roles and how to create them, see *SAS Management Console: Guide to Users and Permissions* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

Modifying Roles

The User Manager plug-in in SAS Management Console enables you to modify roles by selecting or deselecting different capabilities.

CAUTION:

No automated method can revert a role to its original set of capabilities. Instead of adjusting the capabilities of a predefined role, consider creating a new role. This advice is especially important if you need to make major changes.

If you modify a role, then follow these best practices:

- Do not rename the predefined roles. Renaming the predefined roles makes it difficult for SAS Technical Support to help you resolve problems.
- Back up the metadata server before modifying roles, and keep a record of the changes that you make.

When modifying a role, you can use only the capabilities that already appear in User Manager. You cannot create new capabilities.

For more information about roles and how to modify them, see *SAS Management Console: Guide to Users and Permissions* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

Modifying System Permissions for SAS Business Rules Manager Users

In Windows environments, you must assign users the right to **Log on as a Batch Job**. This permission is required to access functionality in the **Data** category.

Chapter 5

Import And Export Macro Reference

Introduction to the Import and Export Macros	31
Dictionary	32
%BRM_CREATE_TEMP_TERM	32
%BRM_EXPORT_FOLDER	33
%BRM_EXPORT_LOOKUP	34
%BRM_EXPORT_RULE_FLOW	35
%BRM_EXPORT_RULESET	35
%BRM_EXPORT_VOCABULARY	36
%BRM_IMPORT_FOLDER	37
%BRM_IMPORT_LOOKUP	38
%BRM_IMPORT_MARKET_BASKETS	40
%BRM_IMPORT_RULE_FLOW	47
%BRM_IMPORT_RULESET	49
%BRM_IMPORT_VOCABULARY	52
%BRM_LOAD_VOCABULARY	54

Introduction to the Import and Export Macros

SAS Business Rules Manager provides the following macros for importing data into the rules database and exporting data from the rules database. These macros must be run on the server tier.

- %BRM_CREATE_TEMP_TERM**
reads a CSV file or a SAS data set and produces a SAS data set named WORK.TERM that can be used as input to the %BRM_LOAD_VOCABULARY macro.
- %BRM_EXPORT_FOLDER**
exports definitions of business rules folders into a CSV file.
- %BRM_EXPORT_LOOKUP**
exports the contents of lookup tables into a CSV file.
- %BRM_EXPORT_RULE_FLOW**
exports rule flows from the rules database into a CSV file
- %BRM_EXPORT_RULESET**
exports rule sets from the rules database into a CSV file
- %BRM_EXPORT_VOCABULARY**
exports vocabularies from the rules database into a CSV file

%BRM_IMPORT_MARKET_BASKET
imports association rules that are generated by SAS Enterprise Miner and creates a rule flow that deploys the association rules.

%BRM_IMPORT_FOLDER
imports the folder definitions that are in the specified CSV file into the rules database.

%BRM_IMPORT_LOOKUP
imports lookup tables from the specified CSV file into the rules database.

%BRM_IMPORT_RULE_FLOW
imports rule flows from a CSV file into the rules database

%BRM_IMPORT_RULESET
imports rule sets from a CSV file into the rules database

%BRM_IMPORT_VOCABULARY
imports vocabulary terms from a CSV file into the rules database

%BRM_LOAD_VOCABULARY
loads the vocabulary terms into the WORK.TERM data set that was created by the %BRM_CREATE_TEMP_TERM macro.

Dictionary

%BRM_CREATE_TEMP_TERM

Reads a CSV file or a SAS data set and produces a SAS data set named WORK.TERM that can be used as input to the %BRM_LOAD_VOCABULARY macro.

Restriction: This macro must be run on the server tier.

Syntax

```
%BRM_CREATE_TEMP_TERM (DATAFILE=input_file<, BRM_USER=user_ID>);
```

Required Argument

DATAFILE=*input_file*

specifies either a SAS data set name or the full pathname to a CSV file. If the input file is a CSV file, the first row of the file must contain valid SAS column names, and the remaining rows must contain column values. The column values can be numeric or character data only. You cannot specify SAS informats in the column data. The column names must be unique. For example, a simple CSV file that specifies two columns, both with numeric data, might look like the following:

```
patientID,BloodPressure
1,140
2,141
3,142
```

Optional Argument

BRM_USER=*user_ID*

specifies the user ID that you want to be associated with the data that is imported. This user ID is associated with the imported objects in the rules database and is displayed in the SAS Business Rules Manager interface.

Default User ID of the user that is running the macro

Details

This macro reads a CSV file or SAS data set that defines vocabulary terms and that creates a SAS data set named WORK.TERM. You can use the WORK.TERM data set as input to the %BRM_LOAD_VOCABULARY macro. The %BRM_LOAD_VOCABULARY macro loads the vocabulary terms into the rules database. See “%BRM_LOAD_VOCABULARY” on page 54 for more information.

The %BRM_CREATE_TEMP_TERM macro derives domain types and domain values for the vocabulary terms based on the data type of the term as described in Table 5.1.

Table 5.1 Domain Types and Values for Input Terms

Term Data Type	Derived Domain Type	Derived Domain Values
String	Discrete	If there are ten or fewer distinct values in the input data, all of the values are included in the list of domain values. If there are greater than ten distinct values in the input data, individual values are not listed in the domain values.
Date	Continuous	No input values are included in the list of domain values.
Datetime	Continuous	No input values are included in the list of domain values.
Boolean	Boolean	True and False
Numeric	If there are ten or fewer distinct values in the input data, the domain type is Discrete. If there are greater than ten distinct values, the domain type is Continuous.	For Discrete domain types, all of the values in the input data are included in the list of domain values. For Continuous domain types, only the minimum and maximum values are included in the list of domain values.

%BRM_EXPORT_FOLDER

Exports definitions of business rules folders into a CSV file. You can modify the CSV file and use it as input to the %BRM_IMPORT_FOLDER macro.

Restriction: This macro must be run on the server tier.

Syntax

```
%BRM_EXPORT_FOLDER (CSV=output_filename.CSV<, FOLDER_PATH=path_name>);
```

Required Argument

CSV=output_filename

specifies the full pathname to the CSV file where you want to export the data.

Optional Argument

FOLDER_PATH=<path_name>

specifies a business rules folder that you want to export. By default, %BRM_EXPORT_FOLDER exports all lookup tables. You do not need to specify the FOLDER_PATH= option unless you want to export a specific folder.

Example FOLDER_PATH=Loans/Retail/Applications

%BRM_EXPORT_LOOKUP

Exports the contents of lookup tables into a CSV file.

Restriction: This macro must be run on the server tier.

Syntax

```
%BRM_EXPORT_LOOKUP (CSV=output_filename.CSV<, FOLDER_PATH=path-name>  
<, LOOKUP='lookup_table_1'<,'lookup_table_2'>...>);
```

Required Argument

CSV=output_filename

specifies the full pathname to the CSV file where you want to export the data.

Optional Arguments

FOLDER_PATH=path_name

specifies a business rules folder pathname in SAS Business Rules Manager that you want to filter the output by. If you specify a folder pathname, then only the objects in that path are exported.

Use a forward slash to separate folder names.

Example FOLDER_PATH=Loans/Retail/Applications

LOOKUP=%STR('lookup_table_1'<,'lookup_table_2'>...)

specifies the lookup tables that you want to export. Specify the names of the lookup tables, enclosed in single quotation marks. Separate multiple names with commas.

By default, %BRM_EXPORT_LOOKUP exports all lookup tables. You do not need to specify the LOOKUP= option unless you want to export specific tables.

Tip You can filter the lookup tables that are exported by specifying the FOLDER_PATH= option.

Example `lookup=%str('BadVINStates','StateCodes')`

%BRM_EXPORT_RULE_FLOW

Exports rule flows from the rules database into a CSV file.

Restriction: This macro must be run on the server tier.

Syntax

```
%BRM_EXPORT_RULE_FLOW (
  RULEFLOWS=ALL | %STR(rule_flow_1<,rule_flow_2>...),
  CSV=output_filename.CSV<, FOLDER_PATH=path_name>);
```

Required Arguments

CSV=output_filename

specifies the full pathname to the CSV file where you want to export the data.

RULEFLOWS=ALL | %STR(rule_flow_1<, rule_flow_2>...)

specifies the rule flows that you want to export. Specify ALL to export all rule flows. To export only selected rule flows, specify the identification numbers of the rule flows enclosed in quotation marks. Separate multiple identification numbers with commas.

Tip You can filter the rule flows that are exported by specifying the FOLDER_PATH= option.

Example `ruleflows=%str(10168,10043)`

Optional Argument

FOLDER_PATH=path_name

specifies a business rules folder pathname in SAS Business Rules Manager that you want to filter the output by. If you specify a folder pathname, then only the objects in that path are exported. For example, if you specify RULEFLOWS=ALL and FOLDER_PATH=RetailLoans, then only the rule flows in the folder RetailLoans are exported. If you specify RULEFLOWS=%STR(10045,10572) and FOLDER_PATH=RetailLoans, but neither of the specified rule flows are in the RetailLoans folder, then no rule flows are exported.

Use a forward slash to separate folder names.

Example `FOLDER_PATH=Loans/Retail/Applications`

%BRM_EXPORT_RULESET

Exports rule sets from the rules database into a CSV file.

Restriction: This macro must be run on the server tier.

Syntax

```
%BRM_EXPORT_RULESET (RULESETS=ALL | %STR(rule_set_1<, rule_set_2>...),
CSV=output_filename.CSV)<, FOLDER_PATH=path_name>);
```

Required Arguments

CSV=output_filename

specifies the full pathname to the CSV file where you want to export the data.

RULESETS=ALL | %STR(rule_set_1<, rule_set_2>...)

specifies the rule sets that you want to export. Specify ALL to export all rule sets. To export only selected rule sets, specify the identification numbers of the rule sets enclosed in quotation marks. Separate multiple identification numbers with commas.

Tip You can filter the rule sets that are exported by specifying the FOLDER_PATH= option.

Example rulesets=%str(168,43)

Optional Argument

FOLDER_PATH=path_name

specifies a business rules folder pathname in SAS Business Rules Manager that you want to filter the output by. If you specify a folder pathname, then only the objects in that path are exported. For example, if you specify RULESETS=ALL and FOLDER_PATH=RetailLoans, then only the rule sets in the folder RetailLoans are exported. If you specify RULESETS=%STR(10045,10572) and FOLDER_PATH=RetailLoans, but neither of the specified rule sets are in the RetailLoans folder, then no rule sets are exported.

Use a forward slash to separate folder names.

Example FOLDER_PATH=Loans/Retail/Applications

%BRM_EXPORT_VOCABULARY

Exports vocabularies from the rules database into a CSV file.

Restriction: This macro must be run on the server tier.

Syntax

```
%BRM_EXPORT_VOCABULARY (
VOCAB=ALL | %STR('vocabulary_1'<, 'vocabulary_2'>...),
CSV=output_filename.CSV)<, FOLDER_PATH=path_name>);
```

Required Arguments

CSV=output_filename

specifies the full pathname to the CSV file where you want to export the data.

VOCAB=ALL | %STR('vocabulary_1'<, 'vocabulary_2'>...)

specifies vocabularies that you want to export. Specify ALL to export all vocabularies. To export only selected vocabularies, specify the names of the

vocabularies enclosed in quotation marks. Separate multiple identification numbers with commas.

Tip You can filter the vocabularies that are exported by specifying the FOLDER_PATH= option.

Example `vocab=%str('LRAutoVocab','AcmeAuto')`

Optional Argument

FOLDER_PATH=*path_name*

specifies a business rules folder pathname in SAS Business Rules Manager that you want to filter the output by. If you specify a folder pathname, then only the objects in that path are exported. For example, if you specify VOCAB=ALL and FOLDER_PATH=RetailLoans, then only the vocabularies in the folder RetailLoans are exported. If you specify VOCAB=%STR('loanVocab','riskVocabulary') and FOLDER_PATH=RetailLoans, but neither of the specified vocabularies are in the RetailLoans folder, then no vocabularies are exported.

Use a forward slash to separate folder names.

Example FOLDER_PATH=Loans/Retail/Applications

%BRM_IMPORT_FOLDER

Imports the folder definitions that are in the specified CSV file into the rules database.

Restrictions: This macro must be run on the server tier.
The same user can run any of import macros at the same time. However, different users cannot run the same import macro simultaneously.

Syntax

```
%BRM_IMPORT_FOLDER (CSV=input_filename.CSV,  
REJECT=reject_filename.CSV<, BRM_USER=user_ID>);
```

Required Arguments

CSV=*input_filename*

specifies the full pathname to the CSV file where you want to import the data from.

REJECT=*reject_filename*

specifies the full pathname to the CSV file where you want the macro to write any records that were not imported to the rules database. See [“Using the %BRM_IMPORT_FOLDER Macro” on page 38](#) for more information.

Optional Argument

BRM_USER=*user_ID*

specifies the user ID that you want to be associated with the data that is imported. This user ID is associated with the imported objects in the rules database and is displayed in the SAS Business Rules Manager interface.

Default User ID of the user that is running the macro

Details

Using the %BRM_IMPORT_FOLDER Macro

The %BRM_IMPORT_FOLDER macro enables you to create new folders. You cannot update the content in existing folders with this macro. The macro uses the pathname to determine whether a folder already exists. If the pathname already exists, then the folder is rejected.

The %BRM_IMPORT_FOLDER macro runs several validation checks as it imports the folders. For example, it checks whether each folder path begins with a top-level folder and verifies that individual folder names are not longer than 100 characters. If the macro finds an invalid folder definition in the CSV file, it writes a message to the SAS log, and the folder is rejected. The macro writes the input records for the rejected folder to the CSV file that was specified in the REJECT= option.

Format of the Folder CSV Input File

Each row of the CSV input file identifies a folder. The CSV file must contain all of the columns listed in the following table, in the order listed. You must specify values for all columns, except as noted in the following table. To create a blank column in the CSV file, specify two comma separators without any content between them. For example, to import data to a folder named Applications and to specify a blank column for the folder description and default folder flag, specify the following in the CSV file:

```
Applications,,Y,,Loans/Retail
```

Table 5.2 Format of the Folder CSV Input File

Column	Description	Can Column Be Blank
FOLDER_NM	The name of the folder where you want to import the contents of the CSV file.	No
FOLDER_DESC	The description of the folder.	Yes
TOP_LEVEL_FOLDER_FLG	Specifies whether the folder is a top-level folder. Specify Y or N .	No
DEFAULT_FOLDER_FLG	Specifies whether the folder is the default folder. Specify Y or N .	Yes
FOLDER_PATH	The pathname to the business rules folder where you want to import the contents of the CSV file. This path must exist. Separate folder names with forward slashes.	No

%BRM_IMPORT_LOOKUP

Imports lookup tables from the specified CSV file into the rules database.

Restrictions: This macro must be run on the server tier.

The same user can run any of import macros at the same time. However, different users cannot run the same import macro simultaneously.

Syntax

`%BRM_IMPORT_LOOKUP (CSV=input_filename.CSV,
REJECT=reject_filename.CSV<, options>);`

Required Arguments

CSV=*input_filename*

specifies the full pathname to the CSV file where you want to import the data from.

REJECT=*reject_filename*

specifies the full pathname to the CSV file where you want the macro to write any records that were not imported to the rules database. See [“Using the %BRM_IMPORT_LOOKUP Macro” on page 39](#) for more information.

Optional Arguments

BRM_USER=*user_ID*

specifies the user ID that you want to be associated with the data that is imported. This user ID is associated with the imported objects in the rules database and is displayed in the SAS Business Rules Manager interface.

Default User ID of the user that is running the macro

BYPASSLOCK=Y|N

enables you to override the lock that another user has on the importing process. See [“Using the %BRM_IMPORT_LOOKUP Macro” on page 39](#) for more information.

Default N

Details

Using the %BRM_IMPORT_LOOKUP Macro

The %BRM_IMPORT_LOOKUP macro enables you to do the following tasks:

- add new lookup tables
- add new key-value pairs to existing lookup tables
- update (refresh) existing key-value pairs in existing lookup tables

The macro uses the lookup table name and path to determine whether a lookup table already exists. If the lookup table already exists, then it is updated. If the path exists but the lookup table does not exist, the lookup table is created. If the path does not exist, then the lookup table is rejected.

The %BRM_IMPORT_LOOKUP macro runs several validation checks as it imports the lookup tables. For example, the macro checks whether the LOOKUP_NM or NAME columns in the input file are empty or whether the LOOKUP_NM column specifies an invalid lookup name. All valid key-value pairs are imported. If the macro finds an invalid key-value pair in the CSV file, it writes a message to the SAS log, and the key-value pair is rejected. The macro writes the input records for the rejected key-value pairs to the CSV file that was specified in the REJECT= option.

When you run the %BRM_IMPORT_LOOKUP macro, it creates a lock table in the rules database named lock_import_lookup table. The SAS log states which user holds the lock and the time at which the lock started. This lock might remain in place after the macro has finished. If this happens, you can override the lock by specifying the BYPASSLOCK=Y option when you run the macro.

Format of the Lookup CSV Input File

Each row of the CSV input file identifies a key-value pair and the lookup table in which it belongs. The CSV file must contain all of the columns listed in the following table, in the order listed. You must specify values for all columns, except as noted in the table. To create a blank column in the CSV file, specify two comma separators without any content between them. For example, to import the key **AU** and the value **Australia** into the lookup table **Country_Codes** and to specify a blank column for the description, specify the following in the CSV file:

```
Country_Codes,,AU,Australia,Loans/Retail
```

Table 5.3 Format of the Lookup CSV Input File

Column	Description	Can Column Be Blank
FOLDER_PATH	The pathname to the business rules folder where you want to import the lookup table. This path must exist. Separate folder names with forward slashes.	No
LOOKUP_NM	The name of the lookup table.	No
DESCRIPTION	The description of the lookup table.	Yes
NAME	The lookup key.	No
VALUE	The lookup value.	Yes

%BRM_IMPORT_MARKET_BASKETS

Imports association rules that are generated by SAS Enterprise Miner and creates a rule flow that deploys the association rules.

Requirement: Before running this macro, you need to perform a market basket analysis using SAS Enterprise Miner and generate the data set that this macro uses as input.

Syntax

```
%BRM_IMPORT_MARKET_BASKETS (
  RULETABLE= %STR (association-rules-data-set),
  TRANSDATA= %STR (transactional-data-set),
  CUSTOMER= %STR (column-name),
  ITEM= %STR (column-name),
  FOLDER= %STR (top-level-folder</folder...>
    <, options>
  );
```

Required Arguments

CUSTOMER= %STR (column-name)

specifies the variable name of the customer column in the transactional data set. The column name must be a valid SAS variable name.

FOLDER= %STR (top-level-folder</folder...>)

specifies the business rules folder where you want to load the output of the %BRM_IMPORT_MARKET_BASKETS macro.

Requirement This folder must already exist. If it does not exist, the macro does not import any data, and it writes an error to the SAS log.

ITEM= %STR (column-name)

specifies the variable name of the Item Purchased column in the transactional data set. The column name must be a valid SAS variable name.

RULETABLE= %STR (association-rules-data-set)

specifies the data set that contains the association rules that was created by SAS Enterprise Miner. For information, see SAS Enterprise Miner documentation at <http://support.sas.com/documentation/onlinedoc/miner/index.html>.

TRANSDATA= %STR (transactional-data-set)

specifies the transactional SAS data set that contains the customer and item variables. This data set must contain the dictionary of the original transaction data that was used for the market basket analysis. The macro retrieves only the dictionary information for the customer and item variables.

Optional Arguments

BYPASSLOCK=Y|N

enables you to override the lock that another user has on the importing process.

Default N

CLUSTER=OPTGRAPH | FASTCLUS | NONE

specifies the method that was used to group the association rules in clusters. Associations analysis can produce a very large number of rules. Clustering creates multiple rule sets by grouping together rules that have similar sets of conditions. Multiple rule sets are easier to manage and more efficient than deploying a single set of many unrelated rules.

Default OPTGRAPH

DESCRIPTION= %STR (description)

specifies the description for the business rules vocabulary, the rule set, and the rule flow. The maximum length is 256 bytes.

Default Business Rules for Market Basket Analysis

LOADBRM=Y|N

specifies whether to load the output of the macro into the business rules database. If you want to review the data sets that this macro produces before the data is loaded into the business rules database, specify N. The data sets are named brm_ruleflow, Vocabulary, RuleInitialize, ConditionAssign, and ActionAssign_n, where _n is the cluster number.

Default	Y
----------------	---

Requirement	You must have access to the business rules metadata source in order to specify <i>Y</i> .
--------------------	---

NAME= %STR (*name*)

specifies the name of the business rules vocabulary and the prefix for the rule set and rule flow names. The name must be a valid vocabulary name.

Default	BR_for_MBA
----------------	------------

Details

About Market Basket Analysis

Associations analysis is used to find items, such as milk and bread, that are commonly purchased together more often than random chance would explain. Rules are then derived to predict the item that will be purchased when other items are purchased. This type of analysis is frequently referred to as market basket analysis. Market basket analysis can be used to predict the items that are most likely to be added to the basket. These predictions can be expressed as rules. For example:

- If a customer buys bread, then the customer is likely to buy milk. When this prediction is defined as a business rule in SAS Business Rules Manager, bread is a condition term, and milk is an action term.
- If a customer buys bread and milk, then the customer is likely to buy eggs. When this prediction is defined as a business rule in SAS Business Rules Manager, milk and bread are condition terms, and eggs is an action term.

These business rules can be used as recommendations. They can be used to design sales promotions, structure loyalty programs, determine discount plans, plan up-selling and cross-selling strategies, and so on.

The %BRM_IMPORT_MARKET_BASKETS macro uses the market basket data generated by SAS Enterprise Miner to create rule sets and a rule flow. It imports the rule sets and rule flow into the business rules database. Using SAS Business Rules Manager, you can enable or disable rules and modify rule conditions or rule actions. You can combine these market basket rules with other rules concerning business strategy such as pricing, discounts, customer segments, or loyalty. You can test the rule flow by using any input data that is in the same format as data that was used to perform the original associations analysis in SAS Enterprise Miner.

SAS Enterprise Miner provides several features that enable you to perform market basket analysis and produce the input data required by this macro. For more information about associations analysis, see SAS Enterprise Miner documentation at <http://support.sas.com/documentation/onlinedoc/miner/index.html>.

Business Rules Content Created by the Macro

The %BRM_IMPORT_MARKET_BASKETS macro imports business rules derived from a market basket analysis. Specifically, this macro creates a vocabulary, rule sets, and a complex rule flow, and imports all of this content into SAS Business Rules Manager. This content is imported into the folder specified by the FOLDER= option.

The vocabulary created by the macro is described in “[Vocabulary Created by the Macro](#)” on page 43.

In the rule sets created by the macro, purchased products are added to the decision table as condition terms. Recommended products are added to the decision table as action

terms. The statistical measures Support and Confidence are added to the decision table as action terms. No term is added as both a condition term and an action term.

Each rule is one recommendation. For example, a logical rule can be stated as follows:

IF basket CONTAINS ('apples', 'pears') THEN recommendation = celery

The %BRM_IMPORT_MARKET_BASKETS macro might translate this logical rule into a business rule that appears in the decision table as follows:

Condition Term				Action Term				
#	123 C_Apples	123 C_Pears	123 C_Oranges	123 Support	123 Confidence	123 RuleCluster...	123 A_Celery	123 A_Oranges
If	1	1		35.6	91.1	3	1	

The rule flows that %BRM_IMPORT_MARKET_BASKETS creates use the customer identifier variable as the BY-group term. Each unique customer identifier defines a BY-group. The rule flow contains the following rule sets:

- one rule set for initializing the indicator conditions and actions. This rule flow is in the Group Start section of the rule flow.
- one rule set that assigns values to indicator conditions. This rule set is in the Main section of the rule flow.
- one rule set for each cluster of rules. These rule sets are in the Group End section of the rule flow. The rules in the Group End section are the recommendations. You should examine these rules for patterns that you want to use in your business scenario. Look for actions that you might want to modify to fit your business strategy. For example, you might want to sell steak instead of eggs.

CAUTION:

You should not edit the rules in Group Start and Main sections except to change the name of an item. The rules in the Group Start and Main sections transform the transactional data so that it can be processed by the rules in the Group End section.

Vocabulary Created by the Macro

The %BRM_IMPORT_MARKET_BASKETS macro defines a vocabulary with four entities: Actions, Conditions, Data, and Measures.

The terms in each entity are listed in the following table.

Table 5.4 Vocabulary Created by the %BRM_IMPORT_MARKET_BASKETS Macro

Entity	Term	Exclude from input	Exclude from output	Description
Actions	A_product_name	Y	N	<p>The <i>product_name</i> is the name of a product in customer transaction data. The product (item) name must be a valid SAS variable name.</p> <p>In the rule sets, the value of A_product_name is 1 if the product name is a recommendation based on the association pattern. Otherwise, the value of A_product_name is 0.</p>

Entity	Term	Exclude from input	Exclude from output	Description
Conditions	<i>C_product_name</i>	Y	Y	The <i>product_name</i> is the name of a product in customer transaction data. The product (item) name must be a valid SAS variable name. In the rule sets, the value of <i>C_product_name</i> is 1 if the product name is found in the market basket and is a condition in the association pattern. Otherwise, the value of <i>C_product_name</i> is 0.
Data	<i>Customer</i>	N	N	Variable name of the Customer column in the transactional data.
Data	<i>Item</i>	N	Y	Variable name of the Item Purchased column in the transactional data.
Measures	Confidence	Y	Y	Confidence index of the association pattern. Confidence is a measure of the likelihood of the action given that the condition has occurred. This value is provided as information about the rule and is excluded from both input and output.
Measures	Support	Y	Y	Support index of the association pattern. Support is a measure of how often this pattern occurs in the source data. Users often want to find high confidence, low support rules. This value is provided as information about the rule and is excluded from both input and output.
Measures	RuleClusterIndex	Y	Y	Index of the cluster of rules found by the clustering method specified by the CLUSTER= option.

Example: Market Basket Analysis of Groceries Data

Note: This example requires SAS Enterprise Miner.

The following example generates a market basket analysis of the data in the SAMPSIO.ASSOCS data set. It generates a vocabulary, several rule sets, and a complex rule flow. It loads all of this content into the folder **MyFolder/Grocery**.

In the SAMPSIO.ASSOCS data set, the variable Customer is the customer identifier and is a numeric variable. The variable Product is the product identifier and is a character variable.

The following code performs the market basket analysis and generates the output data set that is used as input to the %BRM_IMPORT_MARKET_BASKETS macro.

```
proc dmdb data=sampsio.assocs dmdbcat=d;
  class customer product;
quit;

proc assoc data=sampsio.assocs dmdbcat=d out=a
  items=6 support=90;
  customer customer;
  target product;
```

```

run;

proc rulegen in=a dmdbcat=d out=rulegen
    minconf = 99;
run;

```

The %BRM_IMPORT_MARKET_BASKETS macro uses the output data set that the RULEGEN procedure created in order to generate and load a rule flow into the business rules database.

```

%BRM_IMPORT_MARKET_BASKETS (
    RuleTable = rulegen,
    TransData = sampsis.assocs,
    Customer = customer,
    Item = product,
    Name = %str(MBA_RuleCluster),
    Description = %str(MBA_RuleCluster),
    Folder = %str(MyFolder/Grocery),
    Cluster = OPTGRAPH,
    LoadBRM = Y,
    bypassLock = Y);

```

The following display shows part of the vocabulary that the %BRM_IMPORT_MARKET_BASKETS macro created. Because the NAME= option specifies MBA_RuleCluster, both the vocabulary and the rule flow that the macro created are named MBA_RuleCluster.

Display 5.1 Vocabulary MBA_RuleCluster

▼ Grocery	Folder	Market Basket example using Grocery Data SAMPISIO.ASSOCS		
▼ MBA_RuleCluster	Vocabulary	MBA Rule Cluster		
▼ Actions	Entity	Market Basket Actions		
123 A_avocado	Term	Market Basket Action: A_avocado	Integer	Discrete
123 A_baguette	Term	Market Basket Action: A_baguette	Integer	Discrete
123 A_bourbon	Term	Market Basket Action: A_bourbon	Integer	Discrete
123 A_chicken	Term	Market Basket Action: A_chicken	Integer	Discrete
123 A_coke	Term	Market Basket Action: A_coke	Integer	Discrete
123 A_corned_b	Term	Market Basket Action: A_corned_b	Integer	Discrete
123 A_cracker	Term	Market Basket Action: A_cracker	Integer	Discrete
123 A_heineken	Term	Market Basket Action: A_heineken	Integer	Discrete
123 A_hering	Term	Market Basket Action: A_hering	Integer	Discrete
123 A_ice_crea	Term	Market Basket Action: A_ice_crea	Integer	Discrete
123 A_olives	Term	Market Basket Action: A_olives	Integer	Discrete
123 A_sardines	Term	Market Basket Action: A_sardines	Integer	Discrete
► Conditions	Entity	Market Basket Conditions		
▼ Data	Entity	Market Basket Transaction Data Variables		
123 CUSTOMER	Term	Market Basket Transaction Data Variable: CUSTOMER	Decimal	Discrete
▲ PRODUCT	Term	Market Basket Transaction Data Variable: PRODUCT	String	Discrete
▼ Measures	Entity	Market Basket Measures		
123 Confidence	Term	Market Basket Measure: Confidence	Decimal	Continuous
123 RuleClusterIndex	Term	Market Basket Measure: RuleClusterIndex	Integer	Discrete
123 Support	Term	Market Basket Measure: Support	Decimal	Continuous

The following display shows the structure of the MBA_RuleCluster rule flow. The rules in the Group Start and Main sections transform the transactional data so that it can be processed by the rules in the Group End section.

Display 5.2 Rule Flow MBA_RuleCluster

▼ Group Start (Start of each BY-group) [1]		
	Name	Description
1	MBA_RuleCluster, Initialize	Market Basket Rule: MBA_RuleCluster, Step 1: Initialization
▼ Main [1]		
	Name	Description
1	MBA_RuleCluster, Assign Conditions	Market Basket Rule: MBA_RuleCluster, Step 2: Assignment of Conditions
▼ Group End (End of each BY-group) [5]		
	Name	Description
1	MBA_RuleCluster, Implement Rule Cluster 1	Market Basket Rule: MBA_RuleCluster, Step 3: Implementation of Rule Cluster 1
2	MBA_RuleCluster, Implement Rule Cluster 2	Market Basket Rule: MBA_RuleCluster, Step 3: Implementation of Rule Cluster 2
3	MBA_RuleCluster, Implement Rule Cluster 3	Market Basket Rule: MBA_RuleCluster, Step 3: Implementation of Rule Cluster 3
4	MBA_RuleCluster, Implement Rule Cluster 4	Market Basket Rule: MBA_RuleCluster, Step 3: Implementation of Rule Cluster 4
5	MBA_RuleCluster, Implement Rule Cluster 5	Market Basket Rule: MBA_RuleCluster, Step 3: Implementation of Rule Cluster 5

The following display shows the condition table for the first rule set in the Group End section, RuleCluster1. The rules in the Group End section are the recommendations.

Display 5.3 Condition Table in Rule Set RuleCluster1

Condition Term									
	#	¹²³ C_artichok	¹²³ C_avocado	¹²³ C_baguette	¹²³ C_cracker	¹²³ C_ham	¹²³ C_heineken	¹²³ C_hering	¹²³ C_soda
<div><div>if</div><div></div><div></div></div>	1			1	1				1
<div><div>if</div><div></div><div></div></div>	2	1	1		1				
<div><div>if</div><div></div><div></div></div>	3	1			1	1			
<div><div>if</div><div></div><div></div></div>	4			1	1			1	1
<div><div>if</div><div></div><div></div></div>	5	1	1	1				1	
<div><div>if</div><div></div><div></div></div>	6	1	1		1	1			
<div><div>if</div><div></div><div></div></div>	<div><div>+</div></div>	1			1	1	1		

The following display shows the action table for the first rule set in the Group End section, RuleCluster1. You should examine the patterns in the rule sets in the Group End section. You might find actions that you want to modify to fit your business strategy.

Display 5.4 Action Table in Rule Set RuleCluster1

Action Term				
123 Support	123 Confidence	123 RuleCluster...	123 A_avocado	123 A_heineken
12.58741258741	99.21259842519	1		1
11.18881118881	99.11504424778	1		1
9.990009990009	99.00990099009	1		1
11.48851148851	100	1		1
9.890109890109	100	1		1
9.890109890109	100	1		1
9.890109890109	99	1	1	

%BRM_IMPORT_RULE_FLOW

Imports rule flows from the specified CSV file into the rules database.

Restrictions: This macro must be run on the server tier.
The same user can run any of import macros at the same time. However, multiple users cannot run the same import macro simultaneously.

Syntax

```
%BRM_IMPORT_RULE_FLOW(CSV=input_filename.CSV,
REJECT=reject_filename.CSV<, options>);
```

Required Arguments

CSV=input_filename

specifies the full pathname to the CSV file where you want to import the data from.
For more information, see [“Format of the Rule Flow CSV Input File” on page 48](#).

REJECT=reject_filename

specifies the full pathname to the CSV file where you want the macro to write any records that were not imported to the rules database. See [“Using the %BRM_IMPORT_RULE_FLOW Macro” on page 48](#) for more information.

Optional Arguments

BRM_USER=user_ID

specifies the user ID that you want to be associated with the data that is imported.
This user ID is associated with the imported objects in the rules database and is displayed in the SAS Business Rules Manager interface.

Default User ID of the user that is running the macro

BYPASSLOCK=Y|N

enables you to override the lock that another user has on the importing process. See [“Using the %BRM_IMPORT_RULE_FLOW Macro” on page 48](#) for more information.

Default N

Details

Using the %BRM_IMPORT_RULE_FLOW Macro

The %BRM_IMPORT_RULE_FLOW macro enables you to add new rule flows and to update existing rule flows. The macro uses the rule flow name and rule flow path to determine whether a rule flow already exists. If the rule flow path and name already exist, then the rule flow is updated. If the rule flow path exists but the rule flow name does not exist, the rule flow is created. If the rule flow path does not exist, then the rule flow is rejected.

The %BRM_IMPORT_RULE_FLOW macro runs several validation checks as it imports the rule flows. For example, it checks whether a rule set is referenced in a given rule flow more than once and whether section codes are correct. If the macro finds a validation error in a rule flow, it writes a message to the SAS log, and the rule flow is rejected. The macro writes the input records for the rejected rule flow to the CSV file that was specified in the REJECT= option.

When you run the %BRM_IMPORT_RULE_FLOW macro, it creates a lock table in the rules database named lock_import_rule_flow. The SAS log states which user holds the lock and the time at which the lock started. This lock might remain in place after the macro has finished. If this happens, you can override the lock by specifying the BYPASSLOCK=Y option when you run the macro.

Format of the Rule Flow CSV Input File

Each row of the CSV input file identifies a rule set and a rule flow, and each row provides the information about how that rule set fits into the rule flow. The CSV file must contain all of the columns that are listed in the following table, in the order listed. You must specify values for all columns, except as noted in the table. To create a blank column in the CSV file, specify two comma separators without any content between them. For example, to add a rule set to the main section of the rule flow named assignRisk and to specify a blank column for the rule flow description, specify the following in the CSV file:

```
assignRisk,,main
```

Table 5.5 Format of the Rule Flow CSV Input File

Column	Description	Can Column Be Blank
RULE_FLOW_SK	The identification number of the rule flow.	Yes
RULE_FLOW_NM	The name of the rule flow where you want to add the rule set that is specified in RULE_SET_NM.	No
RULE_FLOW_SHORT_DESC	The description of the rule flow.	Yes

Column	Description	Can Column Be Blank
RULE_SET_SECTION_CODE	<p>The section of the rule flow to which the rule set that is specified in RULE_SET_NM belongs. Specify init, groupstart, main, groupend, or final.</p> <p>The codes groupstart and groupend are valid only if you also specify at least one term for BY_TERM. See “Simple Rule Flows, Complex Rule Flows, and BY Groups” in Chapter 7 of <i>SAS Business Rules Manager: User's Guide</i> for more information.</p>	No
RULE_FLOW_PATH	The pathname to the business rules folder for the rule flow. This path must exist. Separate folder names with forward slashes.	No
RULE_SET_NM	The name of the rule set to be added to the rule flow. A rule set can be added to the same rule flow only once.	No
RULE_SET_PATH	The pathname to the business rules folder to the rule set that is specified by RULE_SET_NM. The rule set must exist at the specified location. Separate folder names with forward slashes.	No
VOCAB_NM	The name of the vocabulary that the rule set uses. All rule sets in the same rule flow must use the same vocabulary.	No
BY_TERM	<p>The list of BY-group terms that the rule set uses. Separate multiple BY-group terms with commas.</p> <p>The BY-group terms must be the same for all rule sets that are in the same rule flow. All of the BY-group terms must belong to the same vocabulary.</p> <p>See “Simple Rule Flows, Complex Rule Flows, and BY Groups” in Chapter 7 of <i>SAS Business Rules Manager: User's Guide</i> for more information.</p>	Yes
ORDER	The order number for the rule set that is in the rule flow. Order numbers must start with 1 and be continuous through the entire rule flow. Do not restart order numbers at section boundaries.	No

%BRM_IMPORT_RULESET

Imports rule sets from the specified CSV file into the rules database.

Restrictions: This macro must be run on the server tier.

The same user can run any of import macros at the same time. However, multiple users cannot run the same import macro simultaneously.

Syntax

```
%BRM_IMPORT_RULESET (CSV=input_filename.CSV,
REJECT=reject_filename.CSV<, options>);
```

Required Arguments

CSV=*input_filename*

specifies the full pathname to the CSV file where you want to import the data from. For more information, see [“Format of Rule Set CSV Input File” on page 50](#).

REJECT=*reject_filename*

specifies the full pathname to the CSV file where you want the macro to write any records that were not imported to the rules database. See [“Using the %BRM_IMPORT_RULESET Macro” on page 50](#) for more information.

Optional Arguments

BRM_USER=*user_ID*

specifies the user ID that you want to be associated with the data that is imported. This user ID is associated with the imported objects in the rules database and is displayed in the SAS Business Rules Manager interface.

Default User ID of the user that is running the macro

BYPASSLOCK=Y|N

enables you to override the lock that another user has on the importing process. See [“Using the %BRM_IMPORT_RULESET Macro” on page 50](#) for more information.

Default N

Details

Using the %BRM_IMPORT_RULESET Macro

The %BRM_IMPORT_RULESET macro enables you to add new rule sets and to update existing rule sets. The macro uses the rule set name and rule set path to determine whether a rule set already exists. If the rule set path and name already exist, then the rule set is updated. If the rule set path exists but the rule set name does not exist, the rule set is created. If the rule set path does not exist, then the rule set is rejected.

The %BRM_IMPORT_RULESET macro runs several validation checks as it imports the rule sets. For example, it verifies that the expressions are valid, ensures that the first rule in each rule set uses the IF operator, and verifies that the specified vocabularies exist. If the macro finds a validation error in a rule set, it writes a message to the SAS log, and the rule set is rejected. The macro writes the input records for the rejected rule set and the reason that the record was rejected to the CSV file that was specified in the REJECT= option.

When you run the %BRM_IMPORT_RULESET macro, it creates a lock table in the rules database named lock_import_rule_set. The SAS log states which user holds the lock and the time at which the lock started. This lock might remain in place after the macro has finished. If this happens, you can override the lock by specifying the BYPASSLOCK=Y option when you run the macro.

Format of Rule Set CSV Input File

Each row of the CSV input file specifies a rule, rule set, term, and an expression for that term. The row also specifies whether the expression is a condition expression or an action expression. Each row of the input file can specify only one condition expression or one action expression for a given rule. The CSV file must contain all of the columns that are listed in the following table, in the order listed. You must specify values for all

columns, except as noted in the table. To create a blank column in the CSV file, specify two comma separators without any content between them. For example, to add a rule to the rule set named assignRisk that uses the loanVocab vocabulary and to specify a blank column for the rule set description, specify the following in the CSV file:

```
assignRisk,,loanVocab
```

Table 5.6 Format of the Rule Set CSV Input File

Column	Description	Can Column Be Blank
RULE_SET_SK	The identification number of the rule set.	Yes
RULE_SET_NM	The name of the rule set where you want to add the rule that is specified in RULE_NM.	No
RULE_SET_DESC	The description of the rule set.	Yes
VOCAB_NM	The name of the vocabulary that the rule set uses. All rules in the same rule set must use the same vocabulary.	No
RULE_SET_PATH	The pathname to the business rules folder for the rule set. This path must exist. Separate folder names with forward slashes.	No
RULE_NM	The name of the rule to be added to the rule set.	No
RULE_DESC	The description of the rule.	Yes
RULE_SEQ_NO	The order number for the rule that is in the rule set. Order numbers in a rule set start with 1.	No
CONDITIONAL_NM	The operator for the rule. Specify if , elseif , or or . The first rule in a rule set must use the if operator. For information about these operators, see “Controlling Which Conditions Are Evaluated” in Chapter 6 of <i>SAS Business Rules Manager: User's Guide</i> .	No
LHS_TERM	The term for the expression specified in the EXPRESSION column. Terms that are specified in the LHS_TERM column are the terms that SAS Business Rules Manager displays at the top or left side of the decision table. These terms appear in the column headings of the decision table when you are viewing the decision table in the horizontal format. They appear in the row headings of the decision table when you are viewing the decision table in the vertical format.	No

Column	Description	Can Column Be Blank
EXPRESSION	A single condition or action expression for the term specified in the LHS_TERM column. This expression is the expression that you would enter into a cell in the decision table. See “Defining New Rules in the Rule Set” in Chapter 6 of <i>SAS Business Rules Manager: User's Guide</i> for more information about expressions.	Yes
EXPRESSION_ORDER	The order number of the rule's condition or action expressions. A rule's condition and action expressions are numbered beginning with 1. For example, a rule might have two condition expressions that are numbered 1 and 2, and it might have three action expressions that are numbered 1, 2, and 3.	No
EXPRESSION_TYPE	The type of expression. Specify condition or action .	No

%BRM_IMPORT_VOCABULARY

Imports vocabulary terms from the specified CSV file into the rules database.

Restrictions: This macro must be run on the server tier.
The same user can run any of import macros at the same time. However, the same import macro cannot be run simultaneously by different users.

Syntax

```
%BRM_IMPORT_VOCABULARY (CSV=input_filename.CSV,
REJECT=reject_filename.CSV<, options>);
```

Required Arguments

CSV=input_filename

specifies the full pathname to the CSV file where you want to import the data from.
For more information, see “Format of the Vocabulary CSV Input File” on page 53.

REJECT=reject_filename

specifies the full pathname to the CSV file where you want the macro to write any records that were not imported to the rules database. See “Using the %BRM_IMPORT_VOCABULARY Macro” on page 53 for more information.

Optional Arguments

BRM_USER=user_ID

specifies the user ID that you want to be associated with the data that is imported.
This user ID is associated with the imported objects in the rules database and is displayed in the SAS Business Rules Manager interface.

Default User ID of the user that is running the macro

BYPASSLOCK=Y|N

enables you to override the lock that another user has on the importing process. See [“Using the %BRM_IMPORT_VOCABULARY Macro” on page 53](#) for more information.

Default N

Details

Using the %BRM_IMPORT_VOCABULARY Macro

The %BRM_IMPORT_VOCABULARY macro enables you to add new vocabulary terms. You cannot use the macro to update existing terms.

The %BRM_IMPORT_VOCABULARY macro runs several validation checks as it imports the vocabulary terms. For example, it verifies that term, entity, and vocabulary names are valid, and ensures that a term is not duplicated in a vocabulary. If the macro finds a validation error, it writes a message to the SAS log, and the term is rejected. The macro writes the input records for the rejected term to the CSV file that was specified in the REJECT= option.

When you run the %BRM_IMPORT_VOCABULARY macro, it creates a lock table in the rules database named lock_import_vocabulary. The SAS log states which user holds the lock and the time at which the lock started. It is possible for this lock to remain in place after the macro has finished. If this happens, you can override the lock by specifying the BYPASSLOCK=Y option when you run the macro.

Format of the Vocabulary CSV Input File

Each row of the CSV input file defines a term, including the term data type, domain type, and the entity and vocabulary that contains the term. The CSV file must contain all of the columns listed in the following table, in the order listed. You must specify values for all columns, except as noted in the table. To create a blank column in the CSV file, specify two comma separators without any content between them. For example, to add a term to the entity named Customer in the vocabulary named loanVocab and to specify a blank column for the vocabulary description, specify the following in the CSV file:

```
loanVocab,,Customer
```

Table 5.7 Format of the Vocabulary CSV Input File

Column	Description	Can Column Be Blank
VOCAB_NM	The name of the vocabulary where you want to add entity and term specified by VOCAB_ENTITY_NM and VOCAB_TERM_NM.	No
VOCAB_SHORT_DESC	The description of the vocabulary.	Yes
VOCAB_ENTITY_NM	The name of the entity that the term in the VOCAB_TERM_NM column belongs to.	No
VOCAB_ENTITY_SHORT_DESC	The description of the entity.	Yes

Column	Description	Can Column Be Blank
VOCAB_TERM_NM	The name of the term.	No
VOCAB_TERM_SHORT_DESC	The description of the term.	Yes
VOCAB_TERM_DATA_TYPE_TXT	The data type of the term. Specify string , decimal , integer , boolean , date , or datetime .	No
VOCAB_TERM_DOMAIN_TYPE_TXT	The domain type for the term. Specify discrete , continuous , or boolean . A domain value is discrete if it is just an individual value such as 5.3 or 18JUL2012:10:25:00 . A domain value is continuous if it specifies a range such as >5 or <18JUL2012:10:25:00 . Terms that are assigned the data type string can have discrete domain values only. Boolean terms can have Boolean domain values only.	No
VOCAB_TERM_DOMAIN_TXT	The set of expected values for a term. Separate individual domain values with a semi-colon (;). See “Specify Domain Values” in Chapter 4 of <i>SAS Business Rules Manager: User's Guide</i> for more information.	Yes
VOCAB_TERM_INPUT_EXCLUDE_FLG	Specifies whether the term must be mapped to a column in an input data set. Specify Y or N .	No
VOCAB_TERM_OUTPUT_EXCLUDE_FLG	Specifies whether to exclude the term from the output data sets created by rule flows. Specify Y or N .	No
FOLDER_PATH	The pathname to the business rules folder for the rule flow. This path must exist. Separate folder names with forward slashes.	No

%BRM_LOAD_VOCABULARY

Loads the vocabulary terms in the WORK.TERM data set that was created by the %BRM_CREATE_TEMP_TERM macro.

Syntax

```
%BRM_LOAD_VOCABULARY (FOLDER_PATH=path,
VOCAB_NM=vocabulary-name,
VOCAB_ENTITY_NM=entity-name<, options>);
```

Required Arguments

FOLDER_PATH=path-name

specifies the pathname to the business rules folder where you want to import the vocabulary terms. Separate folder names with forward slashes.

Requirement The path must exist. If the path does not exist, the macro terminates and writes an error message to the SAS log.

Example FOLDER_PATH=Loans/Retail/Applications

VOCAB_NM=*vocabulary-name*

specifies the name of the vocabulary to which the terms in the WORK.TERM file will be added.

Requirement The vocabulary must not exist. If it already exists, the macro terminates and writes an error message to the SAS log.

VOCAB_ENTITY_NM=*entity-name*

specifies the name of the entity to which the terms in the WORK.TERM file will be added.

Requirement This entity must not exist. If it already exists, the macro terminates and writes an error message to the SAS log.

Optional Arguments

BRM_USER=*user_ID*

specifies the user ID that you want to be associated with the data that is imported. This user ID is associated with the imported objects in the rules database and is displayed in the SAS Business Rules Manager interface.

Default User ID of the user that is running the macro

BYPASSLOCK=Y|N

enables you to override the lock that another user has on the importing process.

Default N

Details

When you run the %BRM_LOAD_VOCABULARY macro, it creates a lock table in the rules database named lock_import_vocabulary. The SAS log states which user holds the lock and the time at which the lock started. It is possible for this lock to remain in place after the macro has finished. If this happens, you can override the lock by specifying the BYPASSLOCK=Y option when you run the macro.

Index

A

access permissions
 capabilities [25](#)
 groups [24](#)
 roles [25](#)
 SAS identities [24](#)
 accounts
 capabilities [25](#)
 roles [25](#)
 SAS identities [24](#)
 administrators
 administrative group [25](#)
 Adobe Flash Player [9](#)
 architecture
 SAS Intelligence Platform [3](#)

C

capabilities [25](#)
 client tier [3](#), [14](#)

D

data tier [3](#)
 deployment plan [9](#)
 downloading software [10](#)

G

groups [24](#)
 creating as pre-installation task [9](#)
 Public [25](#)
 SAS Users [25](#)

I

installing SAS Business Rules Manager
 installing additional products [14](#)
 post-installation tasks [18](#)
 pre-installation requirements and tasks
 [8](#)

running the SAS Deployment Wizard
 [14](#)
 single-machine versus multiple-machine
 installations [14](#)
 using the SAS Deployment Wizard [13](#)
 Instructions.html file [19](#)

M

metadata
 access to [24](#)
 middle tier [3](#), [14](#)
 multiple-machine installations [14](#)

O

operating system requirements [8](#)
 Oracle [9](#)

P

permissions
 capabilities [25](#)
 groups [24](#)
 roles [25](#)
 SAS identities [24](#)
 post-installation tasks [18](#)
 pre-installation requirements and tasks [8](#)
 pre-installation tasks
 database [10](#)
 prerequisite software [9](#)

R

roles [25](#)
 creating [28](#)
 modifying [29](#)

S

SAS Business Rules Manager

- documentation 2
- features 1
- installation and configuration 8
- SAS Deployment Wizard 13
 - running 14
- SAS Download Manager 10
- sas group 9
- SAS identities 24
- SAS Installation Data (SID) file 9
- SAS Integration Technologies 14
- SAS Intelligence Platform
 - architecture 3
 - security administration 23
- SAS Management Console 14
- SAS server tier 3
- SAS Server Users group 9
- SAS Software Depot 10
- security
 - capabilities 25
 - groups 24
 - roles 25

- SAS identities 24
- SAS Intelligence Platform 23
- server tier 14
- SID file 9
- single-machine installations 14

T

- tier
 - client 3
 - data 3
 - definition of 3
 - middle 3
 - SAS server 3

U

- user accounts
 - creating as pre-installation task 9
- user IDs
 - SAS identities 24