



# **SAS<sup>®</sup> Business Rules Manager 3.1: Administrator's Guide**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS® Business Rules Manager 3.1: Administrator's Guide*. Cary, NC: SAS Institute Inc.

**SAS® Business Rules Manager 3.1: Administrator's Guide**

Copyright © 2015, 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, NC 27513-2414

February 2017

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.

3.1-P3:brsag

---

# Contents

## PART 1 Introduction to SAS Business Rules Manager 1

<b>Chapter 1 • Overview of SAS Business Rules Manager</b> .....	<b>3</b>
Enterprise Decision Management Systems .....	3
About SAS Business Rules Manager .....	3
The SAS Intelligence Platform and SAS Business Rules Manager .....	4
Process for Publishing Rule Flows .....	6

## PART 2 Installation and Configuration 7

<b>Chapter 2 • Performing Pre-installation Tasks</b> .....	<b>9</b>
Pre-installation Steps .....	9
Determine the Database to Use .....	10
Determine Time Zone Requirements .....	10
Pre-installation Tasks for SAS Decision Manager Common Data Server .....	10
Pre-installation Tasks for an Oracle Database .....	11
<b>Chapter 3 • Installing SAS Business Rules Manager</b> .....	<b>15</b>
Products Installed with SAS Business Rules Manager .....	15
Running the SAS Deployment Wizard .....	15
<b>Chapter 4 • Performing Post-installation Tasks</b> .....	<b>17</b>
Post-installation Configuration and Verification Steps .....	17
Create Users and Assign Permissions .....	18
Creating Operating System Accounts for Product Administrators and Users .....	18
Create Oracle Database Synonyms .....	22
Create Oracle Database Tables .....	22
Verify the Certificate .....	23
Configure Your Deployment for HTTPS .....	23
Configure the Limit for Character Strings .....	24
Review Business Rules Manager Web Properties .....	25
Modify Log File Settings .....	26
Manage Directories for Business Rules Content .....	28
<b>Chapter 5 • Performing Migration Tasks</b> .....	<b>29</b>
About the Migration Process .....	29
Pre-migration Steps .....	29
Post-migration Steps .....	31
<b>Chapter 6 • Performing Upgrade Tasks</b> .....	<b>37</b>
About the Upgrade Process .....	37
Pre-upgrade Steps .....	38
Run the SAS Deployment Wizard .....	39
Post-upgrade Steps .....	40
<b>Chapter 7 • Configuring Users, Groups, and Roles</b> .....	<b>43</b>
Security Administration Tasks for SAS Business Rules Manager .....	43

Administering SAS Identities for Users .....	44
Groups and Group Membership .....	45
Roles and Capabilities .....	46
Administering Group and Role Membership .....	48
Updating Groups and Roles for SAS Business Rules Manager 3.1 .....	51
<b>Chapter 8 • Configuring SAS Workflow .....</b>	<b>53</b>
Configuring SAS Workflow for Use with SAS Business Rules Manager .....	53
Configure Alert Notifications for SAS Workflow .....	57
PART 3 <b>Macro Reference</b>	<b>59</b>
<b>Chapter 9 • Macros for Importing and Exporting .....</b>	<b>61</b>
Introduction to the Import and Export Macros .....	61
Dictionary .....	62
<b>Chapter 10 • Macros for Running Rule Flows .....</b>	<b>81</b>
Introduction to Macros for Running Rule Flows .....	81
Dictionary .....	81
<b>Index .....</b>	<b>87</b>

## **Part 1**

---

# Introduction to SAS Business Rules Manager

### *Chapter 1*

**Overview of SAS Business Rules Manager** ..... 3



*Chapter 1*

# Overview of SAS Business Rules Manager

---

<b>Enterprise Decision Management Systems</b> .....	<b>3</b>
<b>About SAS Business Rules Manager</b> .....	<b>3</b>
<b>The SAS Intelligence Platform and SAS Business Rules Manager</b> .....	<b>4</b>
<b>Process for Publishing Rule Flows</b> .....	<b>6</b>

## Enterprise Decision Management Systems

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

## About SAS Business Rules Manager

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:

### data management

You can manage your list of data tables from within the application. You can create new Base SAS libraries, add and remove tables, view table data and metadata, create

and delete table summaries, and associate attachments and comments with tables. The application uses these data tables whenever it needs to access data, such as for rule discovery and rule flow testing.

#### vocabulary management

A business vocabulary defines entities and terms. Terms are the building blocks that you use to construct 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 Equation Editor to create the expressions for the rule.

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

#### rule set management

A rule set is a logical collection of rules. A single rule set can have many rules. For example, you might have a rule set that determines a customer's asset balance and another rule set that determines a customer's debt level. SAS Business Rules Manager displays rule sets in decision tables. Each row of the 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.

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

#### rule flow authoring and publishing

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

SAS Business Rules Manager makes it easy to combine rule 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.

---

## 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 an easy-to-use interface. With SAS Business Rules Manager, users author rule sets and rule flows through the SAS Business Rules Manager client.

middle tier

provides an environment in which the SAS Business Rules Manager web application can execute. The middle tier passes analysis and processing requests to the SAS servers.

server tier

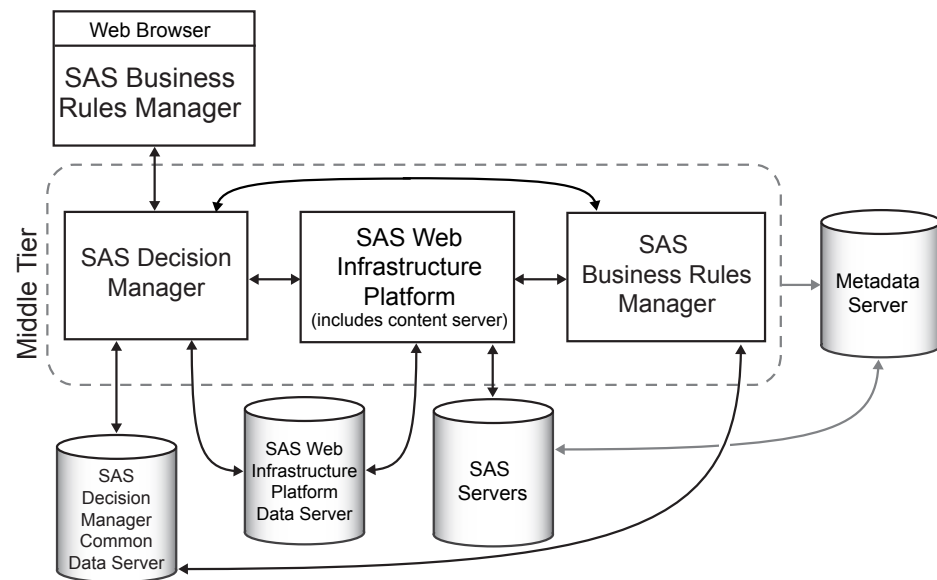
provides SAS servers that process data and handle client requests. For SAS Business Rules Manager, the server tier provides the SAS Business Rules Engine.

data tier

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

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

**Figure 1.1** SAS Intelligence Platform Architecture and SAS Business Rules Manager



SAS Business Rules Manager

the 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 data source and workflow management.

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 SAS Decision Manager 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.

SAS Decision Manager Common Data Server  
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.

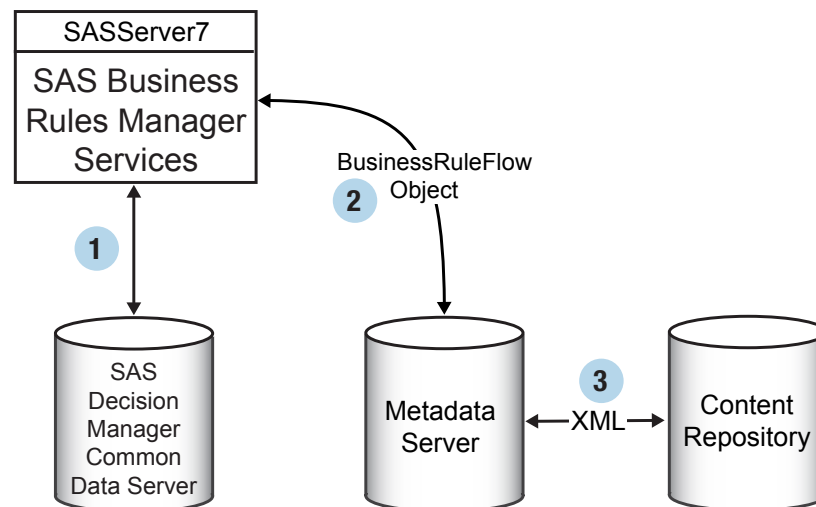
---

## 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 1.2** Process for Publishing Rule Flows



- 1 SAS Business Rules Manager reads the rule flow data in the SAS Decision Manager 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.

## Part 2

---

# Installation and Configuration

<i>Chapter 2</i>	
<b>Performing Pre-installation Tasks</b> .....	9
<i>Chapter 3</i>	
<b>Installing SAS Business Rules Manager</b> .....	15
<i>Chapter 4</i>	
<b>Performing Post-installation Tasks</b> .....	17
<i>Chapter 5</i>	
<b>Performing Migration Tasks</b> .....	29
<i>Chapter 6</i>	
<b>Performing Upgrade Tasks</b> .....	37
<i>Chapter 7</i>	
<b>Configuring Users, Groups, and Roles</b> .....	43
<i>Chapter 8</i>	
<b>Configuring SAS Workflow</b> .....	53



## Chapter 2

# Performing Pre-installation Tasks

---

<b>Pre-installation Steps</b> .....	<b>9</b>
<b>Determine the Database to Use</b> .....	<b>10</b>
<b>Determine Time Zone Requirements</b> .....	<b>10</b>
<b>Pre-installation Tasks for SAS Decision Manager Common Data Server</b> .....	<b>10</b>
<b>Pre-installation Tasks for an Oracle Database</b> .....	<b>11</b>
About the Oracle Pre-installation Tasks .....	11
Verify JDBC Drivers for Oracle .....	11
Determine the Information Required for the Oracle Database .....	12
Specify the Required Database Privileges for Oracle .....	13
Test the Connection to Your Database .....	13

---

## Pre-installation Steps

Before you 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 steps before you install SAS Business Rules Manager:

1. Verify that your system meets the minimum requirements. See the [system requirements](#) documentation on [support.sas.com](http://support.sas.com).
2. [Determine the database that you want to use.](#)
3. [Determine the time zone requirements.](#) You might need to synchronize the time zones that are specified in all of your operating environments.
4. Complete the pre-installation steps for your database. See “[Pre-installation Tasks for SAS Decision Manager Common Data Server](#)” on page 10 or “[Pre-installation Tasks for an Oracle Database](#)” on page 11.

---

## Determine the Database to Use

You can use either Oracle or the SAS Decision Manager Common Data Server for the SAS Decision Manager database.

The SAS Decision Manager Common Data Server is based on PostgreSQL 9.1.9. For more information, see “[SAS Web Infrastructure Platform Data Server](http://support.sas.com/documentation/intellplatform/index.html)” in *SAS Intelligence Platform: Middle-Tier Administration Guide* at <http://support.sas.com/documentation/intellplatform/index.html>.

For Oracle, complete the tasks described in “[Pre-installation Tasks for an Oracle Database](#)” on [page 11](#). For SAS Decision Manager Common Data Server, ensure that you have the information listed in “[Pre-installation Tasks for SAS Decision Manager Common Data Server](#)” on [page 10](#).

---

## Determine Time Zone Requirements

All of your operating environments (on all tiers in a multi-tier environment) must be set to the same time zone only if your site meets both of these conditions:

- You will deploy rule flows by using SAS Real-Time Decision Manager.
- Those rule flows use terms of type Date or Datetime.

SAS Real-Time Decision Manager uses a custom data type that accounts for differences between time zones when it performs calculations. If the time zones do not match across all of your environments, you should not use Date or Datetime data types in rule flows.

---

## Pre-installation Tasks for SAS Decision Manager Common Data Server

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.

If you are using the SAS Decision Manager Common Data Server that is based on Postgres, you need the information in the following table in order to complete the steps in the SAS Deployment Wizard.

**Table 2.1** SAS Deployment Wizard Information for SAS Decision Manager Common Data Server

Property	Description
Database Type	Specifies the database type to use for the SAS Decision Manager database. Select <b>SAS Decision Manager Common Data Server</b> .
Database Name	Specifies the database name. The default name for the database is <b>dcmdb</b> .

---

Property	Description
<b>Database User</b>	Specifies the user name for the database administrator. This user owns the database and has superuser privileges. The default user name is <b>dcmdbowner</b> .
<b>Database Password</b>	Specifies a password for the user ID that is associated with the database account.
<b>Port</b>	Specifies the port that is used by the database. The default port for SAS Decision Manager Common Data Server is 10482.
<b>Host Name</b>	Specifies the fully qualified host name of the server on which the database is installed.
<b>User ID</b>	Specifies the user name for the user whose credentials will be used to access the SAS Decision Manager Common Data Server database. The default user name is <b>dcmdb</b> .

For database-specific information about configuring a database, see *SAS Intelligence Platform: Installation and Configuration Guide*.

## Pre-installation Tasks for an Oracle Database

### About the Oracle Pre-installation Tasks

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

1. [Verify that you have the correct JDBC drivers.](#)
2. Verify that you have a tnsnames.ora file for your Oracle client that corresponds to your database.
3. [Determine the required database information.](#)
4. [Specify the required database privileges.](#)
5. [Test the connection to your database.](#)

### Verify JDBC Drivers for Oracle

Verify that you have the correct JDBC drivers. To ensure proper installation of SAS Business Rules Manager, the drivers must be on each middle-tier server, and they must be in a directory that does not contain any other files.

The JDBC drivers for Oracle are located in the Oracle installation directory. You can also download the ojdbc6.jar file for Oracle Database 11g from <http://www.oracle.com/us/downloads/index.html>. Select **JDBC drivers**. Download the latest Oracle 11.2x driver. The JDBC driver version must match the database version.

For more information about supported database drivers, see the system requirements documents at <http://support.sas.com/documentation/installcenter/>.

## Determine the Information Required for the Oracle Database

During the installation and configuration of SAS Business Rules Manager, the SAS Deployment Wizard requires information about the Oracle database that SAS Business Rules Manager uses. Record the information in the following table.

You enter this information in the SAS Decision Manager Database Properties and SAS Decision Manager Database JDBC Properties windows.

**Table 2.2** SAS Deployment Wizard Information for Oracle

Prompt	Description
<b>Host Name</b>	Specifies the fully qualified host name of the server on which the database is installed.
<b>Port</b>	Specifies the port number that is used by the database. The default port for Oracle is 1521.
<b>Directory containing JDBC driver jars</b>	<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 to configure SAS Decision Manager database.</p> <p>See <a href="#">“Verify JDBC Drivers for Oracle” on page 11</a> for more information.</p>
<b>Database SID or Service Name</b>	<p>Specifies the Oracle database name. The database name must match either the service name or the Oracle site identifier (SID), both of which can be found in the tnsnames.ora file.</p> <p>If you select <b>Use Oracle database name as a Service Name</b>, then you must enter the service name that is specified in the tnsnames.ora file. For example, if you had the following entry in the tnsnames.ora file, you would enter <code>monitordb</code> in the <b>Database SID or Service Name</b> field:</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><i>Note:</i> In the tnsnames.ora file, the <b>Net Service Name</b> and the <b>Service Name</b> fields must be the same.</p> <p>You can also find the Oracle SID in the tnsnames.ora file. Alternatively, you can run the following query using a database user ID on your Oracle instance:</p> <pre> select instance from v\$thread </pre>
<b>User ID</b>	Specifies the user ID of the database user whose credentials are used to access SAS Business Rules Manager data on the server.
<b>Password</b>	Specifies the password of the user ID whose credentials are used to access SAS Business Rules Manager data on the server.



Prompt	Description
<b>Schema Pattern</b>	Specifies the schema name for the database. The default schema is the same as the user ID.

### ***Specify the Required Database Privileges for Oracle***

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

- 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 specified by the PATH variable.



## Chapter 3

# Installing SAS Business Rules Manager

---

<b>Products Installed with SAS Business Rules Manager</b> .....	<b>15</b>
<b>Running the SAS Deployment Wizard</b> .....	<b>15</b>
About Running the SAS Deployment Wizard .....	15
Select the SAS Application Server .....	16
Configure the Database .....	16
Create and Load Tables through the SAS Deployment Wizard .....	16

---

## 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. See the software order e-mail or the `ordersummery.html` file that is in your SAS Software Depot at `software_depot/install_doc/order_number/ordersummary.html`. The SAS Deployment Wizard prompts you to install and configure each of the products in your deployment plan.

---

## 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 *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/94/index.html>.

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.

### 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.

### Configure the Database

During deployment of SAS Business Rules Manager on SAS 9.4, the SAS Deployment Wizard creates and configures the database tables in the SAS Decision Manager Common Data Server database by default. This database uses the PostgreSQL database management system.

You can use a third-party database server with SAS 9.4. In SAS 9.4, Oracle is the only other third-party database management system that is supported for the SAS Decision Manager database. For information about which versions of the alternative databases are supported, see [“Reviewing Third-Party Database Requirements”](#) in *SAS Intelligence Platform: Installation and Configuration Guide*.

The SAS Deployment Wizard prompts you to enter the information that you gathered when you completed the pre-installation tasks for your database. See [“Pre-installation Tasks for SAS Decision Manager Common Data Server”](#) on page 10 and [“Pre-installation Tasks for an Oracle Database”](#) on page 11 for more information.

For database-specific information about configuring a database, see *SAS Intelligence Platform: Installation and Configuration Guide*.

### Create and Load Tables through the SAS Deployment Wizard

If you select **SAS Decision Manager Common Data Server** as the database type, the database tables are automatically created and loaded during the installation and configuration process for the SAS Decision Manager Common Data Server. The default name for the database is **dcmdb**.

If you select **Oracle** as the database type, the **Automatically create tables and load data** check box in SAS Deployment Wizard is enabled. If you want the Oracle database tables to be created and loaded automatically, leave this box selected. If you want to create the tables yourself, then clear the check box, and submit the necessary SQL statements after the wizard finishes running. See [“Create Oracle Database Tables”](#) on page 22 for more information.

## Chapter 4

# Performing Post-installation Tasks

---

<b>Post-installation Configuration and Verification Steps</b> . . . . .	<b>17</b>
<b>Create Users and Assign Permissions</b> . . . . .	<b>18</b>
<b>Creating Operating System Accounts for Product Administrators and Users</b> . . . .	<b>18</b>
About the User Accounts for SAS Business Rules Manager . . . . .	18
Using Operating System Groups to Assign Permissions . . . . .	19
Create Windows Operating System Accounts and Groups for Users . . . . .	19
Create UNIX Operating System Accounts and Groups for Users . . . . .	20
<b>Create Oracle Database Synonyms</b> . . . . .	<b>22</b>
<b>Create Oracle Database Tables</b> . . . . .	<b>22</b>
<b>Verify the Certificate</b> . . . . .	<b>23</b>
<b>Configure Your Deployment for HTTPS</b> . . . . .	<b>23</b>
<b>Configure the Limit for Character Strings</b> . . . . .	<b>24</b>
<b>Review Business Rules Manager Web Properties</b> . . . . .	<b>25</b>
<b>Modify Log File Settings</b> . . . . .	<b>26</b>
Log4j Configuration File . . . . .	26
Logging Priority Levels . . . . .	27
Log Files . . . . .	27
Turn on SQL Logging . . . . .	28
<b>Manage Directories for Business Rules Content</b> . . . . .	<b>28</b>

---

## Post-installation Configuration and Verification Steps

After you install SAS Business Rules Manager using SAS Software Depot, you must perform additional configuration steps before you can use SAS Business Rules Manager.

1. Verify that all installation and configuration steps in the Instructions.html file have been completed. The Instructions.html file is located in `\sasconfigdir\Lev#\Documents\`. Follow the instructions that are provided in this file.
2. [Create application users and assign permissions.](#)
3. (Optional) [Run the scripts to create Oracle synonyms in the database if you do not want to use the default Oracle schema.](#)

4. If you cleared the **Automatically create tables and load data** check box during installation, then you must manually create and load the Oracle database tables for business rules data . For more information, see [“Create Oracle Database Tables” on page 22](#).
5. [Verify that the Certificate Authority certificate is available](#).
6. (Optional) [Configure your deployment to use HTTPS](#).
7. (Optional) Add the Visual Analytics: Data Building and Data Management: Lineage roles to the Decision Manager Users group. See [“Administering Group and Role Membership” on page 48](#) for more information. These roles enable users to run SAS Visual Data Builder and view lineage information for rule flows.
8. (Optional) [Configure SAS Workflow](#).
9. [Review the Business Rules Manager Web properties](#) in SAS Management Console.
10. (Optional) [Modify log file settings](#).

For more information about post-installation tasks, see *SAS Intelligence Platform: Installation and Configuration Guide*.

---

## Create Users and Assign Permissions

The SAS Deployment Wizard does not create application users by default. The SAS Administrator must create users in SAS Management Console with the appropriate group and role permissions. Make sure that all users are granted the appropriate permissions to the SAS Workspace Server. For more information, see [Chapter 7, “Configuring Users, Groups, and Roles,” on page 43](#).

In a Windows environment, each user or group must be granted permission to the **Log on as a batch job** local security policy. This permission is required in order to access functionality in the Data category. For more information, see [“Create Windows Operating System Accounts and Groups for Users” on page 19](#).

In a UNIX environment, all SAS Business Rules Manager users must be part of a group that has the appropriate group permissions. For more information, see [“Create UNIX Operating System Accounts and Groups for Users” on page 20](#).

---

## Creating Operating System Accounts for Product Administrators and Users

### *About the User Accounts for SAS Business Rules Manager*

SAS Business Rules Manager provides two types of user accounts:

#### Product administrator

A SAS Business Rules Manager administrative user is specific to SAS Business Rules Manager. A product administrator account is not the same as a general administrator account, such as the SAS Administrator (sasadm@saspw). These users must have a valid host operating system account, and you must associate that account with a metadata user.

### Users of SAS Business Rules Manager

These users must have a valid host operating system account, and you must associate that account with a metadata user through SAS Management Console.

You must create the operating system account for the administrator and for regular user accounts as a post-installation task. For more information, see the following topics:

- [“Create Windows Operating System Accounts and Groups for Users”](#)
- [“Create UNIX Operating System Accounts and Groups for Users”](#)
- [Chapter 7, “Configuring Users, Groups, and Roles,”](#)

## Using Operating System Groups to Assign Permissions

Users have different operating system privileges on the SAS Workspace Server. By defining a user group for SAS Business Rules Manager, you can assign all users to the same group and grant the same permissions to all users at one time. All SAS Business Rules Manager users must have Read, Write, and Execute permissions for each environment directory that a user is permitted to use. Users also need permissions to all of the files and directories in an environment directory. The operating system must be configured to grant these permissions when new files and directories are created. The steps that you follow to do this depend on which operating system groups are defined and your site’s security policies.

## Create Windows Operating System Accounts and Groups for Users

On the SAS Workspace Server, create an operating system account for the administrator of SAS Business Rules Manager and all SAS Business Rules Manager users.

If the SAS Workspace Server is running in the Windows operating environment, use one of the following methods to create this operating system account:

- If you are using an LDAP server to manage your users, define the user (for example, *domain\username*) on the Active Directory server.
- If you are working on a local machine, complete these steps to create this user account:

1. Right-click the Computer icon on your desktop and select **Manage**. The Computer Management window appears.

*Note:* If you are creating users on a server, you can use the Server Manager.

2. In the left navigation pane, expand the **Local Users and Groups** node. The **Users** and **Groups** nodes appear.
3. Right-click the **Users** node and select **New User**. The New User window appears.
4. In the New User window, complete these tasks:

- Specify a user name and password.

*Note:* In Windows, you cannot enter *<domain>\username* (you enter the user name only), but you must enter *<domain>\username* in the SAS Deployment Wizard and SAS Management Console.

- Clear the **User must change password at next logon** check box.
- Select the **User cannot change password** check box.
- Select the **Password never expires** check box.

Click **Create**.

5. Click **Close** to close the New User dialog box.
6. If you want to add the users that you created to a group, perform the following steps:
  - a. Right-click the **Groups** node in the Computer Management window, and select **New Group**.
  - b. Click **Add**. Enter the user names, separated by semicolons, and click **Check Names**.
  - c. Click **OK**.
7. Assign the security policy of **Log on as batch job** for each user or group.
  - a. Select **Start** ⇒ **Control Panel** ⇒ **System and Security** ⇒ **Administrative Tools** ⇒ **Local Security Policy**.
  - b. From the Local Security Policy window, expand the **Local Policies** node and select **User Rights Assignment**. Then double-click the **Log on as batch job** policy.
  - c. Click **Add user or Group**. Enter the user names or group names, separated by semicolons, and click **Check Names**.
  - d. Click **OK**.

## Create UNIX Operating System Accounts and Groups for Users

### Conditions for the User Group

If you are working in a UNIX operating environment, the following conditions must be met:

- A group of users is created for the UNIX operating environment. The logon IDs for each user must be in this group. The group must also include any user who might run code that is created by SAS Business Rules Manager in a SAS session.
- Users can be members of multiple groups, but the SAS Business Rules Manager group is the primary group for each user.
- The SAS scripts are updated to grant permissions to the SAS Business Rules Manager users on the SAS Workspace Server. For more information, see [“Update the SAS Scripts to Grant Permissions to the User Group” on page 20](#).
- Each environment directory has the correct ownership, and the user group has Read, Write, and Execute permissions.

### Update the SAS Scripts to Grant Permissions to the User Group

Using the **umask** option, you can grant permissions to users on a conditional basis if the user is part of the SAS Business Rules Manager user group.

*Note:* This example might require changes to fit your server configuration. In particular, this example could result in changed permissions on other SAS files, such as OLAP cubes. For example, if you are working with multiple UNIX groups and have a SAS OLAP Server, you must ensure that the account under which the SAS OLAP Server runs has Read and Execute permissions to OLAP files.



To set these permissions:

1. On each SAS Workspace Server, open `/sasconfigdir/Lev1/SASApp/appservercontext_env_usermods.sh`.
2. Enter the configuration information for your operating environment. Here is the general format of this code:

*Note:* The following code uses grave accents and not quotation marks.

```
CMD=<your-operating-system-path>
CURR_GID=`eval $CMD -g`
GID=<solution-group-id>
if [ $CURR_GID -eq $GID ]; then umask 002 fi
```

- 1 In the **CMD=<your-operating-system-path>**, specify the full path on your server where the ID command is stored. You can get this information by entering a **which id** or **whence id** command on your console.
- 2 In the **GID=<solution-group-id>**, specify the group ID. Type **id** on your console to get the GID and UID information.
- 3 A value of 002 is recommended for the **umask** option.

Here are code examples for each UNIX environment where SAS Business Rules Manager is supported:

Operating Environment	Sample Code
AIX	<pre>CMD=/usr/bin/id CURR_GID=`eval \$CMD -g` GID=201 if [ \$CURR_GID -eq \$GID ]; then umask 002 fi</pre>
H64I (HP-Itanium)	<pre>CMD=/usr/bin/id CURR_GID=`eval \$CMD -g` GID=201 if [ \$CURR_GID -eq \$GID ] ; then umask 002 fi</pre>
S64 (Solaris)	<pre>CMD=/usr/xpg4/bin/id CURR_GID=`eval \$CMD -g` GID=201 if [ \$CURR_GID -eq \$GID ] ; then umask 002 fi</pre>
SAX (Solaris for X64)	<pre>CMD=/usr/xpg4/bin/id CURR_GID=`eval \$CMD -g` GID=201 if [ \$CURR_GID -eq \$GID ] ; then umask 002 fi</pre>
LNx (Linux)	<pre>#!/bin/bash CMD=/usr/bin/id CURR_GID=`eval \$CMD -g` GID=500 if [ "\$CURR_GID" -eq "\$GID" ] ; then umask 002 fi</pre>

---

## Create Oracle Database Synonyms

If you use Oracle for your SAS Decision Manager database and you do not want to use the default schema, you can run two SQL scripts to create synonyms for the database tables. These scripts are in

`SASHome\SASDecisionManagerCommonMidTierforDecisionManager\3.1\Config\Deployment\dbscript\Oracle\optional`. Use your preferred Oracle tool to run these scripts. One of these scripts, `brm_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:

1. In the script named `brm_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 Decision 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_grant_priv_synonym.sql`, replace all occurrences of `&usern` with the user ID that you are using to access the database.
3. Run the script named `brm_create_synonym.sql` using your preferred Oracle tool. This script does not require substitution variables.
4. Run `brm_grant_priv_synonym.sql` using your preferred Oracle tool.

---

## Create Oracle Database Tables

When you ran the SAS Deployment Wizard, the **Automatically create tables and load data** check box was selected by default for the SAS Decision Manager database. (See [“Create and Load Tables through the SAS Deployment Wizard”](#) on page 16.) If you cleared the **Automatically create tables and load data** check box and you do not have an existing database instance, you must run the SQL scripts to create and load the tables.

Run the following scripts to create the business rules tables with a compatible database client for your installation. These scripts are located in

`SASHome\SASDecisionManagerCommonMidTierforDecisionManager\3.1\Config\Deployment\dbscript\oracle`.

Before you run these scripts, replace `@schema.name@` in each file with the schema name for your database.

1. `brm_create_table.sql`
2. `brm_create_sequence.sql`
3. `brm_create_constraint.sql`
4. `brm_create_view.sql`

5. brm\_required\_inserts.sql
6. edm\_workflow\_interface\_create\_table.sql
7. edm\_workflow\_interface\_create\_sequence.sql
8. edm\_workflow\_interface\_create\_constraint.sql
9. edm\_workflow\_interface\_create\_trigger.sql
10. edm\_workflow\_interface\_required\_inserts.sql
11. edm\_create\_table.sql
12. edm\_create\_constraint.sql
13. edm\_create\_sequence.sql
14. edm\_required\_inserts.sql
15. dcb\_create\_table.sql
16. dcb\_create\_constraint.sql
17. dcb\_required\_inserts.sql

---

## Verify the Certificate

During installation and configuration of SAS 9.4, the SAS Deployment Wizard enables you to configure the SAS Web Server to use HTTPS and Secure Sockets Layer (SSL) certificates automatically. Verify that the Certificate Authority certificate is available to the trust store for the browser and Java clients such as SAS Workflow Studio and SAS Management Console. For more information, see the Instructions.html file in the directory `\SASConfigDir\Lev1\Documents` and *SAS Intelligence Platform: Middle-Tier Administration Guide* for information about validating your environment.

If you did not use the SAS Deployment Wizard to configure the SAS Web Server to use HTTPS and SSL certificates, you can configure it manually. For more information, see “Configuring SAS Web Server Manually for HTTPS” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

The communication path between SAS Web Server and SAS Web Application Server uses HTTP by default. If you configured the SAS Web Server to use HTTPS by using the SAS Deployment Wizard, you must complete additional steps in order to use HTTPS between SAS Web Server and SAS Web Application Server. For more information, see “Configuring SAS Web Application Server to Use HTTPS” in *SAS Intelligence Platform: Middle-Tier Administration Guide*. If you configure the SAS Web Application Server to use HTTPS, see “Configure Your Deployment for HTTPS” on page 23 for additional instructions.

---

## Configure Your Deployment for HTTPS

The steps listed in “Configuring SAS Web Application Server to Use HTTPS” in *SAS Intelligence Platform: Middle-Tier Administration Guide* change the communication path between the SAS Web Server and SAS Web Application Server to use HTTPS. If you completed those steps and configured the communication path to use HTTPS, then you must also configure the SAS Web Application Server to use HTTPS.

To configure the SAS Web Application Server to use HTTPS:

1. Edit the `sasenv` file for your operating environment.

Environment	File	Directory
UNIX	<code>sasenv.sh</code>	<code>/config-dir/Levn/Web/WebAppServer/SASServer7_1/bin</code>
Windows	<code>sasenv.bat</code>	<code>\config-dir\Levn\Web\WebAppServer\SASServer7_1\bin</code>

Add the following options to the `JVM_OPTS` line:

```
-Dbrm.midtier.use.https=true
-Ddcm.midtier.use.https=true
```

2. In Windows environments, edit the `wrapper.conf` file in the `\config-dir\Levn\Web\WebAppServer\SASServer7_1\conf` directory.

Add the following line to the Java Additional Parameters section:

```
wrapper.java.additional.n=-Ddcm.midtier.use.https=true
```

The number *n* is the next number in the sequence of wrapper parameters.

3. Restart `SASServer7`.

*Note:* The options are needed only on `SASServer7`; they are not needed on `SASServer11`.

---

## Configure the Limit for Character Strings

All business rules action terms that are defined as `String` variables and that are not mapped to input variables are limited to 100 characters unless you define the property `brm.testing.char.length.override` in SAS Management Console.

To change the limit for character strings that are used as action terms in business rules, complete the following steps in SAS Management Console:

1. On the **Plug-ins** tab, select **Application Management** ⇒ **Configuration Manager** ⇒ **SAS Application Infrastructure** ⇒ **Enterprise Decision Manager 3.1**.
2. Right-click **Business Rules Manager Web 3.1** and select **Properties**. The **Business Rules Manager Web 3.1 Properties** dialog box appears.
3. Click the **Advanced** tab, and then click **Add**. The Define New Property dialog box appears.
4. Enter `brm.testing.char.length.override` for the **Property Name**, enter the string length in the **Property Value** field, and click **OK**.

---

## Review Business Rules Manager Web Properties

Review the Business Rules Manager Web properties in SAS Management Console to ensure that the values are appropriate for your environment. Complete the following steps in SAS Management Console:

1. On the **Plug-ins** tab, select **Application Management** ⇒ **Configuration Manager** ⇒ **SAS Application Infrastructure** ⇒ **Enterprise Decision Manager 3.1**.
2. Right-click **Business Rules Manager Web 3.1** and select **Properties**. The Business Rules Manager Web 3.1 Properties dialog box appears.

*Note:* If you are migrating to SAS Business Rules Manager 3.1, SAS Management Console displays the release number that you are migrating from. For example, if you are migrating from SAS Business Rules Manager 2.2, SAS Management Console displays **Business Rules Manager 2.2**.

3. Click the **Settings** tab.

Review the following properties:

### **Location of Code generation macro**

the location of the macro that generates the SAS code for rule sets and rule flows. This property is used only by rule flows that were published using the first maintenance release of SAS Business Rules Manager 1.2. It is not used by rule flows published with later versions.

### **Max row count per table**

the maximum number of rows per rule flow test table. This property applies only to the first maintenance release of SAS Business Rules Manager 1.2 and to SAS Business Rules Manager 2.1 and SAS Decision Manager 2.1. It is not used by later versions.

### **Temporary test code generation directory**

applies only to SAS Business Rules Manager 1.2 and the first maintenance release of SAS Business Rules Manager 1.2. This directory is not used by later versions.

### **Largest allowed uploaded lookup table row count**

the maximum number of rows that can be uploaded for a lookup table. The default is 5000. This value should not exceed 10,000.

### **Maximum Testing Log Length (in lines) of SAS log displayed within User Interface**

the maximum number of lines from the SAS log that are displayed on the SAS log section on the **Results** tab for the rule flow.

### **Support macros in rule expressions**

determines whether macros are allowed in rule expressions. Macros are not supported in decision flows that are deployed by SAS Real-Time Decision Manager.

### **Temporary Location used in Rule Generation**

a temporary directory that SAS Business Rules Manager uses while it generates the SAS code for vocabularies, rule sets, and rule flows. A best practice is to specify a directory that is outside of the SAS configuration directory.

**Test Library Root File System Directory**

the directory where rule flow tests and test results are saved. As users create additional rule flow tests, administrators might need to delete old test results, or ask users to delete old test cases. A best practice is to specify a directory that is outside of the SAS configuration directory.

*Note:* If you are running tests that use data sources that are in a distributed environment such as the Hadoop Distributed File System (HDFS), ensure that the SAS Server file system has enough space to accommodate output data in the form of SAS data sets. If sufficient space is not available, you might need to create subsets of the input data to use for testing.

**Test Metadata Library Root Directory**

the folder in which metadata for rule flow tests is stored.

4. Click the **Advanced** tab.

Review the following properties:

**brm.csvfile.separator**

the character that is used as a separator in CSV files that are used to create lookup tables through the SAS Business Rules Manager interface.

**brm.runtime.codetype**

determines whether SAS Business Rules Manager generates DS2 code or DATA step (DS1) code for rule flows. In many cases, you will get better performance by specifying DS1. However, consider specifying DS2 if your input data is in Teradata, Greenplum, or Hadoop, and you have installed the SAS Code Accelerator. In addition, the LIKE operator is not supported for DS1.

*Note:* This property is not predefined. You can add it as a new property.

**brm.default.application.server.context.override**

the default application server for rule flow testing. You can specify the short logical name such as SASApp. If a value is not specified for this property, SAS Business Rules Manager looks for a server context named SASApp. If that server does not exist, SAS Business Rules Manager uses the last one returned from the metadata server.

*Note:* This property is not predefined. You can add it as a new property.

5. Click **OK** to close the Business Rules Manager Web 3.1 Properties dialog box.

---

## Modify Log File Settings

**Log4j Configuration File**

SAS Business Rules Manager uses log4j to perform logging. When SAS Business Rules Manager starts, the log4j configuration file for the web application is read from `SAS-config-dir\Lev1\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 a configuration file unless you are instructed to do so by SAS Technical Support.

## Logging Priority Levels

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

**Table 4.1** 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

SAS Business Rules Manager writes information to the following log files:

SASBusinessRulesManagerWeb3.1.log

contains messages from SAS Business Rules Manager

SASDecMgrCommon3.1.log

contains messages from the Workflow and Data plug-ins

SASDecMgrShell3.1.log

contains general messages from the Shell

By default, SAS Business Rules Manager writes log files to ***SAS-config-dir\Lev1\Web\Logs\SASServer7\_1\***. You can change the location of these log files 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 *SAS Intelligence Platform: Middle-Tier Administration Guide* for more information about this configuration file.

SAS Business Rules Manager creates new log files each day. For information about logging configurations, see [“Modifying Your Server Logging Configurations”](#) in *SAS Intelligence Platform: System Administration Guide*.

### Turn 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>
```

---

## Manage Directories for Business Rules Content

SAS Business Rules Manager creates two directories for business rules metadata: **Products** and **/System**.

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.

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 no BusinessRuleFlow objects refer to the content.

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.



## Chapter 5

# Performing Migration Tasks

---

<b>About the Migration Process</b> . . . . .	<b>29</b>
<b>Pre-migration Steps</b> . . . . .	<b>29</b>
<b>Post-migration Steps</b> . . . . .	<b>31</b>
Overview of Post-migration Steps . . . . .	31
Copying Rule Flow Test Results . . . . .	32
Migrate from Version 1.2 to 3.1 for Oracle . . . . .	33
Migrate from Version 2.x to 3.1 for Oracle . . . . .	33
Migrate to Version 3.1 for SAS Decision Manager Common Data Server (Single Tier) . . . . .	33
Migrate to Version 3.1 for SAS Decision Manager Common Data Server (Multiple Tiers) . . . . .	35

---

## About the Migration Process

You can migrate from any release of SAS Business Rules Manager to SAS Business Rules Manager 3.1.

The SAS Business Rules Manager migration process supports migrating to a database from the same vendor as the database that you are currently using. Migrating from a database based on Oracle to a database based on PostgreSQL, or vice versa, is not supported.

If you are using Oracle for your database, the migration process assumes that the migrated environment uses the same instance of Oracle. The migration process does not support moving to a different Oracle database server.

---

## Pre-migration Steps

For information about the pre-migration tasks that you must perform, see [“Performing Pre-migration Tasks”](#) in *SAS Intelligence Platform: Migration Guide*. Here are some important steps to help with your migration:

- Back up your SAS system, including servers and desktop clients.
- Back up the your database if you are migrating from SAS Business Rules Manager 2.1 or later to SAS Business Rules Manager 3.1 on a SAS 9.4 system.

- If you are moving to a new system, ensure that the required operating system user accounts that you use for SAS in your current operating system also exist in your new operating system.
- Before you migrate to SAS Business Rules Manager 3.1, you should record the database settings in your current environment. Verify that these settings are entered in the SAS Deployment Wizard when you run the migration.

- If you are migrating from SAS Business Rules Manager 2.1 and you are using the SAS Web Infrastructure Platform Data Server for your database, record the database name and the user ID for the database. The default database name is **brmdb**.

You can find the database name in the `SASCONFIG/Web/WebAppServer/SASServer7_1/conf/server.xml` file on the middle-tier server. Find the resource with the name `sas/jdbc/dcmSharedDataSource`, and look for the value of the `url` attribute. The database name is the text after the final forward slash (/) in the URL. For example, if the attribute is `url="jdbc:postgresql://host:10482/brmdb"`, then the database name is **brmdb**.

- If you are migrating from SAS Business Rules Manager 2.2 or later, record the database name and the user ID for your SAS Decision Manager Common Data Server database. The default database name is **dcmdb**.

You can find the database name in the `SASCONFIG/Web/WebAppServer/SASServer7_1/conf/server.xml` file on the middle-tier server. Find the resource with the name `sas/jdbc/DecisionManagerDS`, and look for the value of the `url` attribute. The database name is the text after the final forward slash (/) in the URL. For example, if the attribute is `url="jdbc:postgresql://host:10482/dcmdb"`, then the database name is **dcmdb**.

- If you are using Oracle for your database, ensure that the Oracle client is installed on your server tier and that there is a matching `tnsnames.ora` file that corresponds to your database.
- If you are using Oracle for your database, record the information that is listed in the following table.

You enter this information in the SAS Decision Manager Database Properties and SAS Decision Manager Database JDBC Properties windows.

**Table 5.1** SAS Deployment Wizard Information for Oracle

Prompt	Description
Host Name	Specifies the fully qualified host name of the server on which the database is installed.
Port	Specifies the port number that is used by the database. The default port for Oracle is 1521.
Directory containing JDBC driver jars	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 to configure SAS Decision Manager database.  See <a href="#">“Verify JDBC Drivers for Oracle” on page 11</a> for more information.

Prompt	Description
<b>Database SID or Service Name</b>	<p>Specifies the Oracle database name. The database name must match either the service name or the Oracle site identifier (SID), both of which can be found in the tnsnames.ora file.</p> <p>If you select <b>Use Oracle database name as a Service Name</b>, then you must enter the service name that is specified in the tnsnames.ora file. For example, if you had the following entry in the tnsnames.ora file, you would enter <code>monitordb</code> in the <b>Database SID or Service Name</b> field:</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><i>Note:</i> In the tnsnames.ora file, the <b>Net Service Name</b> and the <b>Service Name</b> fields must be the same.</p> <p>You can also find the Oracle SID in the tnsnames.ora file. Alternatively, you can run the following query using a database user ID on your Oracle instance:</p> <pre>select instance from v\$thread</pre>
<b>User ID</b>	Specifies the user ID of the database user whose credentials are used to access SAS Business Rules Manager data on the server.
<b>Password</b>	Specifies the password of the user ID whose credentials are used to access SAS Business Rules Manager data on the server.
<b>Schema Pattern</b>	Specifies the schema name for the database. The default schema is the same as the user ID.

- If you are using Oracle for your database, verify that you have the correct JDBC drivers. For more information, see [“Verify JDBC Drivers for Oracle” on page 11](#).
- Use the SAS Migration Utility to create a migration package.

For more information, see [“SAS Migration Utility Reference” in SAS Intelligence Platform: Migration Guide](#).

---

## Post-migration Steps

### Overview of Post-migration Steps

After you have migrated to SAS Business Rules Manager 3.1 on SAS 9.4, perform the following post-migration steps:

1. Copy or move the contents of the rule flow testing directories. See [“Copying Rule Flow Test Results” on page 32](#) for more information.
2. Run the appropriate migration script to migrate your database to SAS Business Rules Manager 3.1. See the following topics for additional information:
  - [“Migrate from Version 1.2 to 3.1 for Oracle” on page 33](#)
  - [“Migrate from Version 2.x to 3.1 for Oracle” on page 33](#)
  - [“Migrate to Version 3.1 for SAS Decision Manager Common Data Server \(Single Tier\)” on page 33](#)
  - [“Migrate to Version 3.1 for SAS Decision Manager Common Data Server \(Multiple Tiers\)” on page 35](#)

*Note:* If you are migrating from SAS Business Rules Manager 3.1 to 3.1 (hardware upgrade) and you are using Oracle for your database, you do not need to run a migration script.

3. (Optional) Configure your deployment to use HTTPS. See [“Configure Your Deployment for HTTPS” on page 23](#) for more information.
4. Update your user group memberships, authorization, roles, and capabilities as needed. See [“Updating Groups and Roles for SAS Business Rules Manager 3.1” on page 51](#) for more information.
5. (Optional) Add the Visual Analytics: Data Building and Data Management: Lineage roles to the Decision Manager Users group. See [“Administering Group and Role Membership” on page 48](#) for more information. These roles enable users to run SAS Visual Data Builder and view lineage information for rule flows.
6. Review the configuration properties in SAS Management Console. Ensure that the values are appropriate for the new environment. See [“Review Business Rules Manager Web Properties” on page 25](#) for more information.
7. (Optional) If you are migrating from SAS Business Rules Manager 2.1, perform post-installation configuration and verification steps for SAS Workflow. For more information, see [“Configuring SAS Workflow for Use with SAS Business Rules Manager” on page 53](#).

## Copying Rule Flow Test Results

Rule flow testing results are not automatically migrated. For any results that you want to access with SAS Business Rules Manager 3.1:

1. Copy the test results from your old deployment to the new deployment. The directory path for rule flow test results is specified by the Business Rules Manager Web 3.1 property **Test Library Root File System Directory** in SAS Management Console. You must use the same directory path in the new deployment. See [“Review Business Rules Manager Web Properties” on page 25](#) for more information.
2. (Optional) If you are running in a UNIX operating environment and the user that needs to access the test results is different from the current user, use the **chown -R** command to change the ownership of the testing folders. For more information, refer to the UNIX man page for the **chown** command.
3. Update the relationship information for the test results so that users can view the test results and re-run the test cases. To update the relationship information, run the following REST service:

```
http://host:port/SASBusinessRulesManagerWeb/rest/RuleflowTestLibMigrate
```

## Migrate from Version 1.2 to 3.1 for Oracle

*Note:* After you have migrated to SAS Business Rules Manager 3.1, you must republish rule flows that were published with SAS Business Rules Manager 1.2 in order for them to be valid in a SAS Data Integration Studio job. The previous version can no longer be referenced through SAS Data Integration Studio, and the rule flow cannot be edited in SAS Business Rules Manager 3.1. However, scheduled jobs will continue to work.

The scripts for migrating an Oracle database are located in the following directory:

```
SASHome/SASDecisionManagerCommonMidTierforDecisionManager/3.1/Config/Deployment/dbscript/oracle/migration/
```

To migrate an Oracle database from SAS Business Rules Manager 1.2 to 3.1:

1. Run the `migration_brm_1.2_to_brm_3.1_sql` migration script. For example, you can run the script with SQL:

```
sqlplus username@tnsname @/install/SASHome/
SASDecisionManagerCommonMidTierforDecisionManager/3.1/Config/Deployment/
dbscript/oracle/migration/migration_brm_1.2_to_brm_3.1.sql schemaName
```

2. Sign in to SAS Business Rules Manager 3.1 and republish rule flows.

## Migrate from Version 2.x to 3.1 for Oracle

The scripts for migrating an Oracle database are located in the following directory:

```
SASHome/SASDecisionManagerCommonMidTierforDecisionManager/3.1/Config/Deployment/dbscript/oracle/migration/
```

To migrate an Oracle database from SAS Business Rules Manager 2.1 or 2.2 to 3.1, run the `migration_brm_version_to_brm_3.1.sql` script for your current release of SAS Business Rules Manager.

For example, you can use SQL\*Plus to run the script to migrate from SAS Business Rules Manager 2.2 to 3.1 as follows:

```
sqlplus username@tnsname @/install/SASHome/
SASDecisionManagerCommonMidTierforDecisionManager/3.1/Config/Deployment/
dbscript/oracle/migration/migration_brm_2.2_to_brm_3.1.sql schemaName
```

## Migrate to Version 3.1 for SAS Decision Manager Common Data Server (Single Tier)

If you are migrating from SAS Business Rules Manager 2.1 or later to 3.1 and you are using the SAS Decision Manager Common Data Server, you must run the database migration script for your operating environment.

1. Verify that the SAS Decision Manager Common Data Server is running on both the source and target machines.
2. Shut down all SAS Web Application Server processes.
3. Run the database migration script for your operating environment. The script for Windows platforms is named `postgres-migration.bat`, and the script for

UNIX platforms is named `postgres-migration.sh`. They are located in the following directory:

```
SASHome/
SASDecisionManagerCommonMidTierforDecisionManager/3.1/
Config/Deployment/Migration/
```

*Note:* This directory contains a README.TXT file that contains information about the parameters for these scripts.

When you run the migration script, substitute the correct values for the release you are migrating from, the server name, port number, and user ID for your database. The script prompts you to enter a password. The syntax for these scripts is as follows:

```
postgres-migration.[bat | sh] SASHome version
source_port source_host source_user source_db_name
target_port target_host target_admin target_user target_db_name
<target_exists>
```

*SASHome*

specifies the SAS Home installation directory location. In Windows operating environments, it is recommended that you enclose this parameter in double quotation marks.

*version*

specifies the version number of your current (source) database from which you are migrating content. Specify **2.1**, **2.2**, or **3.1**.

*source\_port*

specifies the port number of the database from which you are migrating content.

*source\_host*

specifies the host name of the database from which you are migrating content.

*source\_user*

specify the user ID for Decision Manager Common Middle Tier. This value must be a user ID that has access to all of the database content that needs to be migrated. You can find the correct value for this parameter in the `SASCONFIG/Web/WebAppServer/SASServer7_1/conf/server.xml` file on the middle-tier server for the source system. Specify the value of the `user` attribute of the resource with name `sas/jdbc/DecisionManagerDS`.

*source\_db\_name*

specifies the name of the database from which you are migrating content. You can find the correct value for this parameter in the `SASCONFIG/Web/WebAppServer/SASServer7_1/conf/server.xml` file on the middle-tier server for the source system. If you are migrating from SAS Business Rules Manager 2.1, find the resource with the name `sas/jdbc/dcmSharedDataSource`. If you are migrating from SAS Business Rules Manager 2.2, find the resource with the name `sas/jdbc/DecisionManagerDS`. Look for the value of the `url` attribute. The database name is the text after the final forward slash (/) in the URL. For example, if the attribute is `url="jdbc:postgresql://host:10482/dcmdb"`, then specify `dcmdb` for *source\_db\_name*.

*target\_port*

specifies the port number of the database to which you are migrating the content.

*target\_host*

specifies the host name of the database to which you are migrating the content.

*target\_admin*

specifies the user ID of the database administrator for the database to which you are migrating the content. This user ID is used to clean the target database and prepare it for the migrated content.

*target\_user*

specifies a database user ID for the database to which you are migrating the content. This user ID is assigned ownership of the migrated content.

*target\_db\_name*

specifies the database name of the database to which you are migrating the content. The default target database name is **dcmdb**.

*target\_exists*

specifies whether the target database exists. The default value is **YES**. If you have attempted to run this migration script but the migration failed, specify **NO**. Normally, the script creates a backup of the source database. If you specify **NO**, the script does not create an additional backup.

*Note:* This parameter is optional.

### **Migrate to Version 3.1 for SAS Decision Manager Common Data Server (Multiple Tiers)**

If you are migrating from SAS Business Rules Manager 2.1 or later to 3.1 and you are using the SAS Decision Manager Common Data Server, you must run the database migration script for your operating environment. You must run the post-migration script on the server tier. In a multiple-tier environment, you must copy the script from the middle tier to the server tier before you run it.

1. Verify that the SAS Decision Manager Common Data Server is running on both the source and target machines.
2. Shut down all SAS Web Application Server processes.
3. Create a directory on the server-tier machine (the machine that is running that SAS Decision Manager Common Data Server) for the migration scripts. The directory can reside in the SAS Home installation directory (for example, *SASHome/DCMDB\_Migrate*), but that is not a requirement.
4. Copy the **Config** directory on the middle-tier machine at *SASHome/SASDecisionManagerCommonMidTierforDecisionManager/3.1/Config* to the directory that you created on the server-tier machine.
5. Edit the database migration script for your platform that is in the directory that you created in [Step 3](#). Change the value that is specified for the `DCM_DATA_DIR=` parameter to point to the directory that you created in [Step 3](#) (for example, `DCM_DATA_DIR=$SAS_HOME/DCMDB_Migrate`).

The migration scripts are in the **Config/Deployment/Migration** subdirectory. The script for Windows platforms is named **postgres-migration.bat**, and the script for UNIX platforms is named **postgres-migration.sh**. If you specified **DCMDB\_Migrate** for the new directory in [Step 3](#), the script that you need to edit is located in the directory *SASHome/DCMDB\_Migrate/Config/Deployment/Migration/*.

6. Run the migration script on the server-tier machine as described in [Step 3](#) of [“Migrate to Version 3.1 for SAS Decision Manager Common Data Server \(Single Tier\)” on page 33](#). Substitute the correct values for the release that you are migrating from, the server name, port number, and user ID for your database. The script

prompts you to enter passwords for different user accounts. Note the user account that is specified in the prompt to ensure that you enter the correct password.



## Chapter 6

# Performing Upgrade Tasks

---

<b>About the Upgrade Process</b> .....	<b>37</b>
<b>Pre-upgrade Steps</b> .....	<b>38</b>
<b>Run the SAS Deployment Wizard</b> .....	<b>39</b>
<b>Post-upgrade Steps</b> .....	<b>40</b>

---

## About the Upgrade Process

You can upgrade from SAS Business Rules Manager 2.1 or 2.2 to SAS Business Rules Manager 3.1.

When you are upgrading to SAS Business Rules Manager 3.1 on the third maintenance release of SAS 9.4, you must complete additional installation and configuration steps. SAS Business Rules Manager 3.1 is integrated with SAS Lineage. Therefore, two passes of the SAS Deployment Wizard installation and configuration process are required in order to complete the upgrade. During the first pass of the SAS Deployment Wizard, it upgrades your existing deployment. After the upgrade has completed successfully, the configuration stage begins. After the configuration stage has completed, you must run the SAS Deployment Wizard again to install the new product components and to complete the configuration.

The SAS Business Rules Manager upgrade process supports upgrading to a database from the same vendor as the database that you are currently using. Upgrading from a database based on Oracle to a database based on PostgreSQL, or vice versa, is not supported.

If you are using Oracle for your SAS Decision Manager database, the upgrade process assumes that the upgraded environment uses the same instance of Oracle. The upgrade process does not support moving to a different Oracle database server.

## Pre-upgrade Steps

Before you upgrade to SAS Business Rules Manager 3.1, record the database settings in your current environment. You must enter this information in SAS Deployment Wizard.

- If you are upgrading from SAS Business Rules Manager 2.1 and you are using the SAS Web Infrastructure Platform Data Server for your database, record the database name and the user ID for the database. The default database name is **brmdb**.

You can find the database name in the **SASCONFIG/Web/WebAppServer/SASServer7\_1/conf/server.xml** file on the middle-tier server. Find the resource with the name **sas/jdbc/dcmSharedDataSource**, and look for the value of the **url** attribute. The database name is the text after the final forward slash (/) in the URL. For example, if the attribute is **url="jdbc:postgresql://host:10482/brmdb"**, then the database name is **brmdb**.

- If you are upgrading from SAS Business Rules Manager 2.2, record the database name and the user ID for your SAS Decision Manager Common Data Server database. The default database name is **dcmdb**.

You can find the database name in the **SASCONFIG/Web/WebAppServer/SASServer7\_1/conf/server.xml** file on the middle-tier server. Find the resource with the name **sas/jdbc/DecisionManagerDS**, and look for the value of the **url** attribute. The database name is the text after the final forward slash (/) in the URL. For example, if the attribute is **url="jdbc:postgresql://host:10482/dcmdb"**, then the database name is **dcmdb**.

- If you are using Oracle for your database, ensure that the Oracle client is installed on your server tier and that there is a matching **tnsnames.ora** file that corresponds to your database. Record the information in the following table.

You enter this information in the SAS Decision Manager Database Properties and SAS Decision Manager Database JDBC Properties windows.

**Table 6.1** SAS Deployment Wizard Information for Oracle

Prompt	Description
Host Name	Specifies the fully qualified host name of the server on which the database is installed.
Port	Specifies the port number that is used by the database. The default port for Oracle is 1521.
Directory containing JDBC driver jars	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 to configure SAS Decision Manager database.  See <a href="#">“Verify JDBC Drivers for Oracle” on page 11</a> for more information.

Prompt	Description
<b>Database SID or Service Name</b>	<p>Specifies the Oracle database name. The database name must match either the service name or the Oracle site identifier (SID), both of which can be found in the tnsnames.ora file.</p> <p>If you select <b>Use Oracle database name as a Service Name</b>, then you must enter the service name that is specified in the tnsnames.ora file. For example, if you had the following entry in the tnsnames.ora file, you would enter <b>monitordb</b> in the <b>Database SID or Service Name</b> field:</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><i>Note:</i> In the tnsnames.ora file, the <b>Net Service Name</b> and the <b>Service Name</b> fields must be the same.</p> <p>You can also find the Oracle SID in the tnsnames.ora file. Alternatively, you can run the following query using a database user ID on your Oracle instance:</p> <pre>select instance from v\$thread</pre>
<b>User ID</b>	Specifies the user ID of the database user whose credentials are used to access SAS Business Rules Manager data on the server.
<b>Password</b>	Specifies the password of the user ID whose credentials are used to access SAS Business Rules Manager data on the server.
<b>Schema Pattern</b>	Specifies the schema name for the database. The default schema is the same as the user ID.

## Run the SAS Deployment Wizard

Follow the instructions provided in [“Add SAS Products That Require Configuration”](#) in *SAS Intelligence Platform: Installation and Configuration Guide* in order to complete the installation and configuration process for an upgrade.

When you run the SAS Deployment Wizard the second time, the following products are installed:

- SAS Lineage Mid-Tier
- SAS Micro Analytic Service Java Interfaces
- SAS Micro Analytic Service Rest API

*Note:* In a multi-machine environment, the server products are typically installed on the SAS Application Server. The rest of the products are on the SAS Middle-Tier Server.

*Note:* If you are configuring multiple Web Application servers, **SASServer13** is added to the deployment for the additional products, so be aware of the need for additional ports.

When you run the SAS Deployment Wizard the second time, verify the following:

- Verify that you are using the new SAS installation data (SID) file. The SID file for your software (your order) is located in the **sid\_files** directory at the root of your SAS Software Depot or, if you have received media, on the first disk of that installation media.
- Verify that your plan file contains SAS Decision Manager Common products.
- Before the configuration stage begins, make sure that the following SAS Services are started:
  - SAS Metadata Server
  - SAS Web Infrastructure Platform Data Server
  - SAS Decision Manager Common Data Server (if you are not using Oracle for your database)
  - SAS Web Server (httpd - WebServer)
  - SAS Object Spawner
  - SAS JMS Broker
  - SAS Cache Locator
- Verify that the following products are selected for configuration:

Product	Upgrading from Version 2.1	Upgrading from Version 2.2 or Later
SAS Decision Manager Common Data Server	Yes	not applicable
SAS Web Application Server Configuration	Yes	Yes
SAS Lineage Mid-Tier	Yes	Yes
SAS Help Viewer for Mid-Tier Applications	Yes	Yes
SAS Decision Manager Common Mid-Tier for Decision Manager	Yes	not applicable
SAS Micro Analytic Service Rest API	Yes	Yes

---

## Post-upgrade Steps

1. After you complete the upgrade process with the SAS Deployment Wizard, see the **Instructions.html** file. The **Instructions.html** file is located in **\sasconfigdir\Lev#\Documents\**. The default URL is **http://host\_name:port/SASDecisionManager**.

2. If you are upgrading from SAS Business Rules Manager 2.1 and you are using the SAS Decision Manager Common Data Server, run the database migration script for your operating system. See [“Migrate to Version 3.1 for SAS Decision Manager Common Data Server \(Single Tier\)”](#) on page 33 or [“Migrate to Version 3.1 for SAS Decision Manager Common Data Server \(Multiple Tiers\)”](#) on page 35 for more information. (You do not need to run a migration script if you are using Oracle or if you are upgrading from SAS Business Rules Manager 2.2.)
3. (Optional) Configure your deployment to use HTTPS. See [“Configure Your Deployment for HTTPS”](#) on page 23 for more information.
4. Update your user group memberships, authorization, roles, and capabilities as needed. See [“Updating Groups and Roles for SAS Business Rules Manager 3.1”](#) on page 51 for more information.
5. (Optional) Add the Visual Analytics: Data Building and Data Management: Lineage roles to the Decision Manager Users group. See [“Administering Group and Role Membership”](#) on page 48 for more information. These roles enable users to run SAS Visual Data Builder and view lineage information for rule flows.
6. Review the configuration properties in SAS Management Console. Ensure that the values are appropriate for the new environment. See [“Review Business Rules Manager Web Properties”](#) on page 25 for more information.
7. (Optional) If you are upgrading from SAS Business Rules Manager 2.1 and are using SAS Workflow, perform the post-installation configuration and verification steps for SAS Workflow. For more information, see [“Configuring SAS Workflow for Use with SAS Business Rules Manager”](#) on page 53.



## Chapter 7

# Configuring Users, Groups, and Roles

---

<b>Security Administration Tasks for SAS Business Rules Manager</b> . . . . .	<b>43</b>
<b>Administering SAS Identities for Users</b> . . . . .	<b>44</b>
Overview of SAS Identities . . . . .	44
Creating SAS Identities . . . . .	44
<b>Groups and Group Membership</b> . . . . .	<b>45</b>
About Groups . . . . .	45
Predefined User Groups in SAS Business Rules Manager . . . . .	45
<b>Roles and Capabilities</b> . . . . .	<b>46</b>
About Roles and Capabilities . . . . .	46
Predefined Roles and Capabilities for SAS Business Rules Manager . . . . .	46
<b>Administering Group and Role Membership</b> . . . . .	<b>48</b>
Viewing Roles and Capabilities in SAS Management Console . . . . .	48
Adding a User to a Group or Role . . . . .	50
Creating New Groups and Roles . . . . .	50
Modifying Roles . . . . .	50
<b>Updating Groups and Roles for SAS Business Rules Manager 3.1</b> . . . . .	<b>51</b>
Update User Group Membership and Authorization . . . . .	51
Update Capabilities for Business Rules Manager Roles . . . . .	52

---

## 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 included here 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*.

## 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 you are entering user IDs for Windows accounts, be sure to qualify the ID (for example, `WIN\myID` or `myID@mycompany.com`).

In a Windows environment, add new users to the **Log on as a batch job** local security policy on the machine that hosts the SAS Workspace Server.

*Note:* Users who log on to SAS Business Rules Manager using an internal account (a user ID that ends in `@saspw`) cannot access all of the features of the application. All users should be assigned external accounts.

The following users are created as part of the SAS Business Rules Manager installation process:

**Table 7.1** Types of Users

User	Description
SAS Administrator	This user has access to all SAS Management Console capabilities and metadata administrative tasks.
SAS Demo User	This user is optional. You can choose to create this user during installation. However, this user is not assigned to a group during installation.

### 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 “User Import Macros” in [SAS Intelligence Platform: Security Administration Guide](#).



---

## 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 User Groups in SAS Business Rules Manager

**Table 7.2** Predefined User Groups

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 Administrators	This group has administrative permissions. Membership in this group is required to administer workflows.  In your initial installation, this group is a member of the following roles: <ul style="list-style-type: none"> <li>Decision Manager Common: Administration</li> <li>Business Rules Manager: All Capabilities</li> </ul>

Group	Description
Decision Manager Users	<p>This group is created during the installation process. Members of this group have permission to read, add, or delete table summary information in the Data category.</p> <p>During configuration, this group was associated with an identity that enables members to access the database during rule flow and table summary execution.</p> <p><i>Note:</i> This group is the only group that is granted permission to publish business rules content to the SAS Content Server by default.</p> <p><i>Note:</i> Unless you make configuration changes, users who do not have administrator permission must be members of this group. If you want these users to be members of a different group, you must grant the group permissions that enable members to do the following:</p> <ul style="list-style-type: none"> <li>• access the database (using the necessary identity)</li> <li>• publish content to the folders for SAS Business Rules Manager on the SAS Content Server</li> </ul>

---

## 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 and Capabilities 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 7.3** Predefined User Roles and Capabilities

Role	Description
Decision Manager Common: Administration	<p>Enables users to perform all Decision Manager Common tasks, including administering workflows.</p> <p>This role is assigned to the group Decision Manager Common Administrators and has the Decision Manager Common: Workflow category capability.</p>
Business Rules Manager: All Capabilities	<p>Enables users to create, edit, and delete all business rules content, including vocabularies, entities, terms, lookup tables, rule sets, and rule flows.</p>
Business Rules Manager: Rule Flow and Rule Set Designer	<p>Enables users to create, edit, and delete rule sets and rule flows.</p>
Business Rules Manager: Rule Flow and Rule Set Read-Only	<p>Enables users to view rule sets and rule flows.</p>
Business Rules Manager: Vocabulary and Lookup Designer	<p>Enables users to create, edit, and delete vocabularies, entities, terms, and lookup tables.</p>
Business Rules Manager: Vocabulary and Lookup Read-Only	<p>Enables users to view vocabularies, entities, terms, and lookup tables.</p>
Comments: Administrator	<p>Enables users to edit or delete comments.</p> <p>The ability to edit and delete comments is controlled by the capabilities under <b>Applications</b> ⇒ <b>SAS Application Infrastructure</b> ⇒ <b>Comments</b> in SAS Management Console.</p>
Data Management: Lineage	<p>Provides default access to the SAS Lineage application. This role is predefined, but it is not automatically added to the Decision Manager Users group. To enable SAS Business Rules Manager users to access SAS Lineage, add the Data Management: Lineage role to the Decision Manager Users group. See “Adjust Group or Role Membership” in <i>SAS Management Console: Guide to Users and Permissions</i> for instructions.</p>

Role	Description
Visual Analytics: Data Building	Enables users to access SAS Visual Data Builder. This role is predefined, but it is not automatically added to the Decision Manager Users group. To enable SAS Business Rules Manager users to access SAS Visual Data Builder, add the Visual Analytics: Data Building role to the Decision Manager Users group. See <a href="#">“Adjust Group or Role Membership”</a> in <i>SAS Management Console: Guide to Users and Permissions</i> for instructions.

---

## Administering Group and Role Membership

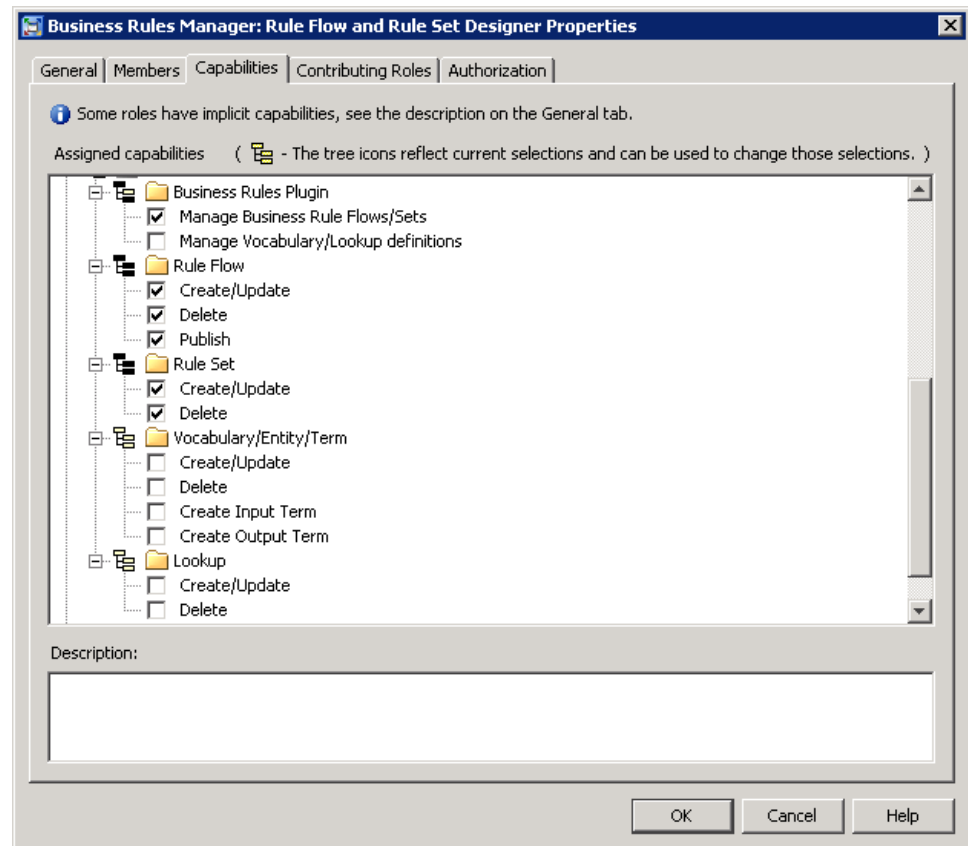
To administer group and role membership, use the User Manager plug-in in SAS Management Console.

### 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: Rule Flow and Rule Set Designer role. These capabilities correspond to the

description of this role in “Predefined Roles and Capabilities for SAS Business Rules Manager” on page 46.



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


The SAS Business Rules Manager capabilities control access to categories in the application. For example, the Rule Sets and Rule Flows categories do not appear when a user signs in to SAS Business Rules Manager if that user is not assigned to either of the following categories:



- Business Rules Manager: Rule Flow and Rule Set Designer
- Business Rules Manager: Rule Flow and Rule Set Read-Only

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 window.

Icon	Description
	None of the capabilities in this category have been specified for this role.

Icon	Description
	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.

### Adding a User to a Group or 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 groups or 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 Administrators group.

For more information, see [SAS Management Console: Guide to Users and Permissions](#).

### Creating New Groups and Roles

The predefined groups and 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 groups or 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](#).

### 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.
- 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](#).

## Updating Groups and Roles for SAS Business Rules Manager 3.1

When you perform a migration or an upgrade, the groups, roles, and capabilities defined in SAS Management Console are preserved in case they have been customized for your site. To take advantage of new roles and capabilities available in SAS Business Rules Manager, you must modify the settings in SAS Management Console.

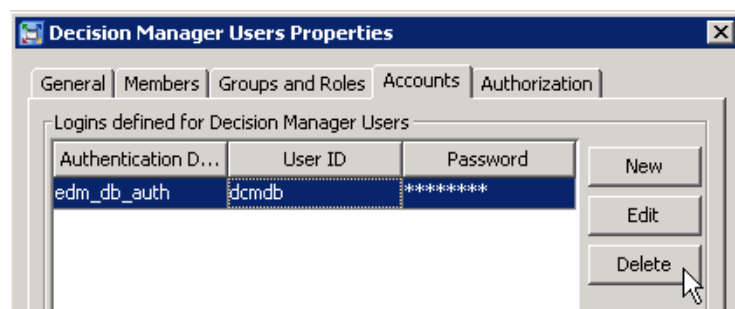
### Update User Group Membership and Authorization

In SAS Business Rules Manager 2.1, all users that needed access to the SAS Business Rules Manager database were assigned to the Business Rules Manager Users group. Beginning with SAS Business Rules Manager 2.2, users are assigned to the Decision Manager Users group. If you migrate or upgrade from SAS Business Rules Manager 2.1 to SAS Business Rules Manager 3.1, you might need to update the membership or authorization for the Business Rules Manager and Decision Manager User groups.

You can continue to use the Business Rules Manager Users group as your primary group, or you can start using the Decision Manager Users group. If you continue to use the Business Rules Manager Users group, you can preserve customized authorization settings.

To continue to use the Business Rules Manager Users group:

1. Ensure that all users that need access to the SAS Business Rules Manager database are assigned to the Business Rules Manager Users group.
2. If you are using a different database instance in the migrated environment, update the authentication domains for migrated user groups. Complete the following steps in SAS Management Console:
  - a. Remove the identity for the authentication domain `edm_db_auth` from the Decision Manager Users group.
    - i. Select the **User Manager** plug-in.
    - ii. Right-click **Decision Manager Users** and select **Properties**.
    - iii. Click the **Accounts** tab. Record the user ID that is specified for the `edm_db_auth` authentication domain. This user ID is the user ID that you need to add to the Business Rules Manager Users group.
    - iv. Select the row for `edm_db_auth` and click **Delete**.



- v. Click **OK**.

- b. Update the identity for the authentication domain `edm_db_auth` for the Business Rules Manager Users group to match the updated login that was added to the Decision Manager Users group.
  - i. Select the **User Manager** plug-in.
  - ii. Right-click **Business Rules Manager Users** and select **Properties**.
  - iii. Click the **Accounts** tab, and click **New**. The New Login Properties dialog box appears.
  - iv. Enter the user ID that you recorded for the `edm_db_auth` domain and the password for this user ID. Select the `edm_db_auth` domain, and click **OK**.
  - v. Click **OK** to save the changes to the Business Rules Manager Users group.

To start using the Decision Manager Users group as your primary group:

1. In SAS Management Console, assign all users that need access to SAS Business Rules Manager to the Decision Manager Users group.
2. Log on to the SAS Content Server, and update the permissions on all of the directories under `sasdav/Products/SAS Business Rules Manager`. Give members of the Decision Manager Users group recursive Read and Write access. The URL for the SAS Content Server is `http://sasserver1_host:port/SASContentServer/dircontents.jsp`.

### Update Capabilities for Business Rules Manager Roles

Beginning with SAS Business Rules Manager 2.2, new capabilities have been added. If you migrate or upgrade to SAS Business Rules Manager 3.1, you must manually add these new capabilities to take advantage of them. The new capabilities are Create Input Term, Create Output Term, and Publish.

The following table lists the roles to which these capabilities are assigned by default.

Role	Capabilities
Business Rules Manager: All Capabilities	Publish, Create Input Term, and Create Output Term
Business Rules Manager: Rule Flow and Rule Set Designer	Publish
Business Rules Manager: Vocabulary and Lookup Designer	Create Input Term and Create Output Term



## Chapter 8

# Configuring SAS Workflow

---

<b>Configuring SAS Workflow for Use with SAS Business Rules Manager . . . . .</b>	<b>53</b>
Overview . . . . .	53
Prerequisites for Using SAS Workflow . . . . .	53
Guidelines for Creating Workflow Definitions . . . . .	54
How to Add the Approval Attribute to a Status . . . . .	54
How to Make Workflow Definitions Available to SAS Business Rules Manager . .	55
Log On to the Server . . . . .	55
Add Tag Attributes to a Workflow Definition . . . . .	56
Upload a Workflow Definition . . . . .	56
Verify That the Workflow Definitions Are Available In SAS Business Rules Manager . . . . .	56
<b>Configure Alert Notifications for SAS Workflow . . . . .</b>	<b>57</b>

---

## Configuring SAS Workflow for Use with SAS Business Rules Manager

### Overview

SAS Workflow provides services that work together to model, automate, integrate, and streamline business processes. It provides a platform for more efficient and productive business solutions.

SAS Workflow is used by SAS solutions that benefit from business process management. SAS Workflow Studio is a desktop client application that is used to design and deploy workflow definitions. The SAS middle tier hosts the workflow engine and the workflow services as part of the SAS Web Infrastructure Platform. SAS Business Rules Manager is used to manage the workflows that are associated with versions. For more information about SAS Workflow, see “[SAS Workflow](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

### Prerequisites for Using SAS Workflow

To use SAS Workflow with SAS Business Rules Manager, be sure the following prerequisites are met:

1. SAS Workflow Engine, SAS Workflow Services, and SAS Workflow Studio must be installed and configured. For more information, see *SAS Intelligence Platform: Installation and Configuration Guide*.

2. Users must be a member of the Decision Manager Common Administrators Group or another user group that is associated with the Decision Manager Common: Administration role. The users or groups must also be assigned to a workflow template management role. For more information, see [“Deploying and Maintaining Workflows”](#) in *SAS Workflow Studio: User's Guide*.
3. Workflow definitions must be created using SAS Workflow Studio. For more information about creating workflow definitions, see [“Defining Workflows with SAS Workflow Studio”](#) in *SAS Workflow Studio: User's Guide*.

### **Guidelines for Creating Workflow Definitions**

When you create workflow definitions in SAS Workflow Studio to use with SAS Business Rules Manager, follow these guidelines:

- Participants, and policies must be added to the task level. Statuses added at the task level and the default statuses at the workflow definition level can be used for a task status. Data objects can be added at the workflow definition level or task level. Users can see only the data objects defined at the task level from the Workflows category in SAS Business Rules Manager.
- Only the Potential Owner and Business Administrator workflow roles are used by SAS Business Rules Manager and they can be used in either a participant or swimlane definition. The Actual Owner workflow role should not be used as part of a workflow definition.
- In order to assign additional participants to tasks in SAS Business Rules Manager, the user must have or be in a group that is assigned the workflow role of Business Administrator. Also, in order to manage workflows and assign participants, the user must be in the Decision Manager Common Administrators group, or in a group that is a member of the Decision Manager Common Administrators group or that is associated with the Decision Manager Common: Administration role in SAS Management Console.

The following groups are created at installation time:

- Decision Manager Common Administrators Group
- Decision Manager Users Group

For more information, see [“Security Administration Tasks for SAS Business Rules Manager”](#) on page 43.

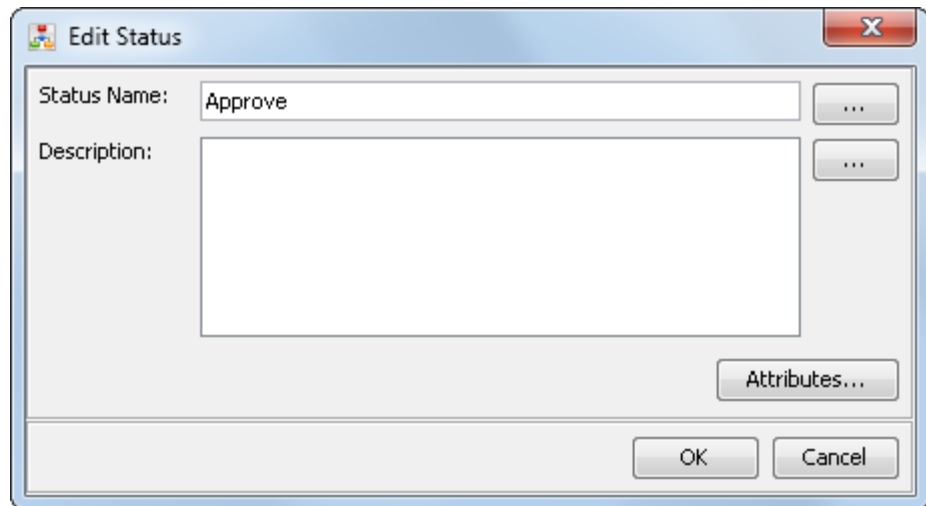
- Only workflow definitions that are activated in the Workflow repository, that are associated with the `mmapi` tag attribute in the file properties, are available to SAS Business Rules Manager.

### **How to Add the Approval Attribute to a Status**

The **Approval** attribute allows a workflow designer to signify that a specific task approves the associated version for a rule flow. This attribute then notifies the users of the version that a rule flow is approved. For business rules the **Approval** attribute must be set so that a workflow can be used to manage rule flows.

To add the **Approval** attribute to a status:

1. Expand the **Statuses** folder in the **Workflow Tree**.
2. Right-click a status and select **Edit**.



3. Click **Attributes**.
4. Click **Add** and enter the following values for the new attribute.

**Key**

Approval

*Note:* This key is case-sensitive.**Value**

true

5. Click **OK** twice to save.

### **How to Make Workflow Definitions Available to SAS Business Rules Manager**

After you have created a workflow definition in the SAS Workflow Studio, you must make the workflow definition available to SAS Business Rules Manager.

To save the workflow definition to the Workflow repository:

1. Save the workflow definition to your local drive.
2. [Log on to the server.](#)
3. [Add the tag attribute](#) of `mmapi` to the workflow definition file properties.
4. [Upload the workflow definition.](#)
5. [Verify that the workflow definition](#) is available in the Workflows category.

For more information, see “[Deploying and Maintaining Workflows](#)” in *SAS Workflow Studio: User's Guide*.

### **Log On to the Server**

With SAS Workflow Studio, you are limited to managing locally stored workflow definitions on your system until you have logged on to the SAS Content Server. After you are connected, you can access additional workflow definitions that are stored in the SAS Content Server.

To log on to the server:

1. Select **Server** ⇒ **Log On**.
2. In the Log On window, select the host-name from the **SAS environment** drop-down list.  
*Note:* For more information, see “Configuring the SAS Environment File” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.
3. Enter a user ID and password, and click **Log On**.
4. Click **OK** if a confirmation message appears.

### **Add Tag Attributes to a Workflow Definition**

Only those workflow definitions in the Workflow repository that contain the **mmapi** tag attribute in the file properties are available to SAS Business Rules Manager.

To add a tag attribute to the file properties of a workflow template in SAS Workflow Studio:

1. Select **File** ⇒ **Properties** and click **Add**.
2. Enter the tag value of **mmapi**.  
*Note:* The file properties are case sensitive. This value must be lowercase.
3. Click **OK** twice.

### **Upload a Workflow Definition**

To upload a workflow:

1. From the **Server** menu, select the **Save to Repository** menu option. The Save to Workflow Repository window appears.
2. (Optional) Enter relevant comments to associate with the workflow definition.
3. Select the **Activate** option if you want to activate the current version in the Workflow repository.
4. Click **OK**.
5. Click **OK** if confirmation messages appear.

### **Verify That the Workflow Definitions Are Available In SAS Business Rules Manager**

To verify that the workflow definitions are available in the Workflows category view of SAS Business Rules Manager:

1. Enter the URL `http://hostname:port/SASDecisionManager` in your web browser.
2. Enter the user ID and password for a user that is in the Decision Manager Common Administrators Group or a user group that is associated with the Decision Manager Common: Administration role.
3. Verify that the uploaded workflow definition is available in the Workflows category view. From the Workflows category view, select **Actions** ⇒ **Set Mappings**. The Set Mappings window appears with a list of the available workflow definitions.

For more information, see “Set Mappings” in *SAS Business Rules Manager: User’s Guide*.

---

## Configure Alert Notifications for SAS Workflow

To enable workflow participants to receive alert notifications from SAS Workflow, you must configure the **E-mail** notification type in SAS Management Console. After you have configured the alert notifications, you can then use the Notify Participant policy and other workflow notification policies for workflow tasks in SAS Workflow Studio. The notifications setting in SAS Management Console is a global setting. Preferences and notifications can also be configured for individual users.

The Send Notification By Data Object policy in SAS Workflow Studio integrates with the SAS Web Infrastructure Platform’s Notification Service. Recipients are notified according to their preferences (e-mail or portlets).

1. Log on to SAS Management Console as an administrator.
2. On the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure**.
3. Right-click **SAS Application Infrastructure** and select **Properties**.
4. Click the **Settings** tab.
5. Select **Notifications** in the left panel. Use the menus or text fields to set the property.
6. Select the **E-mail** notification type.
7. Click **OK**.
8. To apply this setting and make it available, restart the SAS Web Infrastructure Platform Services, SAS Shared Services, and applications using SAS Workflow.

For more information about the notification properties, see “Setting Global Properties for SAS Applications” in *SAS Intelligence Platform: Middle-Tier Administration Guide*. For more information about setting the notification policies for SAS Workflow, see the SAS Workflow Studio Help or *SAS Workflow Studio: User’s Guide*.



## Part 3

---

# Macro Reference

<i>Chapter 9</i>	
<b>Macros for Importing and Exporting</b> .....	<i>61</i>
<i>Chapter 10</i>	
<b>Macros for Running Rule Flows</b> .....	<i>81</i>





## Chapter 9

# Macros for Importing and Exporting

<b>Introduction to the Import and Export Macros</b> . . . . .	<b>61</b>
<b>Dictionary</b> . . . . .	<b>62</b>
%BRM_CREATE_TEMP_TERM . . . . .	62
%BRM_EXPORT_FOLDER . . . . .	63
%BRM_EXPORT_LOOKUP . . . . .	64
%BRM_EXPORT_RULE_FLOW . . . . .	65
%BRM_EXPORT_RULESET . . . . .	66
%BRM_EXPORT_VOCABULARY . . . . .	66
%BRM_IMPORT_FOLDER . . . . .	67
%BRM_IMPORT_LOOKUP . . . . .	69
%BRM_IMPORT_RULE_FLOW . . . . .	70
%BRM_IMPORT_RULESET . . . . .	73
%BRM_IMPORT_VOCABULARY . . . . .	76
%BRM_LOAD_VOCABULARY . . . . .	78

## Introduction to the Import and Export Macros

SAS Business Rules Manager provides the following macros for importing data into the SAS Decision Manager database and exporting data from the 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 SAS Decision Manager database into a CSV file.

### **%BRM\_EXPORT\_RULESET**

exports rule sets from the SAS Decision Manager database into a CSV file.

### **%BRM\_EXPORT\_VOCABULARY**

exports vocabularies from the SAS Decision Manager database into a CSV file.

**%BRM\_IMPORT\_FOLDER**  
imports folder definitions from a CSV file into the SAS Decision Manager database.

**%BRM\_IMPORT\_LOOKUP**  
imports lookup tables from a CSV file into the SAS Decision Manager database.

**%BRM\_IMPORT\_RULE\_FLOW**  
imports rule flows from a CSV file into the SAS Decision Manager database.

**%BRM\_IMPORT\_RULESET**  
imports rule sets from a CSV file into the SAS Decision Manager database.

**%BRM\_IMPORT\_VOCABULARY**  
imports vocabulary terms from a CSV file into the SAS Decision Manager database.

**%BRM\_LOAD\_VOCABULARY**  
loads the vocabulary terms that are defined in the WORK.TERM data set into the SAS Decision Manager database.

*Note:* The same macro can be run simultaneously by multiple users. However, running import macros concurrently is not recommended.

---

## Dictionary

---

### **%BRM\_CREATE\_TEMP\_TERM**

Reads a CSV file or a SAS data set that defines vocabulary terms and produces a SAS data set named WORK.TERM. You can use the WORK.TERM data set as input to the %BRM\_LOAD\_VOCABULARY macro.

**Requirement:** This macro must be run on the server tier.

---

### **Syntax**

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

### **Required Argument**

**DATAFILE=%STR('input\_file')**

specifies either a SAS data set name or the full pathname to a CSV file. Enclose the filename in single quotation marks.

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 SAS Decision Manager database and is displayed in the interface.

**Default** User ID of the user that is logged on to the server and running the macro

## Details

This macro reads a CSV file or SAS data set that defines vocabulary terms and 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 SAS Decision Manager database. See “%BRM\_LOAD\_VOCABULARY” on page 78 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 9.1.

**Table 9.1** Domain Types and Values for Input Terms

Term Data Type	Derived Domain Type	Derived Domain Values
Character	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.

**Requirement:** This macro must be run on the server tier.

**Tip:** You can modify the data values in the exported CSV file, and then re-import the data. However, do not modify the CSV file structure (column or row order).

---

## Syntax

```
%BRM_EXPORT_FOLDER (CSV=%STR(output_filename.CSV)
<, FOLDER_PATH=<%STR(pathname_1<, pathname_2>...)>>);
```

### Required Argument

**CSV=%STR(output\_filename)**  
specifies the full pathname to the CSV file for the exported data.

### Optional Argument

**FOLDER\_PATH= <%STR(pathname\_1 <, pathname\_2> ...)>**  
specifies the full pathname of the business rules folders that you want to export. Use a forward slash to separate folder names. By default, %BRM\_EXPORT\_FOLDER exports all business rules folders. You can explicitly specify that you want to export all folders by entering an empty folder list:

```
FOLDER_PATH=
```

You do not need to specify the FOLDER\_PATH= option unless you want to export a specific folder.

**Example** folder\_path=%str(Applications, Retail/ApprovedLoans)

---



---

## %BRM\_EXPORT\_LOOKUP

Exports the contents of lookup tables into a CSV file. You can modify the CSV file and use it as input to the %BRM\_IMPORT\_LOOKUP macro.

**Requirement:** This macro must be run on the server tier.

**Tip:** You can modify the data values in the exported CSV file, and then re-import the data. However, do not modify the CSV file structure (column or row order).

---

## Syntax

```
%BRM_EXPORT_LOOKUP (CSV=%STR(output_filename.CSV)
<, FOLDER_PATH=%STR(pathname)>
<, LOOKUP=%STR('lookup_table_1'<, 'lookup_table_2'>...)>);
```

### Required Argument

**CSV=%STR(output\_filename)**  
specifies the full pathname to the CSV file for the exported data.

### Optional Arguments

**FOLDER\_PATH=%STR(pathname)**  
specifies the full pathname to the business rules folder from which you want to export lookup tables. Use a forward slash to separate folder names.

If you specify a folder pathname, then only the lookup tables in that folder are exported. For example, if you specify `FOLDER_PATH=%STR(Loans/Retail)`, then only the lookup tables in the folder Loans/Retail are exported. If you specify both `LOOKUP=%STR('CountryCodes', 'ZipCodes')` and `FOLDER_PATH=%STR(Loans/Retail)`, but neither of the specified lookup tables are in the Loans/Retail folder, then no lookup tables are exported.

**LOOKUP=%STR('lookup\_table\_1'<, 'lookup\_table\_2'>...)**

specifies names of the lookup tables that you want to export. Enclose each table name 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.

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

---

## %BRM\_EXPORT\_RULE\_FLOW

Exports rule flows from the SAS Decision Manager database into a CSV file. You can modify the CSV file and use it as input to the %BRM\_IMPORT\_RULE\_FLOW macro.

**Requirement:** This macro must be run on the server tier.

**Tip:** You can modify the data values in the exported CSV file, and then re-import the data. However, do not modify the CSV file structure (column or row order).

---

### Syntax

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

### Required Arguments

**CSV=%STR(output\_filename)**

specifies the full pathname to the CSV file for the exported 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. Separate multiple identification numbers with commas.

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

### Optional Argument

**FOLDER\_PATH=%STR(pathname)**

specifies a business rules folder from which you want to export rule flows. Use a forward slash to separate folder names.

If you specify a folder pathname, then only the rule flows in that folder are exported. For example, if you specify `RULEFLOWS=ALL` and `FOLDER_PATH=%STR(Loans/Retail)`, then only the rule flows in the folder Loans/Retail are exported. If you specify `RULEFLOWS=%STR(10045,10572)` and `FOLDER_PATH=%STR(Loans/Retail)`, but neither of the specified rule flows are in the Loans/Retail folder, then no rule flows are exported.

---

## %BRM\_EXPORT\_RULESET

Exports rule sets from the SAS Decision Manager database into a CSV file. You can modify the CSV file and use it as input to the %BRM\_IMPORT\_RULESET macro.

**Requirement:** This macro must be run on the server tier.

**Tip:** You can modify the data values in the exported CSV file, and then re-import the data. However, do not modify the CSV file structure (column or row order).

---

### Syntax

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

### Required Arguments

**CSV=%STR(output\_filename)**

specifies the full pathname to the CSV file for the exported 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. Separate multiple identification numbers with commas.

**Example** rulesets=%str(10168,10043)

---

### Optional Argument

**FOLDER\_PATH=%STR(pathname)**

specifies the full pathname for the business rules folder from which you want to export rule sets. Use a forward slash to separate folder names.

If you specify a folder pathname, then only the rule sets in that folder are exported. For example, if you specify RULESETS=ALL and FOLDER\_PATH=%STR(Loans/Retail), then only the rule sets in the folder Loans/Retail are exported. If you specify RULESETS=%STR(10045,10572) and FOLDER\_PATH=%STR(Loans/Retail), but neither of the specified rule sets are in the Loans/Retail folder, then no rule sets are exported.

---

## %BRM\_EXPORT\_VOCABULARY

Exports vocabularies from the SAS Decision Manager database into a CSV file. You can modify the CSV file and use it as input to the %BRM\_IMPORT\_VOCABULARY macro.

**Requirement:** This macro must be run on the server tier.

**Tip:** You can modify the data values in the exported CSV file, and then re-import the data. However, do not modify the CSV file structure (column or row order).

---

## Syntax

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

### Required Arguments

**CSV=%STR(output\_filename)**

specifies the full pathname to the CSV file for the exported data.

**VOCAB=ALL | %STR('vocabulary\_1'<, 'vocabulary\_2'>...)**

specifies the names of the 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 names with commas.

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

### Optional Argument

**FOLDER\_PATH=%STR(pathname)**

specifies the full pathname of the business rules folder from which you want to export vocabularies. Use a forward slash to separate folder names.

If you specify a folder pathname, then only the vocabularies in that folder are exported. For example, if you specify VOCAB=ALL and FOLDER\_PATH=%STR(Loans/Retail), then only the vocabularies in the folder Retail are exported. If you specify VOCAB=%STR('loanVocab','riskVocabulary') and FOLDER\_PATH=%STR(Loans/Retail), but neither of the specified vocabularies are in the Retail folder, then no vocabularies are exported.

---

## %BRM\_IMPORT\_FOLDER

Imports the folder definitions in the specified CSV file into the SAS Decision Manager database.

**Requirement:** This macro must be run on the server tier.

---

## Syntax

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

### Required Arguments

**CSV=%STR(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 Folder CSV Input File” on page 68](#).

**REJECT=%STR(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 SAS Decision Manager database. See [“Using the %BRM\\_IMPORT\\_FOLDER Macro” on page 68](#) 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 SAS Decision Manager database and is displayed in the interface.

**Default** User ID of the user that is logged on to the server and 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 pathname 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 create a folder named Applications and to specify a blank column for the folder description, specify the following in the CSV file:

```
Applications,,N,Loans/Retail
```

**Table 9.2** 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 <b>Y</b> or <b>N</b> .	No
FOLDER_PATH	The pathname to the business rules folder where you want to import the contents of the CSV file. This pathname must exist. Separate folder names with forward slashes.	No



---

## %BRM\_IMPORT\_LOOKUP

Imports lookup tables from the specified CSV file into the SAS Decision Manager database.

**Requirement:** This macro must be run on the server tier.

---

### Syntax

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

### Required Arguments

**CSV=%STR(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 Lookup CSV Input File” on page 70](#).

**REJECT=%STR(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 SAS Decision Manager database. See [“Using the %BRM\\_IMPORT\\_LOOKUP Macro” on page 69](#) 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 SAS Decision Manager database and is displayed in the interface.

**Default** User ID of the user that is logged on to the server and running the macro

---

### 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 pathname to determine whether a lookup table already exists. If the lookup table already exists, then it is updated. If the pathname exists but the lookup table does not exist, the lookup table is created. If the pathname 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.

### Format of the Lookup CSV Input File

The first row of the input file must be a header row. SAS Business Rules Manager begins importing data with row two.

Each subsequent 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. The following example specifies the keys **AU** and **CA** and associates them with the values **Australia** and **Canada**, respectively. These key-value pairs will be imported into the lookup table **Country\_Codes**.

```
FOLDER_PATH,LOOKUP_NM,DESCRIPTION,NAME,VALUE
Loans/Retail,Country_Codes,,AU,Australia
Loans/Retail,Country_Codes,,CA,Canada
```

This input file would appear in Microsoft Excel as shown below.

	A	B	C	D	E
1	FOLDER_PATH	LOOKUP_NM	DESCRIPTION	NAME	VALUE
2	Loans/Retail	Country_Codes		AU	Australia
3	Loans/Retail	Country_Codes		CA	Canada

*Note:* When you import a lookup table with the %BRM\_IMPORT\_LOOKUP macro, the first line of the input file must be a header row.

**Table 9.3** 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 pathname 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\_RULE\_FLOW

Imports rule flows from the specified CSV file into the SAS Decision Manager database.

**Requirement:** This macro must be run on the server tier.

---

## Syntax

```
%BRM_IMPORT_RULE_FLOW (CSV=%STR(input_filename.CSV),
REJECT=%STR(reject_filename.CSV)<, BRM_USER=user_ID><, OVERWRITE=Y|
N>;
```

### Required Arguments

**CSV=%STR(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 71](#).

**REJECT=%STR(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 SAS Decision Manager database. See [“Using the %BRM\\_IMPORT\\_RULE\\_FLOW Macro” on page 71](#) 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 SAS Decision Manager database and is displayed in the interface.

**Default** User ID of the user that is logged on to the server and running the macro

**OVERWRITE=Y|N**

specifies whether existing rule flows can be updated. If you specify **N**, the updates are rejected.

## 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 pathname to determine whether a rule flow already exists. If the rule flow name and pathname already exist, then the rule flow is updated (unless OVERWRITE=N is specified when the macro is invoked). If the rule flow pathname exists but the rule flow name does not exist, the rule flow is created. If the rule flow pathname 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.

### Format of the Rule Flow CSV Input File

Each row of the CSV input file identifies a rule set, and a rule flow 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 position 1 in the main section of the rule flow named assignRisk in the **Retail/Loans** folder, you can specify the following in the CSV file:

```
. , assignRisk, , Y, main, Y, Loans/Retail, RuleSet1, Loans/Retail, Loan_Vocab, , 1
```

**Table 9.4** 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
RULE_FIRED_OUTPUT_FLG	Specifies whether to create output only for records that fire rules. Specify <b>Y</b> or <b>N</b> . For some types of applications, only the output records for which at least one rule has fired are of interest. Limiting output is useful for applications that detect outliers, such as applications that detect fraud.	No
RULE_SET_SECTION_CODE	The section of the rule flow to which the rule set that is specified in RULE_SET_NM belongs. Specify <b>init</b> , <b>groupstart</b> , <b>main</b> , <b>groupend</b> , or <b>final</b> .  The codes <b>groupstart</b> and <b>groupend</b> 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 <i>SAS Business Rules Manager: User’s Guide</i> for more information.	No
INCLUDE_NODE_OBJECT_FLG	Specifies whether the rule set specified in the RULE_SET_NM field is run when the rule flow executes. Specify <b>Y</b> or <b>N</b> . Selectively running certain rule sets is useful during rule flow development and testing.	No
RULE_FLOW_PATH	The pathname to the business rules folder for the rule flow. This pathname 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 for 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

Column	Description	Can Column Be Blank
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 <i>SAS Business Rules Manager: User’s Guide</i> for more information.</p>	Yes
ORDER	<p>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.</p>	No

## %BRM\_IMPORT\_RULESET

Imports rule sets from the specified CSV file into the SAS Decision Manager database.

**Requirements:** The vocabulary used by the rule set must exist.  
This macro must be run on the server tier.

**CAUTION:** **If the server is running in lockdown mode, you should not edit input files that are generated by the %BRM\_EXPORT\_RULESET macro before you import the data.** If the input file contains errors, the database might become corrupted. See “Locked-Down Servers” in *SAS Intelligence Platform: Security Administration Guide* and “LOCKDOWN Statement” in *SAS Intelligence Platform: Application Server Administration Guide* for more information.

### Syntax

```
%BRM_IMPORT_RULESET (CSV=%STR(input_filename.CSV),
REJECT=%STR(reject_filename.CSV)<, BRM_USER=user_ID>
<, OVERWRITE=Y|N><, LOCK=Y|N>);
```

### Required Arguments

**CSV=%STR(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 74.

**REJECT=%STR(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 SAS Decision Manager database. See “Using the %BRM\_IMPORT\_RULESET Macro” on page 74 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 SAS Decision Manager database and is displayed in the interface.

**Default** User ID of the user that is logged on to the server and running the macro

---

### **LOCK=Y|N**

specifies whether to lock the imported rule set.

**Default** N

---

### **OVERWRITE=Y|N**

specifies whether existing rule sets can be updated. If you specify **N**, the updates are rejected.

**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 pathname to determine whether a rule set already exists. If the rule set pathname and name already exist, then the rule set is updated. If the rule set pathname exists but the rule set name does not exist, the rule set is created. If the rule set pathname 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.

Rule sets that you import with the %BRM\_IMPORT\_RULESET macro are imported as unlocked versions. Before you can publish rule flows that contain the imported rule sets, you must lock the rule sets.

### **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, the following two lines add a rule to the rule set named riskSet, which uses the Loan\_Vocab vocabulary. The first line adds the condition term **CondTerm** and assigns to it the expression <5000. The second line adds the action term **ActionTerm** and assigns to it the expression 'Bad'.

```
., riskSet, , Loan_Vocab, Loans/Retail, RuleName1, , 1, if, Y, CondTerm, <5000, 1, CONDITION
```

```
.,riskSet,,Loan_Vocab,Loans/Retail,RuleName1,,1,if,Y,ActionTerm,'Bad',1,ACTION
```

**Table 9.5** 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 pathname 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 <b>if</b> , <b>elseif</b> , or <b>or</b> . The first rule in a rule set must use the <b>if</b> operator. For information about these operators, see <a href="#">“Controlling Which Conditions Are Evaluated”</a> in <i>SAS Business Rules Manager: User’s Guide</i> .	No
RECORD_RULE_FIRED_FLG	Specifies whether a rule-fired record is created when the condition for the rule identified in the RULE_NM field evaluates to <b>True</b> . Specify <b>Y</b> or <b>N</b> . If you specify <b>N</b> , a rule-fired record is not created regardless of what the condition evaluates to.	No
LHS_TERM	The term for the expression specified in the EXPRESSION column. Terms that are specified in the LHS_TERM column are the condition and action terms for the rule set. In the rule set editor, these terms appear in the Term column in the list view. They appear as column headings in the horizontal view and as row headings in the vertical view.	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 rule set editor. Enclose character strings in quotation marks. To specify a missing expression, enter a period and an underscore (. _). See “Defining New Rules in the Rule Set” in <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 <b>condition</b> or <b>action</b> .	No

---

## %BRM\_IMPORT\_VOCABULARY

Imports vocabulary terms from the specified CSV file into the SAS Decision Manager database.

**Requirement:** This macro must be run on the server tier.

---

### Syntax

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

### Required Arguments

**CSV=%STR(input\_filename)**

specifies the full pathname to the CSV file that defines the vocabulary that you want to import. For more information, see “Format of the Vocabulary CSV Input File” on page 77.

**REJECT=%STR(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 SAS Decision Manager database. See “Using the %BRM\_IMPORT\_VOCABULARY Macro” on page 77 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 SAS Decision Manager database and is displayed in the interface.



**Default** User ID of the user that is logged on to the server and running the macro

## 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.

### 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, the following lines add two terms to the Loan\_Vocab vocabulary. The first term is named Priority, and it is an integer with domain values in the range 1–10. The second term is named RiskCategory, and it is a character string with domain values 'Low' and 'High'.

```
Loan_Vocab,,ApplicationEnt,,Priority,,Integer,discrete,(1-10),N,N,Loans/Retail
Loan_Vocab,,ApplicationEnt,,RiskCategory,,Character,discrete,(Low;High),N,N,Loans/Retail
```

**Table 9.6** 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
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 <b>Character</b> , <b>Decimal</b> , <b>Integer</b> , <b>Boolean</b> , <b>Date</b> , or <b>Datetime</b> .	No

Column	Description	Can Column Be Blank
VOCAB_TERM_DOMAIN_TYPE_TXT	The domain type for the term. Specify <b>discrete</b> , <b>continuous</b> , or <b>Boolean</b> . A domain value is discrete if it is just an individual value such as <b>5.3</b> or <b>18JUL2012:10:25:00</b> . A domain value is continuous if it specifies a range such as <b>&gt;5</b> or <b>&lt;18JUL2012:10:25:00</b> . Terms that are assigned the data type <b>Character</b> 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 <a href="#">“Specify Domain Values”</a> in <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 <b>Y</b> or <b>N</b> .	No
VOCAB_TERM_OUTPUT_EXCLUDE_FLG	Specifies whether to exclude the term from the output data sets created by rule flows. Specify <b>Y</b> or <b>N</b> .	No
FOLDER_PATH	The pathname to the business rules folder for the rule flow. This pathname must exist. Separate folder names with forward slashes.	No

## %BRM\_LOAD\_VOCABULARY

Loads the vocabulary terms that are defined in the WORK.TERM data set into the SAS Decision Manager database. You can create the WORK.TERM data set by using the %BRM\_CREATE\_TEMP\_TERM macro.

**Requirement:** This macro must be run on the server tier.

### Syntax

```
%BRM_LOAD_VOCABULARY (FOLDER_PATH=%STR(pathname),
VOCAB_NM=%STR(vocabulary_name),
VOCAB_ENTITY_NM=%STR(entity_name)<, BRM_USER=user_ID>);
```

### Required Arguments

**FOLDER\_PATH=%STR(*pathname*)**

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

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

**Example** `folder_path=%str(Loans/Retail/Applications)`

**VOCAB\_NM=%STR(*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=%STR(*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 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 SAS Decision Manager database and is displayed in the interface.

**Default** User ID of the user that is logged on to the server and running the macro

---



## Chapter 10

# Macros for Running Rule Flows

---

<b>Introduction to Macros for Running Rule Flows</b> . . . . .	<b>81</b>
<b>Dictionary</b> . . . . .	<b>81</b>
%BRM_GET_RULE_FLOW_CODE . . . . .	81
%BRM_RULE_FLOW . . . . .	82

---

## Introduction to Macros for Running Rule Flows

SAS Business Rules Manager provides the following macros for running rule flows. These macros must be run on the server tier.

`%BRM_GET_RULE_FLOW_CODE`  
creates a package that contains the SAS code for a rule flow.

`%BRM_RULE_FLOW`  
runs a rule flow.

---

## Dictionary

---

### `%BRM_GET_RULE_FLOW_CODE`

Creates a package that contains the SAS code for a specific rule flow. You can run this rule flow package by using the `%BRM_RULE_FLOW` macro.

**Requirement:** This macro must be run on the server tier.

---

### Syntax

```
%BRM_GET_RULE_FLOW_CODE (RULEFLOW_NAME=%STR(name),
RULEFLOW_SK=number, FOLDER_PATH=%STR(pathname),
FILELOCATION=%STR(package_pathname)<, WEBAUTHDOMAIN=
%STR(domain)>);
```

**Required Arguments****RULEFLOW\_NAME=%STR(*name*)**

specifies the name of the rule flow that you want to export.

**Example** ruleflow\_name=%str(Ruleflow1)

**RULEFLOW\_SK=*number***

specifies the identification number of the rule flow. The identification number is shown in parentheses after the rule flow name on the rule flow **History** page or in the **Properties** section of the **Results** tab on the rule flow **Tests** page.

**Example** ruleflow\_sk=10014

**FOLDER\_PATH=%STR(*pathname*)**

specifies the full pathname to the business rules folder where the rule flow is defined. Separate folder names with forward slashes.

**Example** folder\_path=%str(Claims/Processing)

**FILELOCATION=%STR(*package\_pathname*)**

specifies the full pathname to the file for the DS2 package that is produced by the macro. The pathname must exist.

**Example** filelocation=%str(C:\MgrApprovals\approvalFlow.sas)

**Optional Argument****WEBAUTHDOMAIN=%STR(*domain*)**

specifies the authentication domain for the user ID that is used to access the rule flow.

**Default** DefaultAuth

**Requirement** A password must be defined for the user ID.

---

**%BRM\_RULE\_FLOW**

Runs rule flows. You can use the %BRM\_RULE\_FLOW macro to run packages that were created with the %BRM\_GET\_RULE\_FLOW\_CODE macro.

**Requirement:** This macro must be run on the server tier.

**Syntax**

```
FILENAME fileref "pathname";
%BRM_RULE_FLOW (INPUTTABLE=libref.table_name,
MAPPING=mapfile.mapping,
FILELOCATION=fileref, RULEFIRE=Y | N<, THREADCOUNT=number> );
```

## Required Arguments

**INPUTTABLE=libref.table\_name**

specifies the libref and table name for the input table against which you want to run the rule flow.

**MAPPING=mapfile.mapping**

specifies the file that contains the variable mappings. This file is typically a SAS file. See [“Creating a Mapping Table”](#) on page 84.

**FILELOCATION=fileref**

specifies the fileref for the file that contains the DS2 package code for the rule flow. See [SAS Statements: Reference](#) for information about the FILENAME statement and how to define filerefs.

**RULEFIRE=Y | N | S | D | Q**

specifies whether rule-fired data is recorded when the rule flow is run.

- Y records both summary and detailed rule-fired data.
- N does not record any rule-fired data.
- S records only summary rule-fired data.
- D records only detailed rule-fired data.
- Q collects rule-fired data but does not generate summary or detailed rule-fired tables. The rule-fired data is added to the output table in columns named Rule Fired Count and `_RULEFIREDCOUNTS_1`.

## Optional Arguments

**CODETYPE=DS1|DS2**

determines whether SAS Business Rules Manager generates DS2 code or DATA step (DS1) code for rule flows. In many cases, you will get better performance by specifying DS1. However, consider specifying DS2 if your input data is in Teradata, Greenplum, or Hadoop, and you have installed the SAS Code Accelerator.

Default DS2

**THREADCOUNT=number**

specifies the number of processors that are available for concurrent processing. If the rule flow contains rules in either the INIT or FINAL sections, the value of the THREADCOUNT option is set to 1 when rule flow tests are run. This option is ignored when rule flows are executed in the database.

Default the value of the CPUCOUNT= system option

See [“CPUCOUNT= System Option”](#) in [SAS System Options: Reference](#)

## Details

### Running Rule Flows Dynamically

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

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

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

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

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

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

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

### Creating a Mapping Table

*Note:* You must create a mapping table only if you are invoking the `%BRM_RULE_FLOW` macro in SAS code. In SAS Data Integration Studio and in the SAS Business Rules Manager test interface, the mapping table is created for you.

You must supply a file that maps terms in the rule flow to columns in the input table. You can create this file manually, or you can create and run a rule flow test in SAS Business Rules Manager. The mapping tables that are created when a rule flow test is run are written to the WORK library. The code that produces the mapping table is written to the SAS log.

The mapping table also defines the names and structure of the output table, the rule-fired summary table, the rule-fired details table, and the test information table that are generated by the rule flow. The structure of the rule-fired summary table, rule-fired details table, and test information table is static, and you must define them as shown in [“Example: Creating a Mapping File for a Simple Rule Flow” on page 85.](#)

The number in the data set ID column in the example specifies which table the column that is being defined belongs to. The following table lists the possible values for this column and the default table names that are generated when a rule flow is run in SAS Data Integration Studio.

Data Set ID	Table	Contents	Name Generated by SAS Data Integration Studio
1	Rule-fired details	One row for each time that a rule evaluates to true. There might be multiple entries for the same rule, but each entry has different values for the <code>_recordCorrelationKey</code> and <code>RULE_ACTION_FIRE_ID</code> columns.	DCM_RULE_ACTION_FIRE
2	Test information	A single record that holds aggregate information about the execution of the rule flow.	DCM_DEPLOYMENT_EXECUTION
4	Input	Input data	



Data Set ID	Table	Contents	Name Generated by SAS Data Integration Studio
5	Output	Output data	
6	Rule-fired summary	A summary of how many times each rule fired.	DCM_RULE_FIRE_SUMMARY

*Note:* See “Rule-Fired and Test Information Tables” in *SAS Business Rules Manager: User’s Guide* for information about the columns in these tables.

### **Example: Creating a Mapping File for a Simple Rule Flow**

The following example creates a mapping table that maps terms in the rule flow to an input table with five columns. The column names are EngineSize, Make, Model, MSRP, and Type.

The example assumes that the following librefs have been defined: RULEFIRE, DEPLOY, INDATA, and OUTLIB. It uses the table names listed the following table.

Data Set ID	Table	Libref and Table Name
1	Rule-fired details	RULEFIRE.Details
2	Test information	DEPLOY.ThisRun
4	Input	INDATA.InData
5	Output	OUTLIB.MyResults
6	Rule-fired summary	RULEFIRE.Summary

```

data work.MAPPING;
  attrib table length = $100;
  attrib column length = $100;
  attrib termid length = $100;
  attrib type length = $100;
  attrib datasetid length = $100;
  attrib col_type length = $1;
  attrib col_length length = $5;
  attrib col_format length=$32;
  attrib col_informat length=$32;
  call missing(of _all_);
stop;
run;
proc sql;
  insert into work.MAPPING
  values ('RULEFIRE.Details', 'RULE_ACTION_FIRE_ID', 'RULE_ACTION_FIRE_ID', 'output', '1', 'C', '100', '', '')
  values ('RULEFIRE.Details', 'RULE_SET_SK', 'RULE_SET_SK', 'output', '1', 'N', '8', '', '')
  values ('RULEFIRE.Details', 'RULE_SET_NM', 'RULE_SET_NM', 'output', '1', 'C', '100', '', '')
  values ('RULEFIRE.Details', 'RULE_SK', 'RULE_SK', 'output', '1', 'N', '8', '', '')
  values ('RULEFIRE.Details', 'RULE_NM', 'RULE_NM', 'output', '1', 'C', '100', '', '')
  values ('RULEFIRE.Details', 'DEPLMT_SK', 'DEPLMT_SK', 'output', '1', 'N', '8', '', '')
  values ('RULEFIRE.Details', 'RULE_FLOW_SK', 'RULE_FLOW_SK', 'output', '1', 'N', '8', '', '')

```

```

values ('RULEFIRE.Details','RULE_FLOW_NM','RULE_FLOW_NM','output','1','C','100','','')
values ('RULEFIRE.Details','RULE_FIRE_DTTM','RULE_FIRE_DTTM','output','1','N','8','nldatm.',
'nldatm.')
values ('RULEFIRE.Details','DEPLMT_EXECUTION_ID','DEPLMT_EXECUTION_ID','output','1','C','100','','')
values ('RULEFIRE.Details','ENTITY_PRIMARY_KEY','ENTITY_PRIMARY_KEY','output','1','C','1024','','')
values ('RULEFIRE.Details','TRANSACTION_DTTM','TRANSACTION_DTTM','output','1','N','8','nldatm.',
'nldatm.')
values ('RULEFIRE.Details','_RECORDSEQUENCEKEY','_RECORDSEQUENCEKEY','output','1','N','8','','')
values ('DEPLOY.ThisRun','DEPLMT_SK','DEPLMT_SK','output','2','N','8','','')
values ('DEPLOY.ThisRun','DEPLMT_NM','DEPLMT_NM','output','2','C','100','','')
values ('DEPLOY.ThisRun','TRANSACTION_MODE_CD','TRANSACTION_MODE_CD','output','2','C','20','','')
values ('DEPLOY.ThisRun','RECORDS_PROCESSED_NO','RECORDS_PROCESSED_NO','output','2','N','8','','')
values ('DEPLOY.ThisRun','TEST_FLG','TEST_FLG','output','2','C','1','','')
values ('DEPLOY.ThisRun','START_DTTM','START_DTTM','output','2','N','8','nldatm.', 'nldatm.')
values ('DEPLOY.ThisRun','END_DTTM','END_DTTM','output','2','N','8','nldatm.', 'nldatm.')
values ('RULEFIRE.Summary','RULE_SK','RULE_SK','output','6','N','8','','')
values ('RULEFIRE.Summary','RULE_NM','RULE_NM','output','6','C','100','','')
values ('RULEFIRE.Summary','RULE_SET_SK','RULE_SET_SK','output','6','N','8','','')
values ('RULEFIRE.Summary','RULE_SET_NM','RULE_SET_NM','output','6','C','100','','')
values ('RULEFIRE.Summary','RULE_FLOW_SK','RULE_FLOW_SK','output','6','N','8','','')
values ('RULEFIRE.Summary','RULE_FLOW_NM','RULE_FLOW_NM','output','6','C','100','','')
values ('RULEFIRE.Summary','ruleFiredCount','ruleFiredCount','output','6','N','8','','')
values ('INDATA.InData','EngineSize','EngineSize','input','4','N','8','','')
values ('INDATA.InData','Make','Make','input','4','C','13','','')
values ('INDATA.InData','Model','Model','input','4','C','40','','')
values ('INDATA.InData','MSRP','MSRP','input','4','N','8','','')
values ('INDATA.InData','Type','Type','input','4','C','8','','')
values ('OUTLIB.MyResults','EngineSize','EngineSize','output','5','N','8','','')
values ('OUTLIB.MyResults','Make','Make','output','5','C','13','','')
values ('OUTLIB.MyResults','Model','Model','output','5','C','40','','')
values ('OUTLIB.MyResults','MSRP','MSRP','output','5','N','8','','')
values ('OUTLIB.MyResults','Type','Type','output','5','C','8','','')
;
quit;

```

# Index

---

## A

- access permissions
  - capabilities [46](#)
  - groups [45](#)
  - roles [46](#)
  - SAS identities [44](#)
- accounts
  - capabilities [46](#)
  - roles [46](#)
  - SAS identities [44](#)
- administrators
  - administrative group [45](#)

## C

- capabilities [46](#)
- configuration
  - SAS Workflow [53](#)

## G

- groups [45](#)
  - creating [50](#)
  - Public [45](#)
  - SAS Users [45](#)

## I

- installing SAS Business Rules Manager
  - installing additional products [15](#)
  - pre-installation requirements and steps [9](#)
  - running the SAS Deployment Wizard [15](#)
- Instructions.html file [17](#)

## L

- Log on as a batch job [19](#)

## M

- metadata
  - access to [45](#)

## O

- operating system accounts
  - users [18](#)
- operating system requirements [9](#)

## P

- permissions
  - capabilities [46](#)
  - groups [45](#)
  - roles [46](#)
  - SAS identities [44](#)
- post-installation
  - configuration [17](#)
  - verification [17](#)
- pre-installation requirements and steps [9](#)
- pre-installation steps [9](#)
- pre-installation tasks
  - database [11](#)

## R

- roles [46](#)
  - creating [50](#)
  - modifying [50](#)

## S

- SAS Business Rules Manager
  - features [3](#)
- SAS Deployment Wizard
  - running [15](#)
- SAS identities [44](#)
- SAS Intelligence Platform
  - security administration [43](#)
- security
  - capabilities [46](#)
  - groups [45](#)
  - roles [46](#)
  - SAS identities [44](#)
  - SAS Intelligence Platform [43](#)

**U**

- user accounts
  - create [18](#)
- user IDs
  - SAS identities [44](#)

**W**

- Windows
  - Log on as a batch job [19](#)
- workflow
  - configuring [53](#)



# Gain Greater Insight into Your SAS<sup>®</sup> Software with SAS Books.

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

 [support.sas.com/bookstore](http://support.sas.com/bookstore)  
for additional books and resources.

  
THE POWER TO KNOW.

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

