



THE
POWER
TO KNOW.

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

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

SAS® Business Rules Manager 2.1: User's Guide

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

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

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

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

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

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

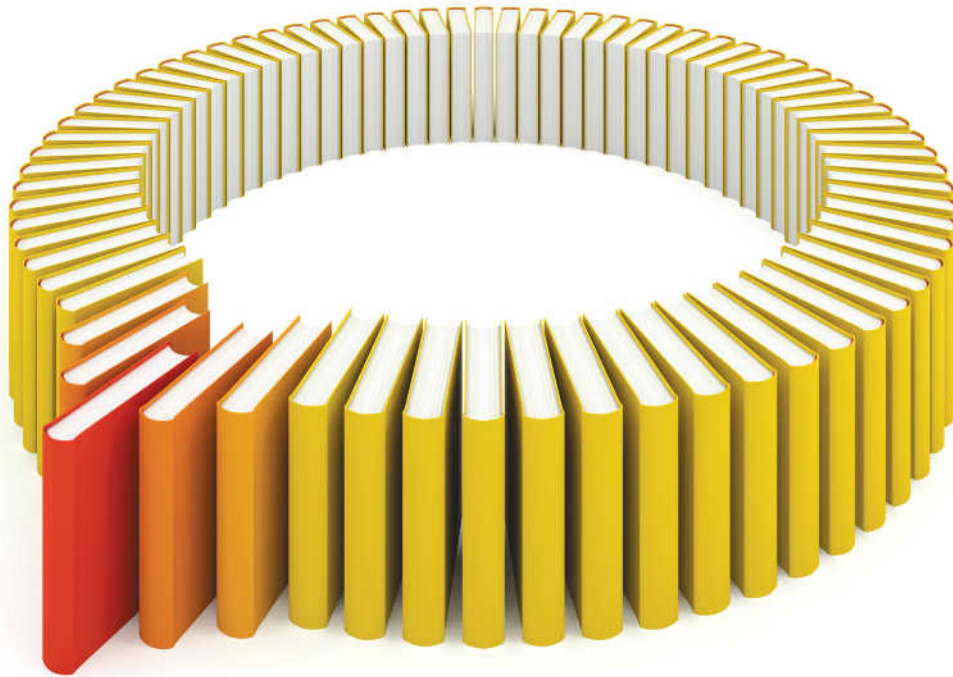
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

November 2013

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

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

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



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

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



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

Contents

<i>What's New in SAS Business Rules Manager 2.1</i>	<i>vii</i>
Chapter 1 • Introduction to SAS Business Rules Manager	1
About SAS Business Rules Manager	1
Accessing SAS Business Rules Manager	2
Exploring the User Interface	3
Managing Preferences	6
Viewing Help and Documentation	7
Create and Publish Business Rules	7
Chapter 2 • Managing Business Rule Folders and Tables	9
Managing Business Rule Folders	9
Managing Data Tables	10
Chapter 3 • Managing Vocabularies	17
Introduction to Vocabularies, Entities, and Terms	17
Tips for Creating Entities and Terms	18
Create a Vocabulary	18
Create New Entities	18
Create New Terms	19
Specify Domain Values	19
Edit Existing Vocabularies, Entities, or Terms	20
Delete Vocabularies, Entities, or Terms	20
Chapter 4 • Using Lookup Tables And Functions	23
About Lookup Tables and Functions	23
Create a New Lookup Table	24
Refresh a Lookup Table	25
Delete a Lookup Table	25
Verify Lookup Keys (Lookup Function)	25
Get Lookup Values (LookupValue Function)	26
Chapter 5 • Managing Rules and Rule Sets	27
About Rules and Expressions	28
About Rule Sets	28
Create a New Rule Set	29
Open an Existing Rule Set	29
Defining New Rules in the Rule Set	29
Validate the Expressions in a Rule Set	41
Change the Order of Rules in a Rule Set	41
Copy Rules and Expressions	42
Delete Rules and Expressions	43
Edit the Name and Description of a Rule Set	43
Edit the Name and Description of a Rule	44
Copy a Rule Set	44
Move a Rule Set to a Different Folder	44
Delete a Rule Set	44
Save a Rule Set	45

Chapter 6 • Creating and Publishing Rule Flows	47
Introduction to Rule Flows	47
Simple Rule Flows, Complex Rule Flows, and BY Groups	48
Create a Rule Flow	48
Open an Existing Rule Flow	49
Open Rule Sets from the Rule Flow Editor	49
Add Attachments to a Rule Flow	49
Add Comments to a Rule Flow	50
Change the Order of the Rule Sets	50
View the Terms Used in a Rule Flow	50
Edit Name and Description for a Rule Flow	51
Copy a Rule Flow	51
Move a Rule Flow to a Different Folder	51
Delete a Rule Set from a Rule Flow	51
Delete a Rule Flow	52
Testing a Rule Flow	52
Publish a Rule Flow	54
Display Publish Information for Rule Flows	54
Chapter 7 • SAS Business Rules Manager Macro Reference	57
About SAS Business Rules Manager Macros	57
Dictionary	57
Appendix 1 • DS2 Functions Supported by SAS Business Rules Manager	71
Glossary	79
Index	81

What's New in SAS Business Rules Manager 2.1

Overview

SAS Business Rules Manager 2.1 includes several new features and enhancements. Highlights include the following:

- integration of SAS Business Rules Manager into the SAS Decision Manager suite of products
- support for authorization
- ability to define and view data tables through SAS Business Rules Manager
- ability to easily run SAS Visual Data Builder to define and create data tables
- ability to generate rule flows from market basket data
- macros for importing and exporting folders and lookup tables

Integration into the SAS Decision Manager Suite

SAS Business Rules Manager has been incorporated into the SAS Decision Manager suite of products. As a result, SAS Business Rules Manager has a new interface with the same look and feel as the other products in the SAS Decision Manager suite. SAS Business Rules Manager now has workspaces and a different set of categories as well as numerous other enhancements to its user interface. The selection path that you take to get to certain features has changed slightly, but the majority of the dialog boxes that you use for individual features have not changed. For more information, see [“Exploring the User Interface” on page 3](#).

Authorization Support

System administrators can use roles to control access to application features in SAS Business Rules Manager. Also, SAS Business Rules Manager defines several default roles when it is installed. For information about the roles and capabilities used by SAS Business Rules Manager, see *SAS Business Rules Manager: Administrator's Guide*.

Ability to Define and View Tables

You can now define tables through SAS Business Rules Manager. You can add tables that have already been registered in metadata, or you can run SAS Visual Data Builder. In addition to other features, SAS Visual Data Builder has query and data import features that enable you to build tables from spreadsheets, delimited files, and SAS data sets.

After you have added a table, you can view table data, view table metadata, and create summaries of the table data.

For more information, see [“Managing Data Tables” on page 10](#).

Ability to Generate Rule Flows from Market Basket Data

SAS Business Rules Manager provides a new macro, %BRM_IMPORT_MARKET_BASKETS, which creates a rule flow using market basket data that was generated with SAS Enterprise Miner.

See *SAS Business Rules Manager: Administrator's Guide* for more information.

Macros for Importing and Exporting Data

SAS Business Rules Manager now provides four new macros for importing and exporting data between SAS Business Rules Manager and comma-delimited (CSV) files.

Macro	Description
%BRM_EXPORT_FOLDER	Exports the objects in a business rules folder to a CSV file.
%BRM_EXPORT_LOOKUP	Exports a lookup table to a CSV file.
%BRM_IMPORT_FOLDER	Imports the data in a CSV file into a business rules folder.
%BRM_IMPORT_LOOKUP	Imports the data in a CSV file into a lookup table.

See *SAS Business Rules Manager: Administrator's Guide* for more information.

Ability to View Publish Information for Rule Flows

You can now view information about each of the published versions of a rule flow. The information includes the folder path to which the rule flow was published and the list of versions that have been published. For each version, SAS Business Rules Manager displays the date on which the version was published and the user ID that was used to publish the rule flow. See [“Display Publish Information for Rule Flows”](#) on page 54 for more information.

Support for SAS Web Infrastructure Platform Data Server

You can choose to use the SAS Web Infrastructure Platform Data Server as a rules database. The SAS Web Infrastructure Platform Data Server is based on PostgreSQL 9.1.9.

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

See *SAS Business Rules Manager: Administrator's Guide* for more information.

Support for Intermediate Terms

When you are defining vocabulary terms, you can now select the new option **Exclude from output**. Terms that are excluded from output are not written to output data sets that are created by rule flows.

When you also select the option **Exclude from input**, the term becomes a temporary term that is available for use only when rule flows are executing.

User Interface Enhancements

There are several enhancements to the user interface that make it easier to create and edit rule sets.

- You can add all of the terms that are in an entity to the decision table at the same time by adding the entity to the decision table in the same way that you add a single term.
- You can add new rules to a rule set by clicking **+** or by clicking in the row next to the icon.
- SAS Business Rules Manager now supports rules that do not have condition expressions. Rules that do not have condition expressions always execute.

x *SAS Business Rules Manager 2.1*

Chapter 1

Introduction to SAS Business Rules Manager

About SAS Business Rules Manager	1
Accessing SAS Business Rules Manager	2
Log On as a Registered User	2
Log Off from SAS Business Rules Manager	2
Exploring the User Interface	3
The Business Rules Category Pane	3
The Rule Set Tile	4
Control the View in a Category Pane	5
Select Pages by Using the Page Manager	5
Managing Preferences	6
Set Global Preferences	6
SAS Business Rules Manager Preferences	6
Viewing Help and Documentation	7
Create and Publish Business Rules	7

About SAS Business Rules Manager

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

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

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

vocabulary management

A business vocabulary identifies the objects and actors in your business domain. It defines the entities and terms that are the building blocks of business rules. SAS Business Rules Manager enables you to easily create and edit entities and terms. For

individual terms, you can create a list of allowable values, which makes creating rules even easier.

business rule authoring

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

rule set management

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

rule flow authoring and publishing

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

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

Accessing SAS Business Rules Manager

Log On as a Registered User

To log on to SAS Business Rules Manager:

1. In the address bar of your web browser, enter the URL for SAS Business Rules Manager and press Enter. The Log On window appears.

Note: Contact your system administrator if you need the URL for the SAS Business Rules Manager.

2. In the Log On window, enter a user ID and password. Your user ID might be case sensitive, depending on the operating system that is used to host the application server. Your password is case sensitive.
3. Click **Log On**.

Log Off from SAS Business Rules Manager

To log off from SAS Business Rules Manager, click **Log Off** in the upper right corner of the window.

By default, if SAS Business Rules Manager is open for 30 minutes with no activity, then a warning message appears. You are logged off unless you choose to continue the session.

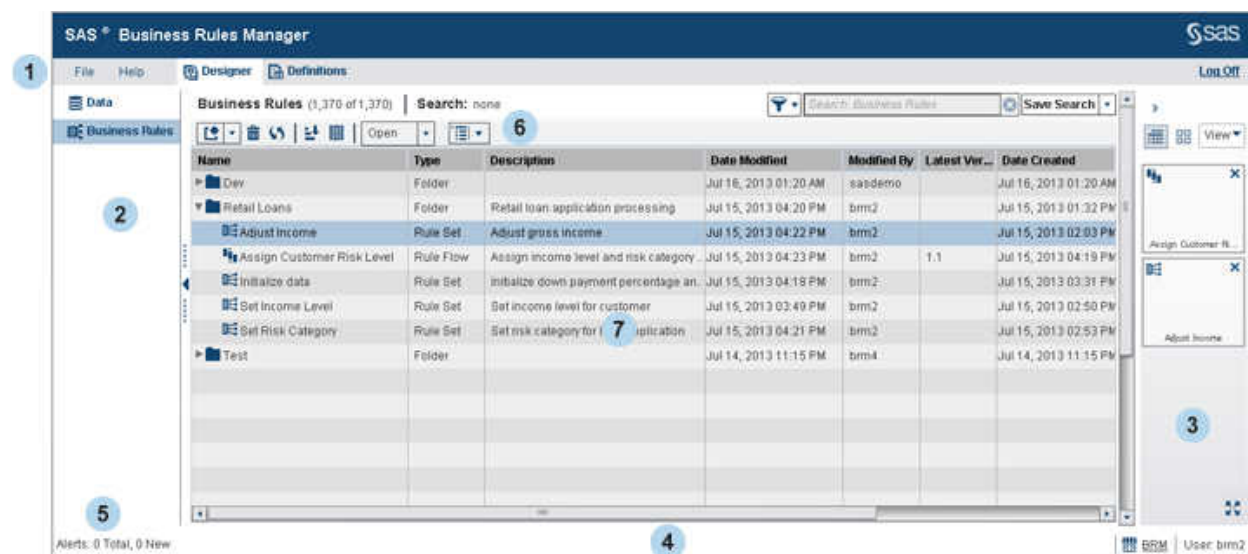
Exploring the User Interface

The SAS Business Rules Manager interface has several category panes and several detail tiles that display information about different types of items in the rules database.

The Business Rules Category Pane

Each of the category panes are similar. The following figure shows the Business Rules category pane.

Figure 1.1 Business Rules Category Pane



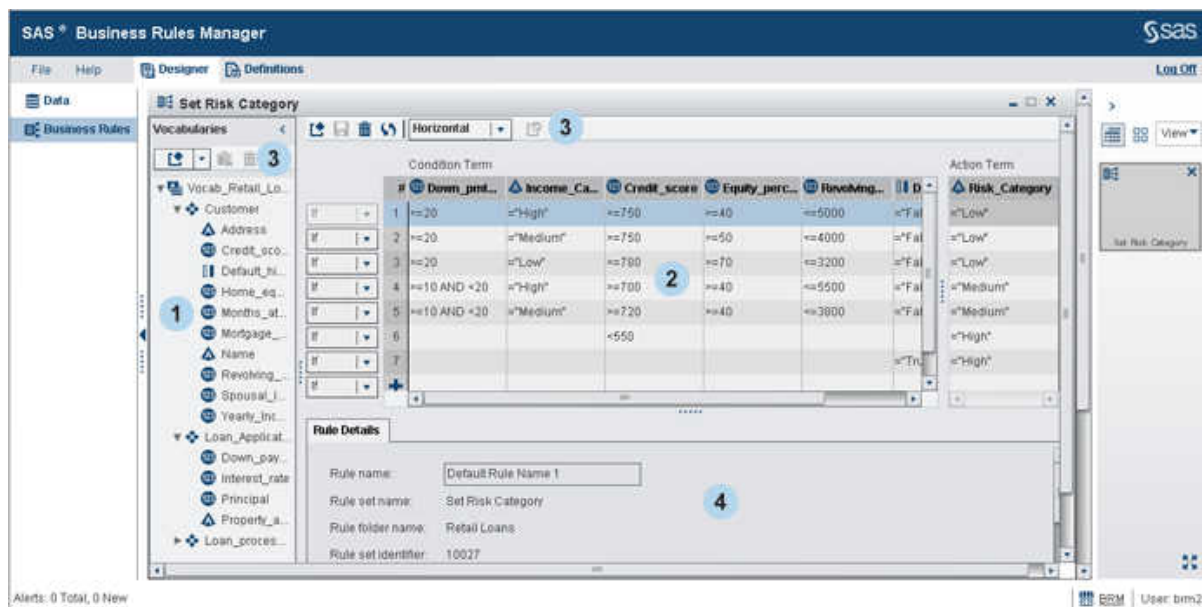
- 1 The menu bar provides options for managing preferences, accessing Help resources, and opening recently edited items.
- 2 The Navigation pane lists the categories of items in the rules database. Click a category to display the list of items in that category.
- 3 The Tile pane displays icons for open items. When you open an item, its icon is automatically added to the tile pane and remains in the tile pane until you click to close the item's tile.
Use the icons in the View toolbar to switch between the most recently viewed category pane and a detail tile. Click to return to the previously displayed category pane. Click to display the most recently viewed detail tile. Click at the far right of the Tile pane to maximize the display of the currently opened objects.
To close all open items, click **View**, and select **Close All**. To open the tiles for all of the minimized items in the tile pane, click **View**, and select **Show All**.
- 4 The status bar displays the user ID or display name of the user that is logged on to the metadata server, the current business context, and alert notices.

- 5 Alerts are displayed when an action that you take has failed. Click **Alerts** to view the list of alerts.
- 6 Panes and tiles can have their own toolbars, so you might see multiple toolbars when the window contains multiple tiles.
- 7 The category pane lists the items that are in the currently selected category. In the category pane, you can open and edit any of the items that are in your rules database. See “[Control the View in a Category Pane](#)” on page 5 for information about controlling the display of information in the category pane.

The Rule Set Tile

Each of the detail tiles is specific to the type of item. For example, when you open a rule set, SAS Business Rules Manager opens a tile that shows the decision table for that rule set.

Figure 1.2 Rule Set Tile (Decision Table)



- 1 The **Vocabulary** pane lists the entities and terms in the vocabulary assigned the rule set. You use this list to add terms to the decision table.
- 2 The decision table contains all of the rules that are defined in the rule set. By using the menu in the toolbar above the decision table, you can choose whether you want to display the decision table in a horizontal format or a vertical format. [Figure 1.2](#) shows the decision table in the horizontal format.




In the horizontal format, each row represents a single rule. Terms are listed across the top of the table. The left side of the table contains the condition expressions, and the right side contains the action expressions.





In the vertical format, each column represents a single rule. Terms are listed on the left side of the table. The top section of the table contains the condition expressions, and the bottom section contains the action expressions.

- 3 There are two toolbars in the rule set tile. Each section has its own toolbar.

- 4 The **Rule Details** tab contains detailed information about the currently selected rule. The **Rule expression** field displays the expressions as they are rendered by SAS Business Rules Manager in the published rule flow.

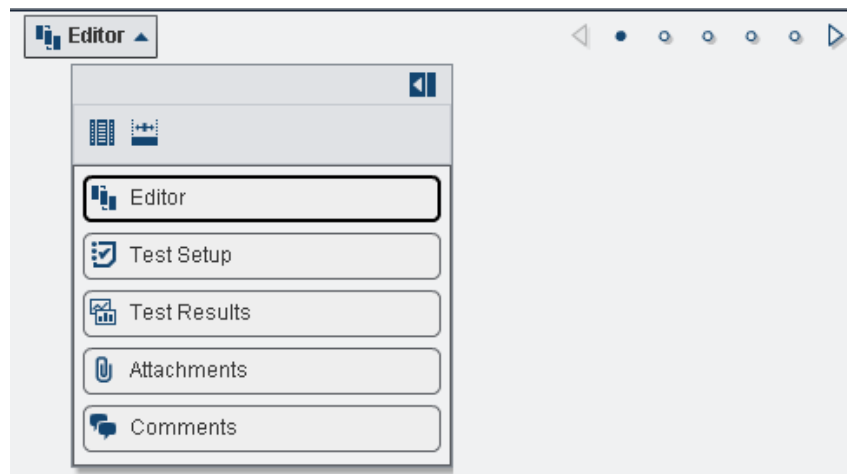
Control the View in a Category Pane

To change which columns are displayed in a category pane, click . SAS Business Rules Manager opens the Manage Columns dialog box. To remove a column from the category pane, select the item in the **Displayed columns** list, and click . To add a column, select the column in the **Available columns** list, and click . Click **OK** to save your changes.


To change how columns are sorted in the category pane, click . SAS Business Rules Manager opens the Sort dialog box. To sort the category pane by one or more columns, select those columns in the **Items to sort** list, and click . SAS Business Rules Manager adds those columns to the **Sort By** list. For each column in the **Sort By** list, select whether you want the column to be sorted in ascending or descending order. The columns will be listed in the category pane in the order in which they are shown in the **Sort By** list. To change the order of a column, select the column, and click  or .


Select Pages by Using the Page Manager


The Page Manager displays the pages that belong to a component. For example, the rule flow editor consists of five pages: Editor, Test Setup, Test Results, Attachments, and Comments. You can navigate between the pages by using the Page Manager or by using the page control dots.




Click the page name to open or close the Page Manager. Select the page that you want to view.

Click  to resize the Page Manager menu to accommodate the longest page name.

You can drag the Page Manager to any location. Click  to return the Page Manager to its dock.

To customize the contents of the Page Manager, click . SAS Business Rules Manager opens the Select Pages dialog box. Click the up and down arrows to change the order of

the pages. Drag and drop the pages or click the left and right arrows to move pages between the **Hidden pages** and **Shown pages** lists.

You can also click the page control dots () to display different pages. If you have already opened a page but have since left the page, you can position the mouse pointer over the page control dot, and SAS Business Rules Manager displays an image of that page. The image includes any changes that you have made to the page.

Managing Preferences

Preferences provide a way for you to customize the user interface. Preferences for each user are stored in metadata and are retained if your deployment is migrated or reconfigured. To open the Preferences window, select **File** ⇒ **Preferences**. There are two general categories of preferences: global preferences and SAS Business Rules Manager preferences.

Set Global Preferences

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

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

User locale

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

Theme

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

Invert application colors

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

Override settings for focus indicator

controls the appearance of the highlighting that surrounds the currently selected field in the SAS Business Rules Manager interface.

SAS Business Rules Manager Preferences

SAS Business Rules Manager preferences apply to SAS Business Rules Manager only.

Open application using this workspace

controls which workspace is displayed by default when you log on to SAS Business Rules Manager.

Show this number of recent items

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

Workspace bar

controls whether the toolbar for the workspace displays icons, text labels, or both..

Viewing Help and Documentation

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

how-to Help

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

embedded Help

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

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

SAS Business Rules Manager: User's Guide

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

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

SAS Business Rules Manager: Administrator's Guide

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

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

Create and Publish Business Rules

To create a rules database:

1. Create business rule folders where you want to save the business rules. See “[Managing Business Rule Folders](#)” for instructions.
2. Create vocabularies. See “[Create a Vocabulary](#)” for more information.
3. Create entities and terms. See “[Create New Entities](#)” on page 18 and “[Create New Terms](#)” on page 19 for more information.
4. Create rule sets and rules. See [Chapter 5, “Managing Rules and Rule Sets,”](#) on page 27 for more information.
5. Create rule flows. See “[Create a Rule Flow](#)” on page 48 for more information.
6. (Optional) Test rule flows. See “[Test a Rule Flow](#)” on page 53 for more information.

7. Publish rule flows. See [“Publish a Rule Flow” on page 54](#) for more information.

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

Chapter 2

Managing Business Rule Folders and Tables


Managing Business Rule Folders	9
Create New Top-Level Folders	9
Create New Subfolders	10
Delete a Folder	10
Managing Data Tables	10
About Managing Data Tables	10
Adding Tables Using SAS Visual Data Builder	10
Add a Table Already Registered in Metadata	11
Edit Table Properties	11
View Table Metadata	11
View Table Data	12
Filter Data in the Table View	13
Add a New Table Summary	13
Add Attachments to a Table	14
Add Comments to a Table	14

Managing Business Rule Folders

Before you add any vocabularies, rule sets, or rule flows, you need to add business rule folders. Your business rule-related content is stored in business rule folders.


Create New Top-Level Folders

To create a new top-level folder:


1. Click , and select **New Top-level Folder**. Alternatively, you can right-click on an existing folder or on an empty area of a category view and select **New Top-Level Folder**.
2. In the New Folder dialog box, enter the name of the new folder. Folder names are case sensitive. SAS Business Rules Manager considers **myfolder** and **MYFOLDER** to be two unique folders.
3. (Optional) Enter a description for the new folder. Descriptions are limited to 256 characters.
4. Click **OK**.

Create New Subfolders

To create a new folder within another folder:

1. In the Category pane, select the parent folder in which you want to create a new subfolder.
2. Click , and select **New Folder**. Alternatively, you can right-click on an existing folder and select **New Folder**.
3. In the New Folder dialog box, enter the name of the new folder. Folder names are case sensitive. SAS Business Rules Manager considers **myfolder** and **MYFOLDER** to be two unique folders.
4. (Optional) Enter a description for the new folder. Descriptions are limited to 256 characters.
5. Click **OK**.

Delete a Folder

A folder must be empty before you can delete it. To delete a folder, right-click on the folder, and select **Delete**. Alternatively, click on the folder and then click .

Managing Data Tables

Adding data tables in SAS Business Rules Manager enables you to view table data, view table metadata, and create table summaries from within SAS Business Rules Manager.

About Managing Data Tables

In SAS Business Rules Manager, you can view your list of tables in either the Designer workspace or the Definitions workspace by selecting the Data category. You can use SAS Visual Data Builder to create new tables and add them to this list. See [“Adding Tables Using SAS Visual Data Builder” on page 10](#) for more information.

Alternatively, if the table is already registered in metadata, you can add the table to the list as described in [“Add a Table Already Registered in Metadata” on page 11](#).

After you have added a table to the list, you can view table data and metadata, create table summaries, and associate attachments and comments with the table.



Adding Tables Using SAS Visual Data Builder

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

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


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

The application provides a series of features that are used in deployments that include SAS LASR Analytic Server. You can load an existing table directly into memory, load the results of a query into memory, or append rows to an in-memory table on a server.

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

Add a Table Already Registered in Metadata

To add a table that has already been registered in metadata, follow these steps:

1. In the Definitions workspace, select the **Data** category, and select **Tables**.
2. Click  to add a table.
3. Select the table that you want to add, and click **OK**.

Edit Table Properties

1. In the Definitions workspace, select the **Data** category, and select **Tables**.
2. Double click the table whose properties you want to edit. Alternatively, you can select the table, and select **Open**. By default, SAS Business Rules Manager displays the Data Source Properties page.
3. Edit the source name and description, or select a different table as the data source.

View Table Metadata

To view table metadata:

1. Select the **Data** category. If you are working in the Definitions workspace, select **Tables**.
2. Double-click the table to open it. Alternatively, you can select the table, and select **Open**.
3. Use the Page Manager to select the Metadata page.

SAS® Business Rules Manager

File Help Designer Definitions Log Off

Data Business Rules

HPI Forecast

Metadata

Data source name: HPI Forecast
Library: LOANS
Engine: BASE
Columns: 8
Description:
Location: /Shared Data/Loans/HPI_FORECAST

Table name: HPI_FORECAST
Type: Table
Date created: Apr 18, 2013 08:37 PM
Rows:

Metadata Columns

Name	Type	Format	Length	Description
NAME	CHAR		32	
survey_date	NUM	DATE9.	8	
ACTUAL	NUM		8	
PREDICT	NUM		8	
LOWER	NUM		8	
UPPER	NUM		8	
ERROR	NUM		8	
STD	NUM		8	

Alerts: 0 Total, 0 New

BRM User: SAS Demo User

View Table Data

To view table data, select the Designer workspace, and select the **Data** category. Double-click the table to open it. SAS Business Rules Manager displays the Table View page.

SAS® Business Rules Manager

File Help Designer Definitions Log Off

Data Business Rules

HPI Forecast

Table View

Columns

- ☒ _NAME_
- ☒ survey_date
- ☒ ACTUAL
- ☒ PREDICT
- ☒ LOWER
- ☒ UPPER
- ☒ ERROR
- ☒ STD

Statistics Column Information

Column: _NAME_
Minimum: CA_Los_Angeles
Maximum: OR_Portland

Value	Frequency
CA_Los_Angeles	1
CA_San_Diego	1
CA_San_Francisco	1

Filter: None

NAME	survey_date	ACTUAL	PREDICT	LOWER	UPPER	ERROR
CA_Los_Angeles	01NOV2013	182.21998084	164.22439108	200.21557061		
CA_San_Diego	01NOV2013	170.8161786	157.34128385	185.12619297		
CA_San_Francisco	01NOV2013	153.03475082	138.71581023	168.41863721		
CO_Denver	01NOV2013	152.49142489	138.13578038	166.8470694		
DC_Washington	01NOV2013	197.32237675	183.87693652	211.48154237		
FL_Miami	01NOV2013	155.9539617	144.15695643	168.45026721		
FL_Tampa	01NOV2013	136.72345586	127.49777155	146.43385516		
IL_Chicago	01NOV2013	114.97615846	102.8535671	128.12446045		
MA_Boston	01NOV2013	153.80523228	145.20890083	162.77230505		
NC_Charlotte	01NOV2013	118.14504222	113.27878474	123.16422361		
NV_Las_Vegas	01NOV2013	114.60231919	87.021197857	142.18344052		
NY_New_York	01NOV2013	159.784623	152.27294405	167.28630195		
OH_Cleveland	01NOV2013	103.36293178	95.618054208	111.10780934		
OR_Portland	01NOV2013	149.20178022	140.32581156	158.48633694		

Alerts: 0 Total, 0 New


BRM User: SAS Demo User

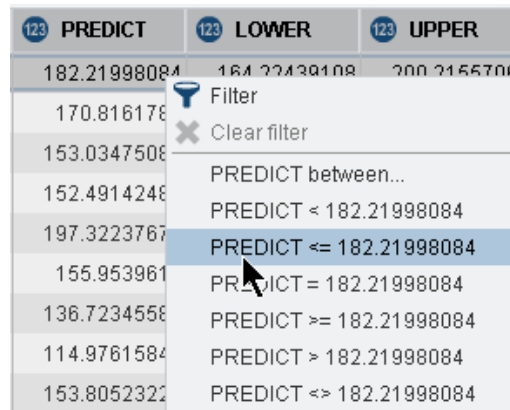
You can choose which columns are shown in the table view by checking the boxes for those columns in the Columns pane. The **Column Information** and **Statistics** tabs display information about the column that is currently selected in the Columns pane.

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

Filter Data in the Table View


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

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



123	PREDICT	123	LOWER	123	UPPER
182.21998084		164.22439108		200.2155706	
170.816178					
153.0347508					
152.4914248					
197.3223767					
155.953961					
136.7234558					
114.9761584					
153.8052322					

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



A	_NAME_	123	survey_da...	123	ACTUAL	123	PREDICT	123	LOWER
	CA_San_Diego		01NOV2013		.		170.8161786		157.34128385
	CA_San_Franci..		01NOV2013		.		153.03475082		138.71581023

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

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

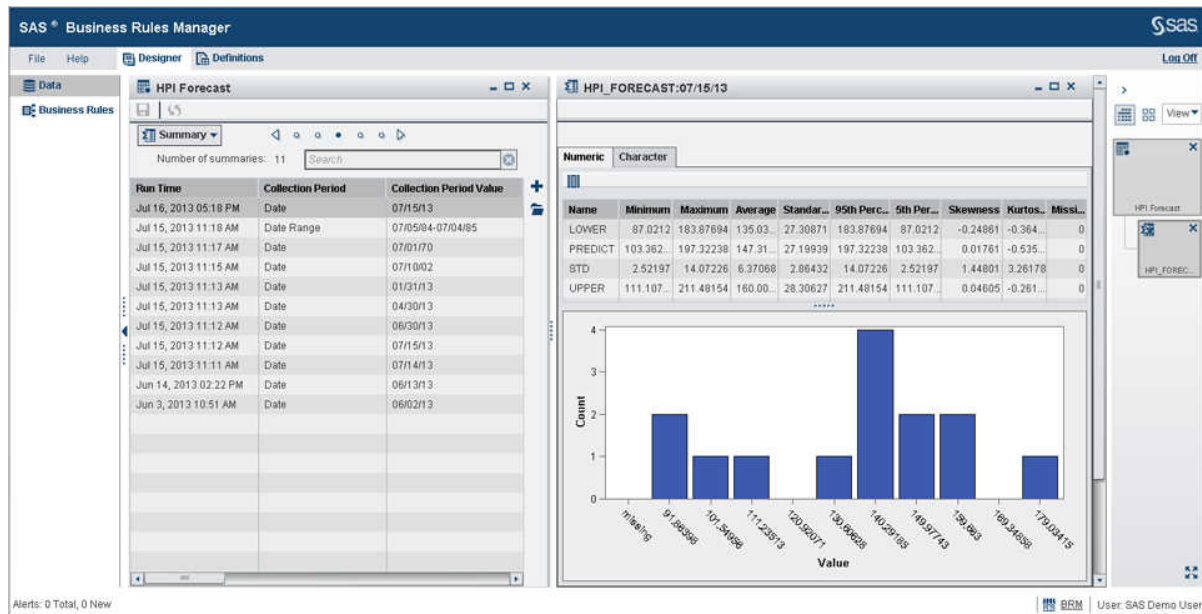
Add a New Table Summary

To add a new table summary:

1. In the Designer workspace, select the **Data** category.
2. Select the table, and select **Open**.
3. Use the Page Manager to select the Summary page.


- Click **+** to add a new summary.
- In the New Summary dialog box, select the **Collection period** and the specific date or time values for the collection period. The summary description is optional.
- Click **Run Now**. SAS Business Rules Manager adds the new summary to the Summary page.
- To view the summary, double-click on the summary in the Summary page.

SAS Business Rules Manager opens the summary in the same layout.



Add Attachments to a Table


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


- Use the Page Manager to select the Attachments page.
- Click **+**, and select the attachment file.
- Click .

Add Comments to a Table

To add a comment:


- Use the Page Manager to select the Comments page.
- Enter a topic title and enter the comment. The topic title is required, and the field for comments does not appear until you enter the topic title.



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

4. Click **Post**.

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

Click  to see comments that have been posted by others.

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

Chapter 3

Managing Vocabularies

Introduction to Vocabularies, Entities, and Terms	17
Tips for Creating Entities and Terms	18
Create a Vocabulary	18
Create New Entities	18
Create New Terms	19
Specify Domain Values	19
Edit Existing Vocabularies, Entities, or Terms	20
Delete Vocabularies, Entities, or Terms	20

Introduction to Vocabularies, Entities, and Terms

Vocabularies, entities, and terms are the basic building blocks of a business rules database. Vocabularies contain entities, and entities contain terms.


Object	Description
Vocabulary	Vocabularies contain one or more business entities. Vocabularies enable you to categorize and structure the entities and terms needed to create a rules database.
Entity	An <i>entity</i> is an object in a business domain. For example, an entity could be Customer, Transaction, or Account. Entities contain terms. They group terms into logical units. Entities are not mapped to tables or to table columns when rules flows are published.
Term	<p>A term is an attribute of an entity. For example, a customer entity might have terms such as name, address, and income. A transaction entity might contain terms for date, time, transaction amount, and account number. Terms are the objects with which you build business rules.</p> <p>A business rule can have <i>condition terms</i> and <i>action terms</i>. Suppose your rule is if balance>1000 then account="premium". The term balance is a condition term, and account is an action term.</p> <p>Terms are mapped to table columns by the applications that use published rule flows that are within metadata.</p>

Tips for Creating Entities and Terms

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


Create a Vocabulary

To create a new vocabulary:

1. In the Definitions workspace, select the **Data** category, and then select **Vocabularies**.
2. Right-click on the folder where you want to create the new vocabulary, and select **New Vocabulary**. Alternatively, you can select the folder, click , and then select **New Vocabulary**.
3. In the New Vocabulary dialog box, enter the name of the new vocabulary. Vocabulary names can contain up to 32 characters and must be unique within an entire rules database. Vocabulary names are case insensitive, and SAS Business Rules Manager considers **name** to be equal to **NAME**.
4. (Optional) Enter a description for the new vocabulary. Descriptions are limited to 256 characters.
5. Click **OK**.

Create New Entities


To create a new entity:

1. Right-click on the vocabulary where you want to create the new entity, and select **New Entity**. Alternatively, you can select the vocabulary, click , and then select **New Entity**.
2. In the New Entity dialog box, enter the name of the new entity. Entity names can contain up to 32 characters and must be unique within a vocabulary. Entity names are case insensitive, and SAS Business Rules Manager considers **name** to be equal to **NAME**.
3. (Optional) Enter a description for the new entity. Descriptions are limited to 256 characters.

4. Click **OK**.

Create New Terms

To create a new term:

1. Right-click on the entity where you want to create the new term, and select **New Term**. Alternatively, you can select the entity, click , and then select **New Term**.
2. In the New Term dialog box, enter the name of the new term. Term names can contain up to 32 characters and must be unique within a vocabulary. Term names are case insensitive, and SAS Business Rules Manager considers **name** to be equal to **NAME**.

Note: You cannot use **_N_** or any DS2 reserved word as a term name. See “Reserved Words in the DS2 Language” in Chapter 19 of *SAS DS2 Language Reference* at <http://support.sas.com/documentation/onlinedoc/base/index.html>.

3. (Optional) Enter a description for the new term. Descriptions are limited to 256 characters.
4. Select the data type for the new term.
5. Select the domain type for the new term.
6. (Optional) Specify the domain values for the new term. Domain values are the set of expected values for a term. See “Specify Domain Values” on page 19 for more information.
7. (Optional) Select **Exclude from input** if you do not want the term to be mapped to a column in an input data set. (The application expects all terms to be mapped to columns in an input data set.)
8. (Optional) Select **Exclude from output** to exclude a term from the output data sets that are generated by rule flows.

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

9. Click **OK**.

Specify Domain Values

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

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

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

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


The following table shows examples of domain values.

Table 3.1 Examples of Domain Values

Data Type	Domain Type	Examples
String	Discrete	high risk;low risk "d'oscail";"d'fhill"
	Continuous	>100;<=myterm
Integer	Discrete	0;800;3500
	Continuous	>100;<=myterm
Decimal	Discrete	3.14;12.98
	Continuous	>1.25;<=N1
Date	Discrete	01jul2012;31jul2012
	Continuous	>='01jan2013'd;<='31dec2013'd
Datetime	Discrete	01jul2012:10:52:00;31jul2012:23:00:00
	Continuous	<='01jul2012:00:00:00'dt;>'31jul2012:23:00:00'dt
Boolean	By default, Boolean values are set to True and False and cannot be changed.	

Edit Existing Vocabularies, Entities, or Terms


To edit an existing object:

1. Right-click on the object that you want to edit, and select **Open**. Alternatively, you can select the object, and then click **Open** in the toolbar.
2. Edit the object's properties as needed.
3. Click .

Delete Vocabularies, Entities, or Terms

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

You can delete an object in one of two ways:

- Right-click on the object, and select **Delete**.
- Select the object, and click .

Chapter 4

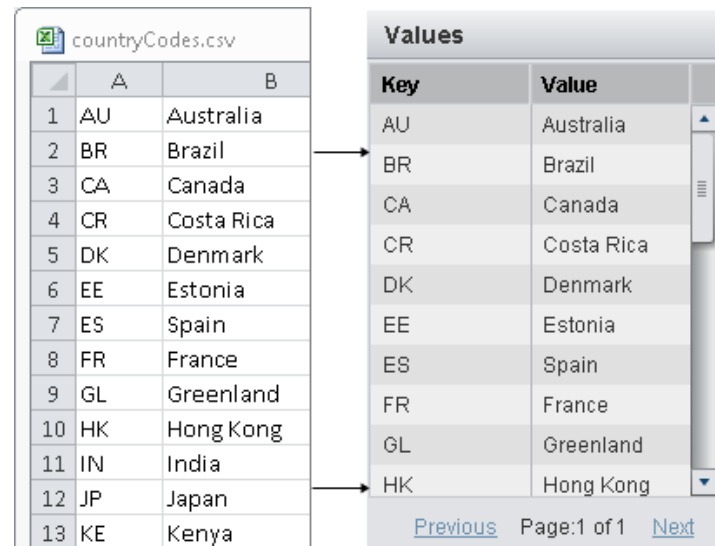
Using Lookup Tables And Functions

About Lookup Tables and Functions	23
Create a New Lookup Table	24
Refresh a Lookup Table	25
Delete a Lookup Table	25
Verify Lookup Keys (Lookup Function)	25
Get Lookup Values (LookupValue Function)	26

About Lookup Tables and Functions

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

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

Figure 4.1 CSV File Imported Into SAS Business Rules Manager


	A	B
1	AU	Australia
2	BR	Brazil
3	CA	Canada
4	CR	Costa Rica
5	DK	Denmark
6	EE	Estonia
7	ES	Spain
8	FR	France
9	GL	Greenland
10	HK	Hong Kong
11	IN	India
12	JP	Japan
13	KE	Kenya

Values	
Key	Value
AU	Australia
BR	Brazil
CA	Canada
CR	Costa Rica
DK	Denmark
EE	Estonia
ES	Spain
FR	France
GL	Greenland
HK	Hong Kong



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

SAS Business Rules Manager provides two functions, `Lookup` and `LookupValue`, that enable you to determine whether a lookup key exists in a lookup table and to retrieve a lookup value from a lookup table.

Create a New Lookup Table


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

To create a new lookup table:

1. In the Definitions workspace, select **Data** category, and then select **Lookups**.
2. Right-click on the folder where you want to create the new lookup table, and select **New Lookup**. Alternatively, select the folder where you want to add the new lookup table, click  on the toolbar, and then select **New Lookup**.
3. In the New Lookup dialog box, enter a name for the new lookup table. Names are limited to 32 characters and can contain only alphanumeric characters and underscores. Lookup table names must be unique within the business rules database.
4. (Optional) Enter a description for the new lookup table. Descriptions are limited to 256 characters.
5. Click , and select the CSV file that contains the lookup data.
6. Click **OK**.

Refresh a Lookup Table


To refresh a lookup table:

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

Delete a Lookup Table

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

To delete a lookup table:

1. In the Definitions workspace, select **Data** category, and then select **Lookups**.
2. Select the lookup table and click . Alternatively, you can right-click on the lookup table, and then select **Delete** from the menu.

Verify Lookup Keys (Lookup Function)

You can use the Lookup function to verify that a key value exists in a lookup table. You can specify the Lookup function in condition expressions only. Specify the Lookup function as the expression for the term whose value is the lookup key that you want to search for. The syntax of the Lookup function is as follows:

Lookup ("lookup_table_name")

For *lookup_table_name*, specify the name of the lookup table that you want to search.

For example, if you want to verify that the value of the term Ctry_Key exists as a key value in the table Country_Codes, enter the Lookup function as the expression for the Ctry_Key term as shown in the following display.

Condition Term	
#	 Ctry_Key
1	Lookup("Country_Codes")

The Lookup function returns a value of **True** or **False**, depending on whether the key value exists in the lookup table. For example, suppose the Ctry_Key column in the current input record contains the value "CA". If the Country_Codes lookup table contains the lookup key "CA", then the expression shown in the display above evaluates to True.

Note: If an expression contains the Lookup function, then the expression cannot contain anything else.

Get Lookup Values (LookupValue Function)

You can use the LookupValue function to retrieve a lookup value from a lookup table. You can specify the LookupValue function in action expressions only. The syntax of the LookupValue function is as follows:

LookupValue ("lookup_table_name", term_or_string)

lookup_table_name

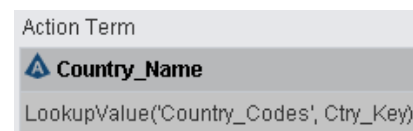
the name of the lookup table that you want to search.

term_or_string

a term or character string that specifies the lookup key for the value that you want to retrieve. Enclose character strings in quotation marks.

For example, suppose the term Ctry_Key contains a lookup key. To retrieve the lookup value that is associated with that key from the table Country_Codes, enter the following expression for the Ctry_Key term:

```
LookupValue('Country_Codes', Ctry_Key)
```



The first few entries in the Country_Codes lookup table are shown below.

Values	
Key	Value
AU	Australia
BR	Brazil
CA	Canada
CR	Costa Rica

For example, suppose the Ctry_Key column in the current input record contains the value "CA". The Country_Codes lookup table contains the lookup key "CA", and the lookup value that corresponds to that key is "Canada". The expression shown in the display above assigns the value "Canada" to the term Country_Name.

Note: If an expression contains the LookupValue function, then the expression cannot contain anything else.

Chapter 5

Managing Rules and Rule Sets

About Rules and Expressions	28
About Rule Sets	28
Create a New Rule Set	29
Open an Existing Rule Set	29
Defining New Rules in the Rule Set	29
Add a New Rule	29
Define Expressions for a Rule	30
Controlling Which Conditions Are Evaluated	32
Using the Expression Editor	33
Punctuation for Data Values	36
Operators for Use in Expressions	37
Using Functions in Expressions	37
Using SAS Business Rules Manager Macros in Expressions	38
Working with Missing Values	38
Terms and Operators Added by SAS Business Rules Manager	38
Examples of Expressions	39
Validate the Expressions in a Rule Set	41
Change the Order of Rules in a Rule Set	41
Move a Rule to a New Position in a Rule Set	41
Swap Two Rules	42
Copy Rules and Expressions	42
Copy an Entire Rule	42
Copy Text within a Rule	42
Delete Rules and Expressions	43
Delete a Rule	43
Delete or Cut and Paste Text within a Rule	43
Edit the Name and Description of a Rule Set	43
Edit the Name and Description of a Rule	44
Copy a Rule Set	44
Move a Rule Set to a Different Folder	44
Delete a Rule Set	44
Save a Rule Set	45

About Rules and Expressions

A rule specifies conditions to be evaluated and actions to be taken if those conditions are satisfied. Most rules correspond to this form:

if condition_expressions then action_expressions

For example, suppose you have the following rule:

if customer_debt > customer_assets then app_status="Decline"

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

The following figure shows the decision table with this rule added to it:

Condition Term		Action Term
#	123 customer_debt	A app_status
If	1 >customer_assets	"Decline"

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

Condition Term		Action Term
#	A HomeOwner 123 Credit_Score	I Approved 123 Loan_Rate
1	"true" >700	"true" =4.5

SAS Business Rules Manager generates the following rule condition:

(HomeOwner='true') AND (Credit_Score > 700)

Action expressions are always assignment statements.

About Rule Sets


Rules are grouped together into rule sets. Rule sets are logical collections of rules that are grouped together because of interactions or dependencies between the rules or because they are processed together when they are published.

By default, the condition expressions for all rules are evaluated regardless of the results of previous rules, and they are executed sequentially. However, you can use the ELSE and OR operators to control when the condition expression for a rule is evaluated. See [“Controlling Which Conditions Are Evaluated” on page 32](#) for more information.

For all rules, if the rule’s condition expression evaluates to true, SAS Business Rules Manager executes the rule’s action expression.

Create a New Rule Set

To create a new rule set:

1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the folder where you want to create the new rule set, and select **New Rule Set**. Alternatively, select the folder where you want to add the new rule set, click  in the toolbar, and then select **New Rule Set**.
3. In the New Rule Set dialog box, enter a name for the new rule set. Rule set names are limited to 100 characters and must be unique within a folder.
4. (Optional) Enter a description for the new rule set. Descriptions are limited to 256 characters.
5. Select the vocabulary that is associated with the new rule set.
6. Click **Create**. SAS Business Rules Manager opens the decision table for the new rule set.

The default view of a rule set is the horizontal view. In the horizontal view, the terms used by the rules in the rule set are displayed across the top of the decision table, and the decision table has one row for each rule in the rule set. To switch to the vertical view, select **Vertical** from the menu in the toolbar above the decision table. In the vertical view, the terms used by the rules are displayed in the left column, and the decision table has one column for each rule in the rule set.

Open an Existing Rule Set



In the Designer workspace, select the **Business Rules** category. You can open a rule set in one of the following ways:


- Double-click on the rule set.
- Select the rule set, and click **Open** in the toolbar.
- Right-click on the rule set, and select **Open**.

Defining New Rules in the Rule Set

Add a New Rule

There are three ways to add a new rule to a rule set:

- Click . You can modify the rule name and description on the **Rule Details** tab.
- Click in the decision table in the same row as . You can modify the rule name and description on the **Rule Details** tab.

- Click  in the toolbar that is above the decision table. SAS Business Rules Manager opens the New Rule dialog box.
 1. (Optional) Enter a name for the new rule.
 2. (Optional) Enter a description for the new rule.
 3. (Optional) Change the order of the new rule. The rule order, in addition to the IF, ELSE, or OR keyword (see [“Controlling Which Conditions Are Evaluated”](#)), controls how rules are evaluated within the rule set.

Note: You can also change the order of the rules later by right-clicking on a rule order number and selecting either **Reorder** or **Swap**. See [“Change the Order of Rules in a Rule Set”](#) on page 41 for more information.
 4. Click **OK**. SAS Business Rules Manager adds a row (or column, in the vertical view) for the new rule to the decision table.

Rule names are limited to 100 characters and must be unique within a rule set. Descriptions are limited to 256 characters.

Define Expressions for a Rule

To define the expression for a rule:


1. Add any additional terms to the decision table that are needed by the rule. You can add a single term in one of two ways:
 - Right-click on the term in the Vocabulary pane, and select either **Use as condition term** or **Use as action term**.
 - Drag the term from the Vocabulary pane onto a column in the decision table.

You can also add multiple terms at the same time.

 - If the terms that you want to add to the table all belong to the same entity, you can add the entire entity to the decision table in the same way that you add a single term. Note that an entity can contain dozens of terms, and there is no way to undo this operation after the terms have been added to the decision table.
 - If the terms do not all belong to the same entity, select the terms and drag them onto the table, or use the right-click menu as you would for a single term. To select a consecutive set of terms, click on the first term, hold down the Shift key, and click on the last term. To select nonconsecutive terms, hold down the Ctrl key, and click on each term that you want to select.

For example, if your rule is **If balance <100 then risk = "high"**, the condition term is **balance** and the action term is **risk**.


TIP By default, terms are listed in the decision table in alphabetical order. You can reorder the terms by dragging the terms in the column or row headings.

TIP You can add new entities and terms by clicking  in the Vocabularies pane. You can also edit existing entities and terms by right-clicking on the entity or term and selecting the appropriate option. SAS Business Rules Manager opens the same dialog box that you would see if you were working in the Definitions workspace. See [“Create New Entities”](#) on page 18 and [“Create New Terms”](#) on page 19 for more information.

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

Expressions can be up to 1024 characters long. They can contain numeric constants, character strings, vocabulary terms, operators, and SAS functions. Condition expressions can also contain SAS Business Rules Manager macros and the Lookup function. Action expressions can contain the LookupValue function. However, if the expression contains the Lookup or LookupValue function, then the expression cannot contain anything else. If the expression contains a SAS Business Rules Manager macro, the macro must appear as the first part of the expression.

TIP A rule that does not have a condition expression always executes.

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


As you enter expressions into each cell, SAS Business Rules Manager displays the rule conditions and actions, including the operators and term names that are added by SAS Business Rules Manager, on the **Rule Details** tab. (See [“Terms and Operators Added by SAS Business Rules Manager” on page 38](#).) For example, suppose you enter the following rule in the decision table:

Condition Term		Action Term
#	 Credit_score  DownPmt	 Risk_Category
1	>=720 AND <750 >=.20	= "Medium"

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

```
Condition: (Credit_score >= 720 and Credit_score < 750) AND (DownPmt >= .20)
Action: Risk_Category = "Medium"
```

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

- [“Using the Expression Editor” on page 33](#)
 - [“Punctuation for Data Values” on page 36](#)
 - [“Operators for Use in Expressions” on page 37](#)
 - [“Verify Lookup Keys \(Lookup Function\)” on page 25](#)
 - [“Get Lookup Values \(LookupValue Function\)” on page 26](#)
 - [“Using Functions in Expressions” on page 37](#)
 - [“Using SAS Business Rules Manager Macros in Expressions” on page 38](#)
 - [“Working with Missing Values” on page 38](#)
 - [“Terms and Operators Added by SAS Business Rules Manager” on page 38](#)
 - [“Examples of Expressions” on page 39](#)
3. (Optional) Select the operator for the rule. The default operator is IF. See [“Controlling Which Conditions Are Evaluated” on page 32](#) for more information.
 4. Click  to save the rule set. SAS Business Rules Manager validates the syntax of the expressions. If it does not detect any problems, it saves the rule set. See [“Validate the Expressions in a Rule Set” on page 41](#) for more information.

Controlling Which Conditions Are Evaluated

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

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

		Condition Term	Action Term
		# 123 Order_Quantity	123 Offer_Percent
If	1	<=5	=5
Else	2	>=10 AND <=15	=10
Else	3	>=20 AND <=25	=15
Else	4	>=30	=20

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

For example, suppose you have a very long condition expression such as the one shown in the following display.

		Condition Term
		# 123 Order_Quantity
If	1	<=5 OR (Order_Quantity >= 10 AND Order_Quantity <=15) OR (Order_Quantity >= 20 AND Order_Quantity <=25) OR Order_Quantity >=30




As shown in the next display, you can break this expression into four different rules and use the OR operator, which makes the rule much easier to read and edit.

		Condition Term
		# 123 Order_Quantity
If	1	<=5
Or	2	>=10 AND <=15
Or	3	>=20 AND <=25
Or	4	>=30

If you assign the OR operator to a rule, then you cannot enter an action expression for the rule. SAS Business Rules Manager uses the action expression of the last rule that was assigned the IF or ELSE operator.

For example, for the following rule set, rules 1 through 4 use the action expression that is defined for rule 1. Rules 5 through 7 use the action expression that is defined for rule 5.

Display 5.1 Rule Set That Defines Seven Rules in Two IF Blocks


		Condition Term	Action Term	
	#	 Order_Quantity	 Direct_mail_offer	 Offer_Percent
If	1	<=5	=False	=5
Or	2	>=10 AND <=15		
Or	3	>=20 AND <=25		
Or	4	>=30		
If	5	>=6 AND <=9	=True	=10
Or	6	>=16 AND <=19		
Or	7	>=26 AND <=29		

An IF block is a series of rules that begins with an IF operator and extends up to but does not include the next IF operator. In the rule set shown in [Display 5.1 on page 33](#), rules 1 to 4 are an IF block and rules 5 to 7 are a second IF block.

Note: A rule that does not have a condition expression must be the last rule in an IF block.

Using the Expression Editor

Open the Expression Editor

To open the Expression Editor, click in a cell in the decision table, and select . Alternatively, right-click in the cell, and select **Expression Editor**.

Display 5.2 Expression Tab for a Condition Term

Rule 6

Expression **Lookup Value**

=High

+ - * / = ^= < > AND OR ! IN NOT IN

Vocabulary

- Vocab_Retail_Loans
 - Customer
 - Loan_Application
 - Down_payment_amt
 - Interest_rate
 - Principal

Domain values

- High
- Medium
- Low

OK Cancel

If you open the Expression Editor for a condition term, then the Expression Editor contains the **Expression** tab and the **Lookup** tab. If you open the Expression Editor for an action term, it contains the **Expression** tab and the **LookupValue** tab. Also, the comparison operators ^=, <, >, AND, OR, IN, and NOT IN are disabled because action expressions can be assignment expressions only.

Build an Expression in the Expression Editor

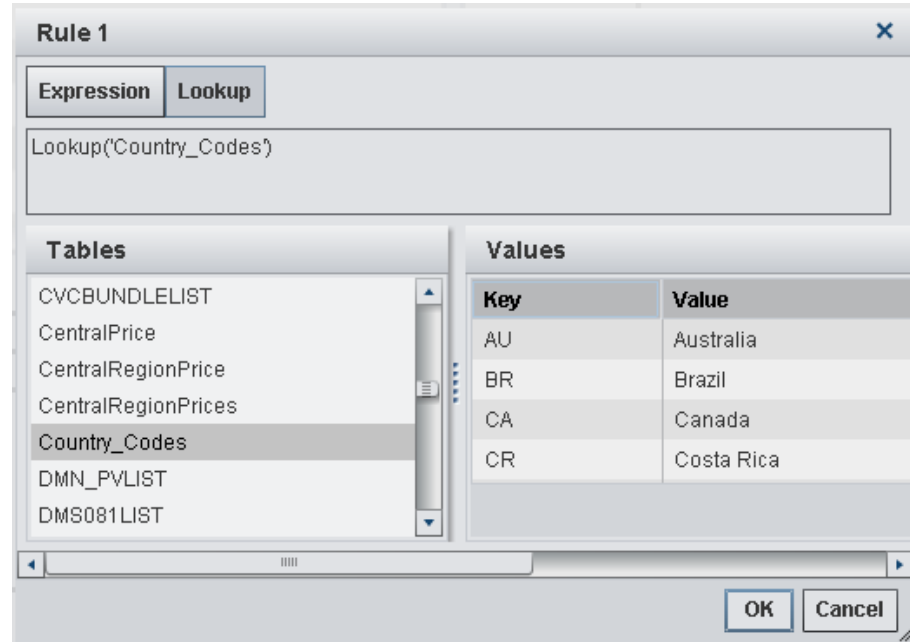
To define expressions that do not use the Lookup or LookupValue functions, enter the expression on the **Expression** tab. Click on the operators, vocabulary terms, and domain values as needed to add them to the expression. The Expression Editor builds the expression in the top field. To add numeric constants, character strings, or functions to the expression, enter them directly into the top field. (Remember to use the correct punctuation. See [“Punctuation for Data Values” on page 36](#).) When you are finished, click **OK**. The Expression Editor adds the expression to the cell in the decision table where you opened the editor.

To build an expression that uses the Lookup or LookupValue functions, switch to the **Lookup** or **LookupValue** tabs. You can enter the Lookup function in condition expressions only, and you can enter the LookupValue function in action expressions only. See [“Specify the Lookup Function”](#) and [“Specify LookupValue Function”](#) for more information.

Specify the Lookup Function

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

Display 5.3 Lookup Tab in the Expression Editor



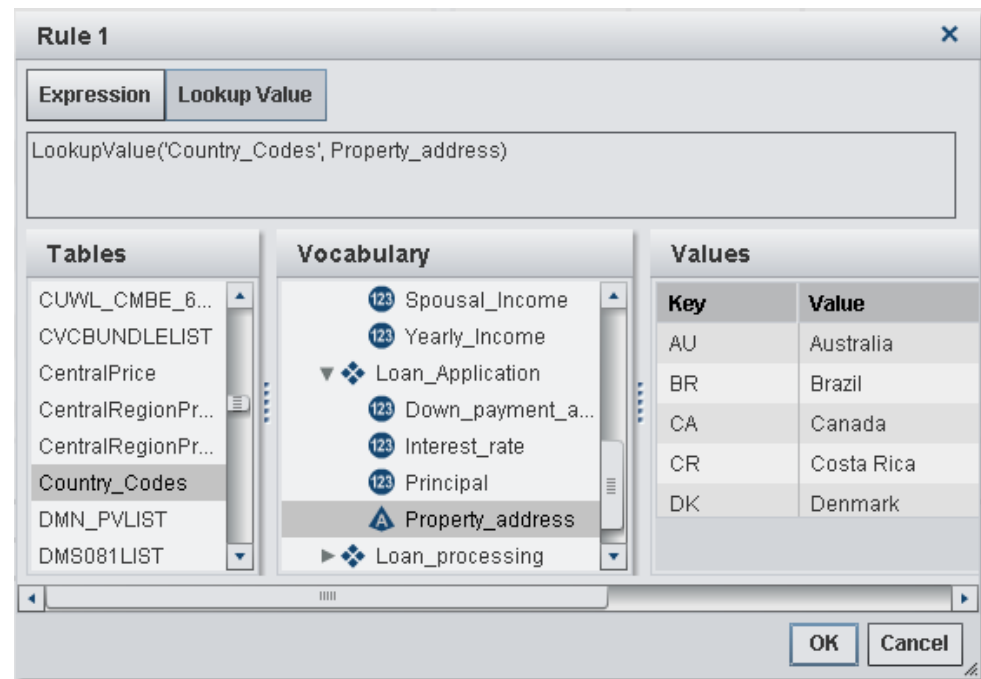
For more information, see [“Verify Lookup Keys \(Lookup Function\)”](#) on page 25.

Specify LookupValue Function

To use the Expression Editor to enter the LookupValue function:

1. Click the **LookupValue** tab.
2. Double-click on the lookup table name.
3. Specify the term name or the character string that contains the lookup key value. To specify a term, double-click on the term name. To specify a character string as the lookup key value, enter the character string in the field at the top of the **LookupValue** tab. Enclose the string in quotation marks.
4. Click **OK**.

For more information, see [“Get Lookup Values \(LookupValue Function\)”](#) on page 26.

Display 5.4 LookupValue Tab in the Expression Editor

Punctuation for Data Values

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

Table 5.1 Punctuation Needed for Data Values

Data Type	Punctuation Needed	Example
String	Enclose strings in either single or double quotation marks.	<code>= 'Gold Account'</code> <code>= "Ineligible"</code> <code>= "d'oscail"</code>
Date	Enter date values by using the format DDMMMYYYY. Enclose each value in quotation marks followed by d .	<code>= '01SEP2012'd</code> <code>>= '31SEP2012'd</code>
Datetime	Enter datetime values by using the format DDMMMYYYY:HH:MM:SS. Use 24-hour clock notation. Enclose each value in quotation marks followed by dt .	<code>= '01SEP2012:15:00:00'dt</code> <code><= '31SEP2012:15:00:00'dt</code>
Boolean	Boolean values do not need to be enclosed in quotation marks. Enter only True or False .	<code>= True</code> <code>= False</code>

Operators for Use in Expressions

The following table lists the operators that you can use in an expression. Do not enter a space between the elements of the operators `<=`, `>=`, or `^=`. See “SAS Operators in Expressions” in Chapter 6 of *SAS Language Reference: Concepts* for more information about the operators shown in the table.

Table 5.2 Operators

Operator	Definition	Example
<code>*</code>	Multiply	<code>.085 * sales</code>
<code>/</code>	Divide	<code>amount / 5</code>
<code>+</code>	Add	<code>num + 3</code>
<code>-</code>	Subtract	<code>sale - discount</code>
<code>=</code>	Equal to	<code>numTries = maxTriesAllowed</code>
<code>^=</code>	Not equal to	<code>insufficientFunds ^= True</code>
<code>></code>	Greater than	<code>daysLate > 5</code>
<code><</code>	Less than	<code>num < 8</code>
<code>>=</code>	Greater than or equal to	<code>balance >= 1000</code>
<code><=</code>	Less than or equal to	<code>balance <= 250</code>
<code>IN (value-list)</code>	Equal to an item in <i>value-list</i>	<code>in ("high","medium","low")</code>
<code>NOT IN (value-list)</code>	Not equal to an item in <i>value-list</i>	<code>not in (10,20,30)</code>
<i>expression</i> AND <i>expression</i>	If both expressions are true, the result is true.	<code>dateExpired >= '01AUG2012'd AND dateExpired <= '31AUG2012'd</code>
<i>expression</i> OR <i>expression</i>	If either expression is true, the result is true.	<code>dateEnrolled >= '01JAN2012' OR member = True</code>

Using Functions in Expressions

SAS Business Rules Manager supports the following functions in rule expressions:

- Lookup and LookupValue functions. See “[Verify Lookup Keys \(Lookup Function\)](#)” on page 25 and “[Get Lookup Values \(LookupValue Function\)](#)” on page 26 for more information.

- DS2 functions that are listed in [Table A1.1 on page 71](#). For information about these DS2 functions, see *SAS DS2 Language Reference* at <http://support.sas.com/documentation/onlinedoc/base/index.html>.
- Base SAS functions INPUTC and PRXMATCH. For information about these functions, see *SAS Functions and CALL Routines: Reference* at <http://support.sas.com/documentation/onlinedoc/base/index.html>.

Using SAS Business Rules Manager Macros in Expressions

SAS Business Rules Manager provides several macros that you can use to search for character strings or patterns in term values. You can specify these macros in condition expressions only. All of these macros return True or False, depending on whether the character string or pattern is found. See [Chapter 7, “SAS Business Rules Manager Macro Reference,” on page 57](#) for information about these macros.

Working with Missing Values

You can enter a missing value for a character string as a null string (`= ""`), and you can use a period (`= .`) to designate missing numeric values.

Missing values have a value of **false** when you use them with logical operators such as AND or OR. For more information, see Chapter 5, “Missing Values,” in *SAS Language Reference: Concepts*.

You can also use the MISSING function to check for missing values. This function returns a 0 (false) or 1 (true). For more information, see *SAS Functions and CALL Routines: Reference*.

Terms and Operators Added by SAS Business Rules Manager

Remember these rules when you are entering expressions:

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

Examples of Expressions

The following table shows examples of expressions that you can specify. Unless otherwise specified, all of these expressions are valid as condition expression.

Table 5.3 Examples of Expressions


Expression As Entered Into the Decision Table For Term X	Resulting Expression	Valid as a Condition Expression	Valid as an Action Expression
5	<code>x=5</code>	Yes	Yes
<code>=5</code>	<code>x=5</code>	Yes	Yes
<code>"mystring"</code>	<code>x="mystring"</code>	Yes	Yes
<code>=term1</code>	<code>x=term1</code>	Yes	Yes
<code>5 or >100</code>	<code>x=5 or x>100</code>	Yes	No See Note 1 .
<code>^=5 and x<10</code>	<code>x^=5 and x<10</code>	Yes	No See Note 1 .
<code>^=5 or >=(100/4)</code>	<code>x^=5 or x>=(100/4)</code>	Yes	No See Note 1 .
<code>in (10,20,30)</code>	<code>x IN (10,20,30)</code>	Yes	No See Note 1 .
<code>not in ('med','high')</code>	<code>x NOT IN ('MED','HIGH')</code>	Yes	No See Note 1 .
<code><'10JUN2012'd</code>	<code>x<'10JUN2012'd</code>	Yes	No See Note 1 .
<code>>'10JUN2012:17:00:00'dt</code>	<code>x>'10JUN2012:17:00:00'dt</code>	Yes	No See Note 1 .
<code>=ABS (-10)</code>	<code>x=ABS (-10)</code>	Yes	Yes
<code>=True</code>	<code>x=True</code>	Yes	Yes
<code>False</code>	<code>x=False</code>	Yes	Yes
<code>&myMacroVar</code>	<code>x=&myMacroVar</code>	Yes See Note 2 .	Yes See Note 2 .

Expression As Entered Into the Decision Table For Term X	Resulting Expression	Valid as a Condition Expression	Valid as an Action Expression
<code>%EVAL (&myMacroVar)</code>	<code>x=%EVAL (&myMacroVar)</code>	Yes See Note 2 .	Yes See Note 2 .
<code>term1=5</code>	<code>x=term1=5</code>	Yes See Note 3 .	Yes See Note 4 .
<code>term1=3 or term2=5</code>	<code>x=term1=3 or term2=5</code>	Yes See Note 5 .	Yes See Note 6 .
<code>5 or (x>10 and <20)</code>	This expression is invalid as both a condition expression and as an action expression. SAS Business Rules Manager does not add column names after nested AND or OR operators.		
<code>>"mystring"</code>	This expression is invalid as both a condition expression and as an action expression. SAS Business Rules Manager checks whether literal types are compatible with the specified operators. Character strings are not compatible with numeric operators.		
<code>+10</code>	This expression is invalid as both a condition expression and as an action expression. SAS Business Rules Manager adds column names to the expression only after AND and OR operators. It does not add column names after other operators such as the plus sign.		

Notes:

1. Action expressions must be assignment expressions only.
2. SAS Business Rules Manager validates macro functions and variables based only on whether the syntax is correct. It does not check to determine whether a macro function or variable will be accessible when the rule set is executed.
3. This expression is valid. However, it should be avoided. As a condition statement, this expression checks to determine whether both **x** and **term1** are equal to 5. The recommended way to enter this expression is `=5 and term1=5`.
4. As an action expression, this expression becomes a Boolean assignment statement. This expression determines whether **term1** is equal to 5, and if so, assigns a value of 1 (true) to **x**. If not, it assigns a value of 0 (false) to **x**.
5. This expression is valid. However, it should be avoided. As a condition statement, this expression checks to determine whether both **x** and **term1** are equal to 3 or whether **term2** is equal to 5. The recommended way to enter this expression is `(=5 or term1=3) or term2=5`.
6. As an action expression, this expression becomes a Boolean assignment statement. This expression determines whether either **term1** is equal to 3 or **term2** is equal to 5, and if so, assigns a value of 1 (true) to **x**. If not, it assigns a value of 0 (false) to **x**.

Validate the Expressions in a Rule Set

To validate the syntax in rule expressions, click .

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

Change the Order of Rules in a Rule Set

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

Move a Rule to a New Position in a Rule Set


There are two ways to move a rule to a new position:

- Drag the rule to the new position.
- Use the **Reorder** menu selection.

To drag a rule to a new position in the rule set, move the mouse pointer over the rule order number, and drag the rule number to the new position.

To use the **Reorder** menu selection to move a rule:

1. Right-click on the order number of the rule that you want to move, and select **Reorder**.



The dialog box titled "Reorder the Rule" contains the following fields and controls:

Rule name:	Default Rule Name 2
Current position:	2
New position:	4

At the bottom right, there are two buttons: **OK** and **Cancel**.

2. Select the new position number for the rule.
3. Click **OK**. SAS Business Rules Manager moves the rule to the new position and repositions the remaining rules up or down as needed.

Swap Two Rules

To swap the position of two rules:

1. Right-click on the order number of one of the rules that you want to move, and select **Swap**.



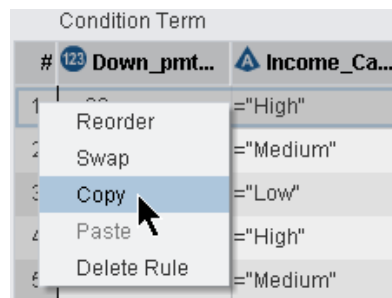
2. Select the position number for the second rule that you want to move.
3. Click **OK**. SAS Business Rules Manager swaps the positions of the two rules and leaves all other rules in their original positions.

Copy Rules and Expressions

Copy an Entire Rule

To copy and paste an entire rule:

1. Right-click on the order number of the rule that you want to copy, and select **Copy**.



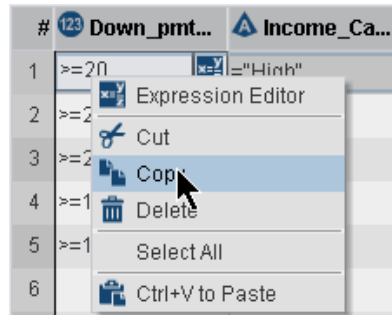
2. Right-click in the decision table, and select **Paste**. SAS Business Rules Manager adds the copied rule to the bottom of the decision table. You can then edit or reorder the new rule as needed.

Copy Text within a Rule

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

1. Click in the table cell that contains the text that you want to copy.
2. Select the text that you want to copy. To select all of the text in a cell, right-click and select **Select All**.

3. Right-click on the text and select **Copy**.




4. Click in the cell where you want to paste the text and press Ctrl+V.

Delete Rules and Expressions

Delete a Rule

You can delete a rule in one of two ways:

- Click on the order number of the rule that you want to delete, and click .
- Right-click on the order number of the rule, and select **Delete Rule**.

Delete or Cut and Paste Text within a Rule

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

1. Select the cell in the decision table containing the text that you want to copy.
2. Select the text that you want to cut. To select all of the text in a cell, right-click and select **Select All**.
3. Right-click in the cell and select **Delete** or **Cut**.
4. To paste the text, click in the cell where you want to paste the text, and press Ctrl+V.


Edit the Name and Description of a Rule Set

To change the name or description of an existing rule set:

1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the rule set that you want to edit, and select **Edit Rule Set**.
3. Change the name and description as needed, and click **OK** to save the changes.

Edit the Name and Description of a Rule

To edit the name or description of an individual rule, you must open the rule set that contains the rule. You can edit the name and description of any of the rules in the rule set by clicking on the rule, and editing the name and description on the **Rules Details** tab.

Click  to save your changes.

Copy a Rule Set

You can create a copy of a rule set within the same folder.

To copy a rule set:

1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the rule set, and select **Copy Rule Set** from the menu. SAS Business Rules Manager opens the Copy Rule Set dialog box.
3. Enter the name for the new copy of the rule set.
4. (Optional) Enter a description for the rule set.
5. Click **OK**.

Move a Rule Set to a Different Folder


To move a rule set to a different folder:

1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the rule set, and select **Move Rule Set** from the menu. SAS Business Rules Manager opens the Choose a Location dialog box.
3. Select the folder where you want to move the rule set to.
4. Click **Move**.


Delete a Rule Set

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

To delete a rule set:

1. In the Designer workspace, select the **Business Rules** category.
2. Select the rule set and click . Alternatively, you can right-click on the rule set, and select **Delete** from the menu.

Save a Rule Set

To save changes to a rule set, click . SAS Business Rules Manager validates the syntax of the expressions and displays an error message if it finds any problems. If SAS Business Rules Manager does not detect any problems with any of the expressions, it saves the rule set. See [“Validate the Expressions in a Rule Set” on page 41](#) for more information.

Chapter 6

Creating and Publishing Rule Flows

Introduction to Rule Flows	47
Simple Rule Flows, Complex Rule Flows, and BY Groups	48
Create a Rule Flow	48
Open an Existing Rule Flow	49
Open Rule Sets from the Rule Flow Editor	49
Add Attachments to a Rule Flow	49
Add Comments to a Rule Flow	50
Change the Order of the Rule Sets	50
View the Terms Used in a Rule Flow	50
Edit Name and Description for a Rule Flow	51
Copy a Rule Flow	51
Move a Rule Flow to a Different Folder	51
Delete a Rule Set from a Rule Flow	51
Delete a Rule Flow	52
Testing a Rule Flow	52
Input Data for Rule Flow Tests	52
Test Setup	52
Test a Rule Flow	53
View Rule Flow Results and Test Output	53
Publish a Rule Flow	54
Display Publish Information for Rule Flows	54

Introduction to Rule Flows

A business rule flow is a logical collection of multiple rule sets that define multiple conditions and actions. In general, the rule sets in a rule flow are executed in the order in which they are defined in the rule flow. However, with complex rule flows, certain sections of rule sets are usually executed more times than others. See [“Simple Rule Flows, Complex Rule Flows, and BY Groups”](#) on page 48 for more information.

After you publish a rule flow to the SAS Metadata Server, other applications can deploy the published rule flows as SAS programs and services. These programs and services process input data, which contains conditions, in order to create output data, which contains actions. The terms used in the rule flows are mapped to table columns by the applications that use the published rule flows within metadata.

Simple Rule Flows, Complex Rule Flows, and BY Groups

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


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

For complex rule flows, you can specify *BY-group terms*. If you specify BY-group terms, then SAS Business Rules Manager sorts the input data by those terms, and results are calculated for each group of input records that have the same value for all of the terms. Output is generated for each BY group instead of for each input record.

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

Create a Rule Flow

To create a rule flow:




1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the folder where you want to create the new rule flow, and select **New Rule Flow**. Alternatively, select the folder where you want to add the new rule flow, click  in the toolbar, and then select **New Rule Flow**.
3. In the New Rule Flow dialog box, enter a name for the new rule flow. Rule flow names are limited to 32 characters and can contain any character except forward slash (/), backslash (\), left brace ({), right brace (}), colon (:), and question mark (?).
4. (Optional) Enter a description for the new rule flow. Descriptions are limited to 256 characters.
5. Click **Create**. SAS Business Rules Manager opens the rule flow editor.
6. (Optional) Select **Complex Rule Flow** from the menu in the toolbar. SAS Business Rules Manager adds Initial and Final sections to the rule flow table. The rules in these sections are run at the start and end of the rule flow.
7. Drag the rule sets that you want to add to the rule flow from the Resources pane onto the rule flow table.

Note: A rule flow can use only rule sets that are defined for the same vocabulary.

After the first rule set is added to the rule flow, the vocabulary for the rule flow is

established. Only the rule sets that use the same vocabulary are displayed in the Resources pane.

Note: A rule set can be added to the same rule flow only once.


8. (Optional) If you selected **Complex Rule Flow**, specify BY-group processing. When you specify BY-group processing, all of the input records that have the same values for the BY-group terms are processed before output is generated. One output record is written for each group.
 - a. Select the terms that serve as BY-group terms. SAS Business Rules Manager adds Group Start and Group End sections to the table.
 - b. (Optional) Drag the rule sets that you want to add to the new sections into the Group Start and Group End sections of the table. The rules in these groups are run at the start and end of each BY group.
9. (Optional) Reorder the rule sets. To move a rule set, select the rule set, and click  or  to move it to a different row in the table. To move a rule set to a different section (Initial, Main, and so on), you must delete the rule set, and then add it to the other section.
10. Click  to save the rule flow.

Open an Existing Rule Flow

To open an existing rule flow:


1. In the Designer workspace, select the **Business Rules** category.
2. Do one of the following:
 - Double-click on the rule flow.
 - Select the rule flow, and click **Open** on the toolbar.
 - Right-click on the rule flow, and select **Open**.

Open Rule Sets from the Rule Flow Editor

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

Add Attachments to a Rule Flow

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

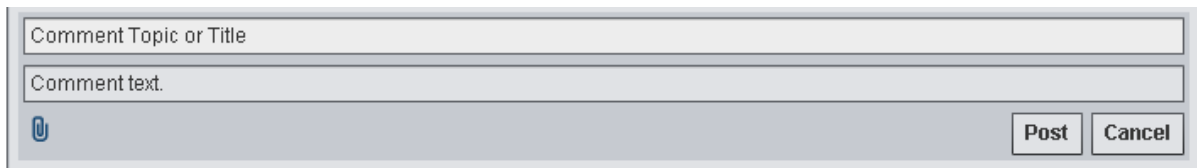
1. Use the Page Manager to select the Attachments page.
2. Click , and select the attachment file.


3. Click .

Add Comments to a Rule Flow


To add a comment:

1. Use the Page Manager to select the Comments page.
2. Enter a topic title and enter the comment. The topic title is required, and the field for comments does not appear until you enter the topic title.



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



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

Click  to see comments that have been posted by others.


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


Change the Order of the Rule Sets


You can reorder the rule sets in a rule flow in one of two ways:

- Select the rule set, and click  or  to move a rule set up or down within the rule flow.
- Drag the order number of the rule set to a new position.

View the Terms Used in a Rule Flow

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

To view only the input terms that are used in all of the rule sets in the rule flow, open the rule flow, and click .

To view only the output terms that are used in all of the rule sets in the rule flow, open the rule flow, and click .

Note: These icons are unavailable if you have made editing changes to the rule flow. You must save the changes to the rule flow before you click on these icons.

Edit Name and Description for a Rule Flow

To change the name or description of an existing rule flow:

1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the rule flow that you want to edit, and select **Edit Rule Flow**.
3. Change the name and description as needed, and click **OK** to save the changes.

Copy a Rule Flow

You can create a copy of a rule flow within the same folder.

To copy a rule flow:


1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the rule flow, and select **Copy Rule Flow** from the menu. SAS Business Rules Manager opens the Copy Rule Flow dialog box.
3. Enter a name for the new copy of the rule flow.
4. (Optional) Enter a description for the rule flow.
5. Click **OK**.

Move a Rule Flow to a Different Folder

To move a rule flow to a different folder:

1. In the Designer workspace, select the **Business Rules** category.
2. Right-click on the rule flow, and select **Move Rule Flow** from the menu. SAS Business Rules Manager opens the Choose a Location dialog box.
3. Select the folder where you want to move the rule flow to.
4. Click **Move**.


Delete a Rule Set from a Rule Flow

To remove a rule set from a rule flow, open the rule flow, select the rule set, and then click . Alternatively, you can right-click on the rule set, and select **Remove**.

Delete a Rule Flow

Note: You cannot delete a rule flow if it has been published.

To delete a rule flow:

1. In the Designer workspace, select the **Business Rules** category.
2. Select the rule flow and click . Alternatively, you can right-click on the rule flow and select **Delete**.

Testing a Rule Flow

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

Input Data for Rule Flow Tests

SAS Business Rules Manager expects the input data for the rule flow test to already exist and to be defined as a data table. See [“Managing Data Tables” on page 10](#) for information on defining data tables. Your user ID must have permission to access the data.

Test Setup

To specify code that you want to run before the rule flow is executed, enter the code on the **Test Setup** tab.

You can use the `&BRMPPrimaryEntityKey` and `&BRMPPrimaryTransactionDTTM` macro variables to specify values that are recorded in the `ENTITY_PRIMARY_KEY` and `TRANSACTION_DTTM` fields in the rule-fired data. Assigning values to these fields helps you determine the exact input record that caused a rule to execute. You can assign a term or a SAS expression to these macro variables.

The `&BRMPPrimaryEntityKey` macro variable must resolve to a character value. For example, you can assign an expression that contains the `CATS` function to this macro variable as follows:

```
%let brmPrimaryEntityKey=cats(make, '_', model);
```

To specify a term of the type string, specify the macro variable assignment as follows:

```
%let brmPrimaryEntityKey=termName;
```

If the term is numeric or if the expression results in a numeric value, specify the macro variable assignment by using the `PUT` function and an appropriate SAS format. Use the SAS format to convert the numeric value to a character string. For example:

```
%let brmPrimaryEntityKey=put(termName, 8.0);
```



The `&BRMPPrimaryTransactionDTTM` macro variable must resolve to a SAS datetime value. If you assign a term to this macro variable, the term must be of the type datetime. Expressions must resolve to a SAS datetime value. For example, you can use the

DATETIME function to assign the current date and time to this macro variable as follows:

```
%let brmPrimaryTransactionDTTM=datetime();
```

Test a Rule Flow

To test a rule flow:

1. Open the rule flow that you want to test.
2. (Optional) On the **Test Setup** tab, enter any SAS code, such as initialization or setup code, that you want to run before the rule flow is run. See [“Test Setup” on page 52](#) for more information.
3. Select the data source that contains the input data for the test, and click **Run**. When the rule flow has completed, the status changes to from  **Not Run** to  **Successful**.

SAS Business Rules Manager displays the results of the rule flow, the rule-fired data, the SAS code that was generated and run by SAS Business Rules Manager, the SAS log, and test reports. See [“View Rule Flow Results and Test Output” on page 53](#) for more information.

4. (Optional) Click **Save Results** to save the results of the test in SAS data sets. The output of the rule flow is saved in a SAS data set whose name begins with OUTPUT. The rule-fired data is saved in a SAS data set whose name begins with RULEFIRE. The test data, such as the number of records processed, is saved in a data set whose name begins with DEPEXEC. The rule flow identifier and your user ID are appended to the data set name. For example, if the rule flow identifier is 10052 and your user ID is BRMUSER1, then the test results are saved in SAS data sets named OUTPUT10052BRMUSER1, RULEFIRE10052BRMUSER1, and DEPEXEC10052BRMUSER1.

View Rule Flow Results and Test Output

After you have tested a rule flow, SAS Business Rules Manager executes the rule flow and displays the output that the rule flow produced on the **Results** tab.

SAS Business Rules Manager also displays the code that was generated and data about the test itself. The SAS code that was generated and run by SAS Business Rules Manager is displayed on the **Code** tab. The SAS log that was generated when that code was run is displayed on the **SAS Log** tab. The **Test Reports** tab displays a graph that shows the number of times that each rule set was executed. The **Rule Fired Data** tab displays the rule-fired data for the rule flow. A new rule-fired record is created every time a rule is executed. Each record in the table contains the following fields:

RULE_ACTION_FIRE_ID

the unique identification string that is generated for the rule each time it is executed.

RULE_SET_SK

the identification number of the rule set.

RULE_SET_NM

the name of the rule set.

RULE_SK



the identification number of the rule.

RULE_NM	the name of the rule.
RULE_FLOW_SK	the identification number of the rule flow.
RULE_FLOW_NM	the name of the rule flow.
RULE_FIRE_DTTM	the date and time that the rule was run.
DEPLMT_EXECUTION_ID	the identification string of the specific instance of the rule flow that was executed.
ENTITY_PRIMARY_KEY	the value of the term that was specified with the &BRMPPrimaryEntityKey macro variable in the test preprocessing code.
TRANSACTION_DTTM	the value of the term that was specified with the &BRMPPrimaryTransactionDTTM macro variable in the test preprocessing code. The TRANSACTION_DTTM is the value of the datetime term for the record that was being processed when the rule was executed.



See “[Test Setup](#)” on page 52 for information about the &BRMPPrimaryEntityKey and &BRMPPrimaryTransactionDTTM variables.

Publish a Rule Flow

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


1. Open the rule flow.
2. If the rule flow contains changes that have not been saved, click . You cannot publish a rule flow if it contains changes that have not been saved.
3. Click .
4. In the Choose a Location dialog box, select the location where you want to publish the rule flow.

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

Note: In the Choose a Location dialog box, to create a new subfolder, click . To delete an existing folder, select the folder, and click .


5. Click **OK**.

Display Publish Information for Rule Flows

To display publish information for a rule flow, open the rule flow, and click . SAS Business Rules Manager opens the rule flow information dialog box. This dialog box displays the name, rule flow identifier, vocabulary, and description for the rule flow. It

also displays the folder path to which the rule flow was published and the list of versions that have been published. For each version, it shows the date on which the version was published and the display name or user ID of the user that published the rule flow.

You can control whether the list is displayed in ascending or descending order according to date by clicking the **Date Published** column heading. If the column is sorted in ascending order, a ▲ appears beside the column heading. When the column is sorted in descending order, a ▼ appears.


Assign Customer Risk Level
✕

Name: Assign Customer Risk Level
Rule flow identifier: 10011
Vocabulary: Vocab_Retail_Loans
Description:

Assign income level and risk category for customer loan

Publish Path: /User Folders/brm2/My Folder
Publish Information:

Version	Date Published ▼	Published By
1.1	Jul 15, 2013 04:23 PM	brm2
1.0	Jul 15, 2013 04:23 PM	brm2

OK

Chapter 7

SAS Business Rules Manager Macro Reference

About SAS Business Rules Manager Macros	57
Dictionary	57
%BRM_CONTAINS	57
%BRM_ENDSWITH	58
%BRM_MATCHANY	59
%BRM_PRXMATCH	59
%BRM_STARTSWITH	68

About SAS Business Rules Manager Macros

SAS Business Rules Manager provides several macros that you can specify in the condition expression for a term in order to search the value of that term. Most of the macros search for character strings only. However, you can use the %BRM_MATCHANY macro to search for numeric values and the %BRM_PRXMATCH macro to search for patterns by using Perl regular expressions.

You can specify these macros in condition expressions only, and you must specify them as the first part of the expression. You cannot specify multiple SAS Business Rules Manager macros in the same expression. These macros do not return character strings or the specific position of a string, as some SAS functions do. These macros evaluate to either True or False.

Dictionary

%BRM_CONTAINS

Determines whether a term's value contains any of the specified character strings.

Restriction: You can specify this macro in condition expressions only, and you must specify it as the first part of the expression. You cannot specify multiple SAS Business Rules Manager macros in the same expression.

Syntax

`%BRM_CONTAINS` (*termToSearch*, *stringOrTerm* <, *stringOrTerm*...>)

Required Arguments

termToSearch

the term whose value you want to search. This term must match the row or column term where you specify the macro.

stringOrTerm

a character string or a term whose value contains a character string that you want to search for. You can specify up to 50 terms or character strings to search for. Enclose character strings in quotation marks. Separate multiple strings or terms with commas.

Example

If the value of `ordCode` is **BT39X2JH**, then all of the following expressions evaluate to True:

```
%brm_contains(ordCode,"39X2JH")
%brm_contains(ordCode,"BT3")
%brm_contains(ordCode,"9")
```

If `productCode` is **BUC7719T**, then the following expression evaluates to False:

```
%brm_contains(ordCode,productCode,"413")
```

%BRM_ENDSWITH

Determines whether a term's value ends with any of the specified character strings.

Restriction: You can specify this macro in condition expressions only, and you must specify it as the first part of the expression. You cannot specify multiple SAS Business Rules Manager macros in the same expression.

Syntax

`%BRM_ENDSWITH` (*termToSearch*, *stringOrTerm* <, *stringOrTerm*...>)

Required Arguments

termToSearch

the term whose value you want to search. This term must match the row or column term where you specify the macro.

stringOrTerm

a character string or a term whose value contains a character string that you want to search for. You can specify up to 50 terms or character strings to search for. Enclose character strings in quotation marks. Separate multiple strings or terms with commas.

Example

If the value of `ordCode` is **BT39X2JH**, then both of the following expressions evaluate to True:

```
%brm_endsWith(ordCode, "39X2JH")
%brm_endsWith(ordCode, "2JH", "NBDK", "123")
```

If productCode is **BT7919T**, then the following expression evaluates to False:

```
%brm_endsWith(ordCode, productCode, "bt3")
```

%BRM_MATCHANY

Determines whether a term's value matches any of the entries in the specified list.

Restriction: You can specify this macro in condition expressions only, and you must specify it as the first part of the expression. You cannot specify multiple SAS Business Rules Manager macros in the same expression.

Syntax

```
%BRM_MATCHANY (termToSearch, entry <, entry...>)
```

Required Arguments

termToSearch

the term whose value you want to search. This term must match the row or column term where you specify the macro.

entry

specifies a character string, a numeric constant, or a term whose value you want to search for. All entries must be of the same type, and they must match the type of the *termToSearch*.

Examples

Example 1

If the value of ordCode is **AUR30X93**, then the following expression evaluates to True:

```
%brm_matchAny(ordCode, "AUR30X93", "CRE54B39")
```

The following expression evaluates to False:

```
%brm_matchAny(ordCode, "AUR30X")
```

Example 2

If the value of both ordCode and prodCode are **5144419**, then the following expression evaluates to True:

```
%brm_matchAny(ordCode, 4982648, 2430762, prodCode)
```

%BRM_PRXMATCH

Determines whether a term's value matches the pattern that is specified in the Perl regular expression.

Restriction: You can specify this macro in condition expressions only, and you must specify it as the first part of the expression. You cannot specify multiple SAS Business Rules Manager macros in the same expression.

Syntax

`%BRM_PRXMATCH` (*termToSearch*, *Perl_regular_expression*)

Required Arguments

termToSearch

the term whose value you want to search. This term must match the row or column term where you specify the macro.

Perl_regular_expression

specifies either a Perl regular expression enclosed in forward slashes (/) or a term whose value is a Perl regular expression enclosed in forward slashes.

Details

You can use metacharacters to construct Perl regular expressions. The following topics describe the common metacharacters that you can use to construct these expressions.

General Constructs

Table 7.1 General Constructs

Metacharacter	Description
()	Indicates grouping.
<i>non-metacharacter</i>	Matches a character.
{ } [] () ^ \$. * + ? \	To match these characters, override (escape) with \.
\	Overrides the next metacharacter.
\n	Matches capture buffer <i>n</i> .
(?:...)	Specifies a non-capturing group.

Basic Perl Metacharacters

The following table lists the metacharacters that you can use to match patterns in Perl regular expressions.

Table 7.2 Basic Perl Metacharacters and Their Descriptions

Metacharacter	Description
\a	Matches an alarm (bell) character.
\A	Matches a character only at the beginning of a string.

Metacharacter	Description
\b	Matches a word boundary (the position between a word and a space): <ul style="list-style-type: none"> • r\b matches the “er” in “never” • er\b does not match the “er” in “verb”
\B	Matches a non-word boundary: <ul style="list-style-type: none"> • er\B matches the “er” in “verb” • er\B does not match the “er” in “never”
\cA-\cZ	Matches a control character. For example, \cX matches the control character control-X.
\C	Matches a single byte.
\d	Matches a digit character that is equivalent to [0–9].
\D	Matches a non-digit character that is equivalent to [^0–9].
\e	Matches an escape character.
\E	Specifies the end of case modification.
\f	Matches a form feed character.
\l	Specifies that the next character is lowercase.
\L	Specifies that the next string of characters, up to the \E metacharacter, is lowercase.
\n	Matches a newline character.
\num \$num	Matches capture buffer <i>num</i> , where <i>num</i> is a positive integer. Perl variable syntax (\$num) is valid when referring to capture buffers, but not in other cases.
\Q	Escapes (places a backslash before) all non-word characters.
\r	Matches a return character.
\s	Matches any whitespace character, including space, tab, form feed, and so on, and is equivalent to [\f\n\r\t\v].
\S	Matches any character that is not a whitespace character and is equivalent to [^\f\n\r\t\v].
\t	Matches a tab character.
\u	Specifies that the next character is uppercase.

Metacharacter	Description
\U	Specifies that the next string of characters, up to the \E metacharacter, is uppercase.
\w	Matches any word character or alphanumeric character, including the underscore.
\W	Matches any non-word character or nonalphanumeric character, and excludes the underscore.
\ddd	Matches the octal character <i>ddd</i> .
\xdd	Matches the hexadecimal character <i>dd</i> .
\z	Matches a character only at the end of a string.
\Z	Matches a character only at the end of a string or before newline at the end of a string.

Metacharacters and Replacement Strings

You can use the following metacharacters in both a regular expression and in replacement text when you use a substitution regular expression:

- \l
- \u
- \L
- \E
- \U
- \Q

These metacharacters are useful in replacement text for controlling the case of capture buffers that are used within replacement text. The following example uses the \u and \L metacharacters to replace the second character in MCLAUREN with a lower case letter:

```
data _null_;
  x = 'MCLAUREN';
  x = prxchange("s/(MC)/\u\L$1/i", -1, x);
  put x;
run;
```

SAS writes the following output to the log:

```
x=McLAUREN
```

For a description of these metacharacters, see [Table 7.2 on page 60](#).

Other Quantifiers

The following table lists other qualifiers that you can use in Perl regular expressions. The descriptions of the metacharacters in the table include examples of how the metacharacters can be used.

Table 7.3 Other Quantifiers

Metacharacter	Description
\	Marks the next character as either a special character, a literal, a back reference, or an octal escape: <ul style="list-style-type: none"> • \n matches a newline character • \\ matches \ • \(matches (
	Specifies the OR condition when you compare alphanumeric strings. For example, the construct x y matches either x or y: <ul style="list-style-type: none"> • z food matches either “z” or “food” • (z f)ood matches “zood” or “food”
^	Matches the position at the beginning of the input string.
\$	Matches the position at the end of the input string.
period (.)	Matches any single character except newline. To match any character including newline, use a pattern such as [.\n].
(pattern)	Specifies grouping. Matches a pattern and creates a capture buffer for the match. To retrieve the position and length of the match that is captured, use CALL PRXPOSN. To retrieve the value of the capture buffer, use the PRXPOSN function. To match parentheses characters, use \(or \).

Greedy and Lazy Repetition Factors

Perl regular expressions support “greedy” repetition factors and “lazy” repetition factors. A repetition factor is considered greedy when the repetition factor matches a string as many times as it can using a specific starting location. A repetition factor is considered lazy when it matches a string the minimum number of times that is needed to satisfy the match. To designate a repetition factor as lazy, add a ? to the end of the repetition factor. By default, repetition factors are considered greedy.

The following table lists the greedy repetition factors. The descriptions of the repetition factors in the table include examples of how they can be used.

Table 7.4 Greedy Repetition Factors

Metacharacter	Description
*	Matches the preceding subexpression zero or more times: <ul style="list-style-type: none"> • zo* matches "z" and "zoo" • * is equivalent to {0,}

Metacharacter	Description
+	Matches the preceding subexpression one or more times: <ul style="list-style-type: none"> zo+ matches "zo" and "zoo" zo+ does not match "z" + is equivalent to {1,}
?	Matches the preceding subexpression zero or one time: <ul style="list-style-type: none"> do(es)? matches the "do" in "do" or "does" ? is equivalent to {0,1}
{n}	Matches at least <i>n</i> times.
{n,}	Matches a pattern at least <i>n</i> times.
{n,m}	<i>m</i> and <i>n</i> are nonnegative integers, where $n \leq m$. They match at least <i>n</i> and at most <i>m</i> times: <ul style="list-style-type: none"> o{1,3} matches the first three o's in "foooooood" o{0,1} is equivalent to "o?" You cannot put a space between the comma and the numbers.

The following table lists the lazy repetition metacharacters.

Table 7.5 Lazy Repetition Factors

Metacharacter	Description
*?	Matches a pattern zero or more times.
+?	Matches a pattern one or more times.
??	Matches a pattern zero or one time.
{n}?	Matches exactly <i>n</i> times.
{n,}?	Matches a pattern at least <i>n</i> times.
{n,m}?	Matches a pattern at least <i>n</i> times but not more than <i>m</i> times.

Class Groupings

The following table lists character class groupings. You specify these classes by enclosing characters inside brackets. These metacharacters share a set of common properties. For a pattern match to be successful, the character class must always match a character. The negated character class must always match a character that is not in the list of characters that are designated inside the brackets. The descriptions of the metacharacters in the table include examples of how the metacharacters can be used.

Table 7.6 Character Class Groupings

Metacharacter	Description
[...]	Specifies a character set that matches any one of the enclosed characters: <ul style="list-style-type: none"> [abc] matches the “a” in “plain”
[^...]	Specifies a negative character set that matches any character that is not enclosed: <ul style="list-style-type: none"> [^abc] matches the “p” in “plain”
[a-z]	Specifies a range of characters that matches any character in the range: <ul style="list-style-type: none"> [a-z] matches any lowercase alphabetic character in the range “a” through “z”
[^a-z]	Specifies a range of characters that does not match any character in the range: <ul style="list-style-type: none"> [^a-z] matches any character that is not in the range “a” through “z”
[:alpha:]	Matches an alphabetic character.
[^:alpha:]	Matches a nonalphabetic character.
[:alnum:]	Matches an alphanumeric character.
[^:alnum:]	Matches a nonalphanumeric character.
[:ascii:]	Matches an ASCII character. Equivalent to [0–177].
[^:ascii:]	Matches a non-ASCII character. Equivalent to [^0–177].
[:blank:]	Matches a blank character.
[^:blank:]	Matches a non-blank character.
[:cntrl:]	Matches a control character.
[^:cntrl:]	Matches a character that is not a control character.
[:digit:]	Matches a digit. Equivalent to \d.
[^:digit:]	Matches a non-digit character. Equivalent to \D.
[:graph:]	Is a visible character, excluding the space character. Equivalent to [[:alnum:]][[:punct:]].
[^:graph:]	Is not a visible character. Equivalent to [^[:alnum:]][[:punct:]].
[:lower:]	Matches lowercase characters.
[^:lower:]	Does not match lowercase characters.
[:print:]	Prints a string of characters.
[^:print:]	Does not print a string of characters.

Metacharacter	Description
[:punct:]	Matches a punctuation character or a visible character that is not a space or alphanumeric.
[[:^punct:]]	Does not match a punctuation character or a visible character that is not a space or alphanumeric.
[:space:]	Matches a space. Equivalent to \s.
[[:^space:]]	Does not match a space. Equivalent to \S.
[:upper:]	matches uppercase characters.
[[:^upper:]]	does not match uppercase characters.
[:word:]	Matches a word. Equivalent to \w.
[[:^word:]]	Does not match a word. Equivalent to \W.
[:xdigit:]	Matches a hexadecimal character.
[[:^xdigit:]]	Does not match a hexadecimal character.

Look-Ahead and Look-Behind Behavior

Look-ahead and look-behind are ways to look ahead or behind a match to see whether a particular text occurs. The text that is found with look-ahead or look-behind is not included in the match that is found. For example, if you want to find names that end with “Jr.”, but you do not want “Jr.” to be part of the match, you could use the regular expression `/(?=\Jr\.)`. For the value “John Wainright Jr.”, the regular expression will find “John Wainright” as a match because it is followed by “Jr.”

Table 7.7 Look-Ahead and Look-Behind Metacharacters

Metacharacter	Description
(?=...)	Specifies a zero-width, positive, look-ahead assertion. For example, in the expression <code>regex1 (?=regex2)</code> , a match is found if both <code>regex1</code> and <code>regex2</code> match. The expression <code>regex2</code> is not included in the final match.
(?!...)	Specifies a zero-width, negative, look-ahead assertion. For example, in the expression <code>regex1 (!regex2)</code> , a match is found if <code>regex1</code> matches and <code>regex2</code> does not match. The expression <code>regex2</code> is not included in the final match.
(?<=...)	Specifies a zero-width, positive, look-behind assertion. For example, in the expression <code>(?<=regex1) regex2</code> , a match is found if both <code>regex1</code> and <code>regex2</code> match. <code>regex1</code> is not included in the final match. Works with fixed-width look-behind only.
(?<!...)	Specifies a zero-width, negative, look-behind assertion. Works with fixed-width look-behind only.

Comments and Inline Modifiers

The metacharacters in this table contain a question mark as the first element inside the parentheses. The characters after the question mark indicate the extension.

Table 7.8 Comments and Inline Modifiers

Metacharacter	Description
(?#text)	Specifies a comment in which the text is ignored.
(?imsx)	Specifies one or more embedded pattern-matching modifiers. If the pattern is case insensitive, you can use (?i) at the front of the pattern. An example is <code>\$pattern="(?i) foobar";</code> . Letters that appear after a hyphen (-) turn the modifiers off.

Selecting the Best Condition By Using Combining Operators

The elementary regular expressions (for example, \a and \w) that are described in the preceding tables can match at most one substring at the given position in the input string. However, operators that perform combining in typical regular expressions combine elementary metacharacters to create more complex patterns. In an ambiguous situation, these operators can determine the best match or the worst match. The match that is the best is always chosen.

Table 7.9 Best Match Using Combining Operators

Metacharacter	Description
ST	In the following example, specifies that AB and A'B', and A and A' are substrings that can be matched by S, and that B and B' are substrings that can be matched by T: <ul style="list-style-type: none"> • If A is a better match for S than A', then AB is a better match than A'B'. • If A and A' coincide, then AB is a better match than AB' if B is a better match for T than B'.
S T	Specifies that when S can match, it is a better match than when only T can match. The ordering of two matches for S is the same as for S. Similarly, the ordering of two matches for T is the same as for T.
S{repeat-count}	Matches as SSS . . . S (repeated as many times as necessary).
S{min,max}	Matches as S{max} S{max-1} . . . S{min+1} S{min}.
S{min,max}?	Matches as S{min} S{min+1} . . . S{max-1} S{max}.
S?, S*, S+	Same as S{0,1}, S{0, big-number}, S{1, big-number}, respectively.
S??, S*?, S+	Same as S{0,1}?, S{0, big-number}?, S{1, big-number}?, respectively.

Metacharacter	Description
(?=S), (?<=S)	Considers the best match for S. (This is important only if S has capturing parentheses and back references are used elsewhere in the whole regular expression.)
(?!S), (?<!S)	Unnecessary to describe the ordering for this grouping operator because the only important thing is whether S can match.

Examples

Example 1

If productCode is **CRE54X39-2013**, then the following expressions all evaluate to True:

```
%brm_prxmatch(productCode, /X39/)
%brm_prxmatch(productCode, /[[:^alnum:]]/) /* Matches the underscore. */
%brm_prxmatch(productCode, /[EBR]/)      /* Matches the E and the R.*/
```

The following expressions both evaluate to False:

```
%brm_prxmatch(productCode, /^2013/) /* The string 2013 is at the end of */
/* productCode, not the beginning. */
%brm_prxmatch(productCode, /[ebr]/) /* The characters are lower case, */
/* but productCode is uppercase. */
```

Example 2

You can also specify the Perl regular expression in a term. For example, suppose you want to determine whether an address contains a ZIP+4 code instead of a basic five-digit ZIP code. If the term Zip contains the regular expression `/\d{5}-\d{4}/`, then you can search the term Address for the ZIP+4 code pattern by specifying the following expression:

```
%brm_prxmatch(Address, Zip)
```

For example, if Address contains **123 Maple Street, Merrill, WI 54452-4321**, then the expression evaluates to True. If Address contains **123 Maple Street, Merrill, WI 54452**, then the expression evaluates to False.

%BRM_STARTSWITH

Determines whether a term's value begins with any of the specified character strings.

Restriction: You can specify this macro in condition expressions only, and you must specify it as the first part of the expression. You cannot specify multiple SAS Business Rules Manager macros in the same expression.

Syntax

```
%BRM_STARTSWITH (termToSearch, stringOrTerm <, stringOrTerm...>)
```

Required Arguments

termToSearch

the term whose value you want to search. This term must match the row or column term where you specify the macro.

stringOrTerm

a character string or a term whose value contains a character string that you want to search for. You can specify up to 50 terms or character strings to search for. Enclose character strings in quotation marks. Separate multiple strings or terms with commas.

Example

If the value of ordCode is **BT39X2JH**, then both of the following expressions evaluate to True:

```
%brm_startsWith(ordCode,"BT3")  
%brm_startsWith(ordCode,"B","T39")
```

If productCode is "BT7919T", then the following expression evaluates to False:

```
%brm_startsWith(ordCode,productCode,"2JH")
```


Appendix 1

DS2 Functions Supported by SAS Business Rules Manager

The following table lists the DS2 functions supported by SAS Business Rules Manager. For information about these functions see *SAS DS2 Language Reference* at <http://support.sas.com/documentation/onlinedoc/base/index.html>.

Table A1.1 DS2 Functions Supported by SAS Business Rules Manager

Category	Function	Description
Bitwise Logical Operations	BAND	Returns the bitwise logical AND of two arguments.
	BLSHIFT	Returns the bitwise logical left shift of two arguments.
	BNOT	Returns the bitwise logical NOT of an argument.
	BOR	Returns the bitwise logical OR of two arguments.
	BRSHIFT	Returns the bitwise logical right shift of two arguments.
	BXOR	Returns the bitwise logical EXCLUSIVE OR of two arguments.
Character	BYTE	Returns one character in the ASCII or the EBCDIC collating sequence.
	CAT	Does not remove leading or trailing blanks, and returns a concatenated character or numeric string.
	CATS	Removes leading and trailing blanks, and returns a character or numeric concatenated string.
	CATT	Removes trailing blanks, and returns a concatenated character or numeric string.

Category	Function	Description
Character	CATX	Removes leading and trailing blanks, inserts delimiters, and returns a concatenated character string.
	COALESCEC	Returns the first non-null or nonmissing value from a list of character arguments.
	COLLATE	Returns a character string in ASCII or EBCDIC collating sequence.
	COMPBL	Removes multiple blanks from a character string.
	COMPRESS	Returns a character string with specified characters removed from the original string.
	DEQUOTE	Removes matching single quotation marks from a character string that begins with a single quotation mark, and deletes all characters to the right of the closing quotation mark.
	INDEX	Searches a character expression for a string of characters, and returns the position of the string's first character for the first occurrence of the string.
	INDEXC	Searches a character expression for specified characters and returns the position of the first occurrence of any of the characters.
	INDEXW	Searches a character expression for a string that is specified as a word, and returns the position of the first character in the word.
	LEFT	Left aligns a character expression.
	LENGTH	Returns the length of a character string, excluding trailing blanks, and returns a 0 for a blank character string.
	LENGTHC	Returns the length of a character string, including trailing blanks.
	LENGTHM	Returns the amount of memory, in characters, that is allocated for a character string.
	LOWCASE	Converts all letters in a character expression to lowercase.
	QUOTE	Adds double quotation marks to a character value.
	RANK	Returns the position of a character in the ASCII or EBCDIC collating sequence.
	REPEAT	Repeats a character expression.
	REVERSE	Reverses a character expression.
	RIGHT	Right aligns a character expression.

Category	Function	Description
Character	SCAN	Returns the nth word from a character expression.
	STRIP	Returns a character string with all leading and trailing blanks removed.
	SUBSTR	Returns a substring, allowing a result with a length of zero.
	TRANSLATE	Replaces specific characters in a character expression.
	TRANWRD	Replaces or removes all occurrences of a word in a character string.
	TRIM	Removes trailing blanks from a character expression.
	UPCASE	Converts all letters in an argument to uppercase.
	VERIFY	Returns the position of the first character that is unique to an expression.
	WHICHC	Returns the first position of a character string from a list of character strings.
Date and Time	DATE	Returns the current date as a SAS date value.
	DATEJUL	Converts a Julian date to a SAS date value.
	DATEPART	Extracts the date from a SAS datetime value.
	DATETIME	Returns the current date and time of day as a SAS datetime value.
	DAY	Returns the day of the month from a SAS date value.
	DHMS	Returns a SAS datetime value from date, hour, minute, and second values.
	HMS	Returns a SAS time value from hour, minute, and second values.
	HOURL	Returns the hour from a SAS time or datetime value.
	INTCK	Returns the number of interval boundaries of a given kind that lie between two SAS dates, times, or timestamp values encoded as DOUBLE.
	INTDT	Specifies the number of days to add to a DATE value.
	INTNX	Increments a SAS date, time, or datetime value encoded as a DOUBLE, and returns a SAS date, time, or datetime value encoded as a DOUBLE.
	INTTS	Specifies the number of seconds to add to a TIMESTAMP value.
	JULDATE	Returns the Julian date from a SAS date value.
	JULDATE7	Returns a seven-digit Julian date from a SAS date value.

Category	Function	Description
Date and Time	MDY	Returns a SAS date value from month, day, and year values.
	MINUTE	Returns the minute from a SAS time or datetime value.
	MONTH	Returns a number that represents the month from a SAS date value.
	QTR	Returns the quarter of the year from a SAS date value.
	SECOND	Returns the second from a SAS time or datetime value.
	TIME	Returns the current time of day as a numeric SAS time value.
	TIMEPART	Extracts a time value from a SAS datetime value.
	TODAY	Returns the current date as a numeric SAS date value.
	WEEKDAY	From a SAS date value, returns an integer that corresponds to the day of the week.
	YEAR	Returns the year from a SAS date value.
	YYQ	Returns a SAS date value from year and quarter year values.
Descriptive Statistics	CSS	Returns the corrected sum of squares.
	CV	Returns the coefficient of variation.
	GEOMEAN	Returns the geometric mean.
	GEOMEANZ	Returns the geometric mean, using zero fuzzing.
	HARMEAN	Returns the harmonic mean.
	HARMEANZ	Returns the harmonic mean, using zero fuzzing.
	IQR	Returns the interquartile range.
	KURTOSIS	Returns the kurtosis.
	LARGEST	Returns the kth largest non-null or nonmissing value.
	MAX	Returns the largest value from a list of arguments.
	MEAN	Returns the arithmetic mean (average) of the non-null or nonmissing arguments.
	MIN	Returns the smallest value.
	N	Returns the number of non-null or nonmissing numeric values.

Category	Function	Description
Descriptive Statistics	NMISS	Returns the number of null and SAS missing numeric values.
	ORDINAL	Orders a list of values, and returns a value that is based on a position in the list.
	PCTL	Returns the percentile that corresponds to the percentage.
	RANGE	Returns the difference between the largest and the smallest values.
	RMS	Returns the root mean square.
	SKEWNESS	Returns the skewness.
	SMALLEST	Returns the kth smallest non-null or nonmissing value.
	STD	Returns the standard deviation.
	STDERR	Returns the standard error of the mean.
	SUM	Returns the sum of the non-null or nonmissing arguments.
	USS	Returns the uncorrected sum of squares.
	VAR	Returns the variance.
Mathematical	ABS	Returns the absolute value of a numeric value expression.
	COALESCE	Returns the first non-null or nonmissing value from a list of numeric arguments.
	DIGAMMA	Returns the value of the digamma function.
	EXP	Returns the value of the e constant raised to a specified power.
	GAMMA	Returns the value of the gamma function.
	GCD	Returns the greatest common divisor for a set of integers.
	LCM	Returns the least common multiple for a set of integers.
	LGAMMA	Returns the natural logarithm of the Gamma function.
	LOG	Returns the natural logarithm (base e) of a numeric value expression.
	LOG10	Returns the base-10 logarithm of a numeric value expression.
	LOG2	Returns the base 2 logarithm of a numeric value expression.

Category	Function	Description
Mathematical	MOD	Returns the remainder from the division of the first argument by the second argument, fuzzed to avoid most unexpected floating-point results.
	MODZ	Returns the remainder from the division of the first argument by the second argument, using zero fuzzing.
	POWER	Returns the value of a numeric value expression raised to a specified power.
	SIGN	Returns a number that indicates the sign of a numeric value expression.
	SQRT	Returns the square root of a value.
	TRIGAMMA	Returns the value of the trigamma function.
	WHICHN	Returns the first position of a number from a list of numbers.
Numeric	CAT	Does not remove leading or trailing blanks, and returns a concatenated character or numeric string.
	CATS	Removes leading and trailing blanks, and returns a character or numeric concatenated string.
	CATT	Removes trailing blanks, and returns a concatenated character or numeric string.
	CATX	Removes leading and trailing blanks, inserts delimiters, and returns a concatenated character string.
Special	INPUTN	Enables you to specify a numeric informat at run time.
	MISSING	Returns a number that indicates whether the argument contains a missing value.
	NULL	Returns a 1 if the argument is null and a 0 if the argument is not null.
	PUT	Returns a value using a specified format.
	SLEEP	For a specified period of time, suspends the execution of a program that invokes this function.
Trigonometric	ARCOS	Returns the arccosine in radians.
	ARSIN	Returns the arcsine in radians.
	ATAN	Returns the arctangent in radians.
	ATAN2	Returns the arctangent of the x and y coordinates of a right triangle, in radians.
	COS	Returns the cosine in radians.

Category	Function	Description
Trigonometric	COSH	Returns the hyperbolic cosine in radians.
	SIN	Returns the trigonometric sine.
	SINH	Returns the hyperbolic sine.
	TAN	Returns the tangent.
	TANH	Returns the hyperbolic tangent.
Truncation	CEIL	Returns the smallest integer greater than or equal to a numeric value expression.
	CEILZ	Returns the smallest integer that is greater than or equal to the argument, using zero fuzzing.
	FLOOR	Returns the largest integer less than or equal to a numeric value expression.
	FLOORZ	Returns the largest integer that is less than or equal to the argument, using zero fuzzing.
	FUZZ	Returns the nearest integer if the argument is within 1E-12 of that integer.
	INT	Returns the integer value, fuzzed to avoid unexpected floating-point results.
	INTZ	Returns the integer portion of the argument, using zero fuzzing.
	ROUND	Rounds the first argument to the nearest multiple of the second argument, or to the nearest integer when the second argument is omitted.
	ROUNDE	Rounds the first argument to the nearest multiple of the second argument, and returns an even multiple when the first argument is halfway between the two nearest multiples.
	ROUNDZ	Rounds the first argument to the nearest multiple of the second argument, using zero fuzzing.
	TRUNC	Truncates a numeric value to a specified length.

Glossary

business entity

an object in your business domain. An entity has one or more terms, which are attributes of an entity. For example, the business entity could be the customer, and the associated terms could be the name, account number, account type, and so on.

business rule

a statement of business logic that specifies conditions to be evaluated and actions to be taken if those conditions are satisfied.

key

See lookup key

lookup key

a value that uniquely identifies a specific record and its order among other records in a database or table.

lookup table

a table that contains lookup keys and their corresponding values.

lookup value

the value that is associated with a lookup key in a lookup table.

publish

to register a business rule flow in a SAS metadata repository.

rule

See business rule

rule flow

a logical collection of multiple rule sets that define multiple conditions and actions. Rule flows can be tested and deployed as SAS programs and services that process input data, which contain conditions, in order to create output data, which contain actions.

rule set

a logical group of business rules.

term

an attribute of a business entity. Terms might or might not have a list of valid values. For example, a customer entity might have terms such as account type or age. Valid values for the account type term might include "commercial" or "personal."

vocabulary

the set of business entities that define your business domain.

Index

Special Characters

&BRMPPrimaryEntityKey macro variable
52, 54

&BRMPPrimaryTransactionDTTM macro
variable 52, 54

%BRM_CONTAINS macro 57

%BRM_ENDSWITH macro 58

%BRM_MATCHANY macro 59

%BRM_PRXMATCH macro 59

%BRM_STARTSWITH macro 68

A

action expressions 28

B

Boolean values

contrast with string values 18

entering 36

business rules

conditional processing 32

creating 30

deleting rules 43

deleting text in a rule 43

description of 28

example 28

executing 8

reordering in a rule set 41

swapping positions of two rules 42

BY groups 48

C

category pane 3

category pane, managing columns

colors, application window preference 6

condition expressions 28

conditional processing 32

copying

rules 42

text within a rule 42

D

date values, entering 36

datetime values, entering 36

decision table 4

domain values 19

E

ELSE operator 32

entities

creating 18

deleting 20

description of 17

editing 20

excluding terms from input data 19

excluding terms from output data 19

Expression Editor 33

expressions

action, description of 28

condition, description of 28

deleting 43

entering data values in 36

examples 39

missing values in 38

operators added by SAS Business Rules

Manager 38

operators for use in 37

terms added by SAS Business Rules

Manager 38

using functions in 37

using macros in 38

validating 41

F

Final, rule flow section 48

focus indicator, preference 6

folders

- creating 9
- deleting 10
- functions in expressions 37

G

- Group End, rule flow section 48
- Group Start, rule flow section 48

I

- IF operator 32
- Init, rule flow section 48

L

- locale, preference 6
- Lookup function 25
 - in the Expression Editor 35
- lookup key 24
 - verifying 25
- lookup table
 - creating 24
 - deleting 25
 - description 23
 - refreshing 25
- lookup value 24
 - retrieving 26
- LookupValue function 26
 - in the Expression Editor 35

M

- macro variables
 - &BRMPPrimaryEntityKey 52, 54
 - &BRMPPrimaryTransactionDTTM 52, 54
- macros
 - %BRM_CONTAINS 57
 - %BRM_ENDSWITH 58
 - %BRM_MATCHANY 59
 - %BRM_PRXMATCH 59
 - %BRM_STARTSWITH 68
 - searching term values 57
- macros in expressions 38
- Main, rule flow section 48
- metacharacters
 - Perl regular expressions 60
- missing values in expressions 38

N

- navigation pane 3

O

- operators 32
- operators in expressions 37
- OR operator 32
- order of rules, changing 41

P

- pattern matching
 - See [macros](#)
- Perl regular expressions
 - class groupings 64
 - combining operators 67
 - comments 67
 - general constructs 60
 - inline modifiers 67
 - look-ahead metacharacters 66
 - look-behind metacharacters 66
 - metacharacters 60
 - repetition factors 63
- preferences
 - global 6
 - SAS Business Rules Manager
 - preferences 6
 - setting 6
- PRX
 - See [Perl regular expressions](#)
- publishing rule flows 54

R

- rule flows
 - BY groups 48
 - changing name or description 51
 - complex 48
 - copying 51
 - creating 48
 - deleteing rule sets from 51
 - deleting 52
 - description of 47
 - displaying rule sets used in 49
 - moving 51
 - opening 49
 - publishing 54
 - reordering rule sets in 50
 - sections, in complex rule flows 48
 - simple 48
 - viewing terms in 50
- rule flows, running a test 53
- rule flows, testing 52
 - input data 52
 - preprocessing code 52
 - viewing test results 53
- rule sets
 - changing name or description 43
 - changing rule order 41

- copying 44
- creating 29
- deleting 44
- deleting from rule flows 51
- description of 28
- moving 44
- opening 29
- opening multiple 49
- reordering in rule flows 50
- saving 45
- validating 41
- rules
 - See [business rules](#)
- rules database, basic steps for creating 7

S

- SAS Business Rules Manager
 - category pane 3
 - documentation 7
 - features 1
 - interface 3
 - navigation pane 3
 - preferences 6
 - tile pane 3
- string values, entering 36

T

- terms
 - creating 19
 - deleting 20
 - description of 17
 - domain values 19
 - editing 20
 - excluding terms from input data 19
 - excluding terms from output data 19
 - temporary 19
 - used in rule flows 50
- theme, preference 6
- tile pane 3

V

- validating expressions 41
- vocabularies
 - creating 18
 - deleting 20
 - description of 17
 - editing 20
 - tips for creating 18
- vocabulary pane 4

