

SAS[®] Business Rules Manager 1.2

User's Guide

Second Edition



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® Business Rules Manager 1.2*. Cary, NC: SAS Institute Inc.

SAS® Business Rules Manager 1.2

Copyright © 2012, SAS Institute Inc., Cary, NC, USA

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

For a hardcopy 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 Restricted Rights Notice: Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

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

Electronic book 2, January 2013

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

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 registered trademarks or trademarks of their respective companies.

Contents

<i>What's New in SAS Business Rules Manager 1.2</i>	v
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
Controlling the View in the Category Pane	6
Managing Preferences	6
Viewing Help and Documentation	7
Create and Publish Business Rules	7
Creating Folders	8
Delete a Folder	9
Chapter 2 • Managing Vocabularies	11
Introduction to Vocabularies, Entities, and Terms	11
Tips for Creating Entities and Terms	12
Create a Vocabulary	12
Create New Entities	12
Create New Terms	13
Specify Domain Values	13
Edit Existing Vocabularies, Entities, or Terms	14
Delete Vocabularies, Entities, or Terms	14
Chapter 3 • Using Lookup Tables And Functions	15
About Lookup Tables and Functions	15
Create a New Lookup Table	16
Refresh a Lookup Table	17
Delete a Lookup Table	17
Verify Lookup Keys (Lookup Function)	17
Get Lookup Values (LookupValue Function)	18
Chapter 4 • Managing Rules and Rule Sets	19
About Rules and Expressions	20
About Rule Sets	20
Create a New Rule Set	20
Open an Existing Rule Set	21
Adding Rules to the Rule Set	21
Create a Rule That Always Executes	31
Validate the Expressions in a Rule Set	32
Change the Order of Rules in a Rule Set	32
Copy Rules and Expressions	33
Delete Rules and Expressions	34
Edit the Name and Description for a Rule Set	35
Copy a Rule Set	35
Move a Rule Set to a Different Folder	36
Delete a Rule Set	36
Save a Rule Set	36
Chapter 5 • Creating and Publishing Rule Flows	37

Introduction to Rule Flows	37
Simple Rule Flows, Complex Rule Flows, and BY Groups	38
Create a Rule Flow	38
Open an Existing Rule Flow	39
Open Rule Sets from the Rule Flow Editor	39
Change the Order of the Rule Sets	39
View the Terms Used in a Rule Flow	40
Edit Name and Description for a Rule Flow	40
Copy a Rule Flow	40
Move a Rule Flow to a Different Folder	41
Delete a Rule Set from a Rule Flow	41
Delete a Rule Flow	41
Testing a Rule Flow	41
Publish a Rule Flow	43
Chapter 6 • SAS Business Rules Manager Macro Reference	45
About SAS Business Rules Manager Macros	45
Dictionary	45
Glossary	57
Index	59

What's New in SAS Business Rules Manager 1.2

Overview

The first maintenance release of SAS Business Rules Manager 1.2 includes the following new features and enhancements:

- lookup tables
- the ability to import and export vocabularies, rule sets, and rule flows
- conditional logic within rule sets
- complex rule flows with group processing
- interactive rule flow testing
- macros for searching term values
- vertical decision table format plus other user interface enhancements

Lookup Tables

SAS Business Rules Manager now supports lookup tables, which contain keys and their corresponding values. For example, a lookup table might contain keys that are country codes and values that are full country names. You can use the key to retrieve the full country name.

To create a lookup table, you create a comma-delimited file (CSV file) that contains the lookup data by using an application such as Microsoft Excel. Next, you import this data into SAS Business Rules Manager. You can view the contents of the lookup table in SAS Business Rules Manager, and you can refresh the lookup table as needed by importing an updated CSV file.

You can use the new Lookup function to verify that a key exists in a lookup table. You can use the new LookupValue function to retrieve the value that is associated with a specified key.

See [Chapter 3, “Using Lookup Tables And Functions,”](#) on page 15 for more information.

Importing and Exporting Vocabularies, Rule Sets, and Rule Flows

SAS Business Rules Manager provides macros for importing and exporting vocabularies, rules sets, and rule flows. You can define vocabularies, rule sets, or rule flows in CSV files, and then use the macros to import the data in those files. You can also use macros to export to CSV files the vocabularies, rule sets, and rules flows that have been created or edited in SAS Business Rules Manager. You can then edit these CSV files and re-import them as needed.

The macros for importing and exporting data are documented in *SAS Business Rules Manager: Administrator's Guide*.

Conditional Logic within Rule Sets

You can use the operators IF, ELSE, and OR to control which rules are executed within a rule set. By default, a rule is assigned the operator IF, and its action is executed only if its condition evaluates to true. If you assign the operator ELSE to a rule, its condition is evaluated only if the previous rule's condition evaluated to false. The OR operator enables you to assign the same action to multiple conditions. See [“Controlling Which Conditions Are Evaluated” on page 23](#) for more information.

Intermediate Terms

When you are defining vocabulary terms, you can identify terms as intermediate (temporary) terms by selecting the option **Exclude from input**. Terms that are excluded from input do not have to be mapped to an input data set for a rule flow. Such terms cannot be mapped to input table columns in a test or production job.

Complex Rule Flows with Group Processing

SAS Business Rules Manager now provides the ability to define complex rule flows. A complex rule flow has 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.

With complex rule flows, you can also add group processing. If you specify terms that serve as *BY-group terms*, then results are calculated for each group of input records that have the same value for all of the BY-group terms. Output is generated for each group instead of for each input record.

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

See [“Simple Rule Flows, Complex Rule Flows, and BY Groups”](#) on page 38 for more information.

Interactive Rule Flow Testing

SAS Business Rules Manager now provides the ability to test rule flows before you deploy them. You can enter preprocessing code that you want to run before the rule flow is run. For example, you might need SAS Business Rules Manager to run initialization code before it runs the rule flow. Then, you can select an input data set and run the rule flow. 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. You can choose to save the rule flow results and rule-fired data in SAS data sets that can then be used as input to other SAS applications.

See [“Testing a Rule Flow”](#) on page 41 for more information.

Macros for Searching Term Values

SAS Business Rules Manager provides five new macros for searching term values for character strings. You can specify these macros in condition expressions.

`%BRM_CONTAINS`

determines whether a term’s value contains any of the specified character strings

`%BRM_ENDSWITH`

determines whether a term’s value ends with any of the specified character strings

`%BRM_MATCHANY`

determines whether a term’s value matches any of a set of character strings

`%BRM_PRXMATCH`

determines whether a term’s value contains a string that matches the specified Perl regular expression

`%BRM_STARTSWITH`

determines whether a term’s value begins with any of the specified character strings

See [Chapter 6, “SAS Business Rules Manager Macro Reference,”](#) on page 45 for more information.

Vertical Decision Table

SAS Business Rules Manager now provides a vertical view for decision tables. Terms are listed on the left side of the table instead of at the top, and rules are listed left-to-right instead of top-to-bottom.

User Interface Enhancements

There are several enhancements to the user interface that make it easier to create and edit data in the first maintenance release of SAS Business Rules Manager. These enhancements provide you with the following capabilities:

- copy rules sets and rule flows within a folder
- move rule sets and rule flows from one folder to another
- drag multiple terms from the vocabulary tree into the decision table at the same time
- manage and sort the columns that are displayed in category views
- open the Expression Editor more easily by selecting the Expression Editor icon or selecting the **Expression Editor** option from the popup menu
- control which columns are displayed in category views, and control how these columns are sorted
- control the default sort order for SAS Business Rules Manager folders

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	3
Exploring the User Interface	3
Controlling the View in the Category Pane	6
Managing Preferences	6
Set Global Preferences	6
SAS Business Rules Manager Preferences	7
Viewing Help and Documentation	7
Create and Publish Business Rules	7
Creating Folders	8
Create New Top-Level Folders	8
Create New Subfolders	8
Delete a Folder	9

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.

Log On to SAS®

sas THE POWER TO KNOW.

About

User ID:
myuserID

Password:
.....

Log On

Copyright © 2002 - 2012, SAS Institute Inc., Cary, NC, USA. All Rights Reserved

2. 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.

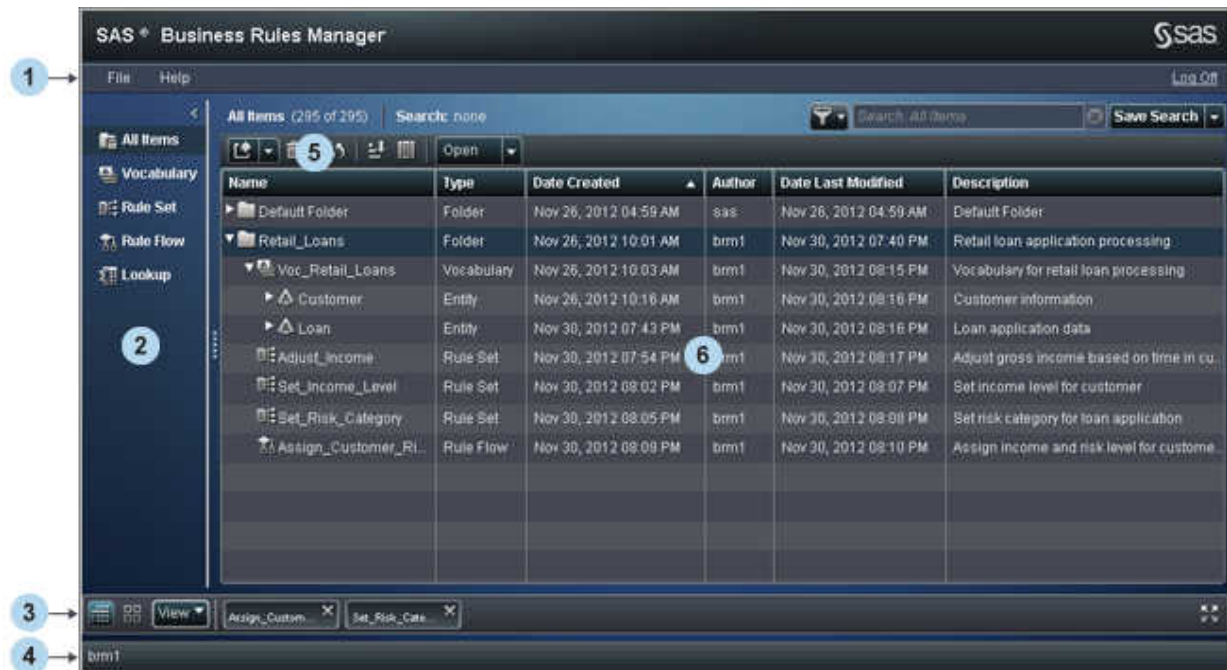
By default, if SAS Business Rules Manager is open for 30 minutes with no activity, then a warning message appears. You are logged off unless you choose to continue the session.

Exploring the User Interface

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

Each of the category panes are similar. The following figure shows the All Items category pane.

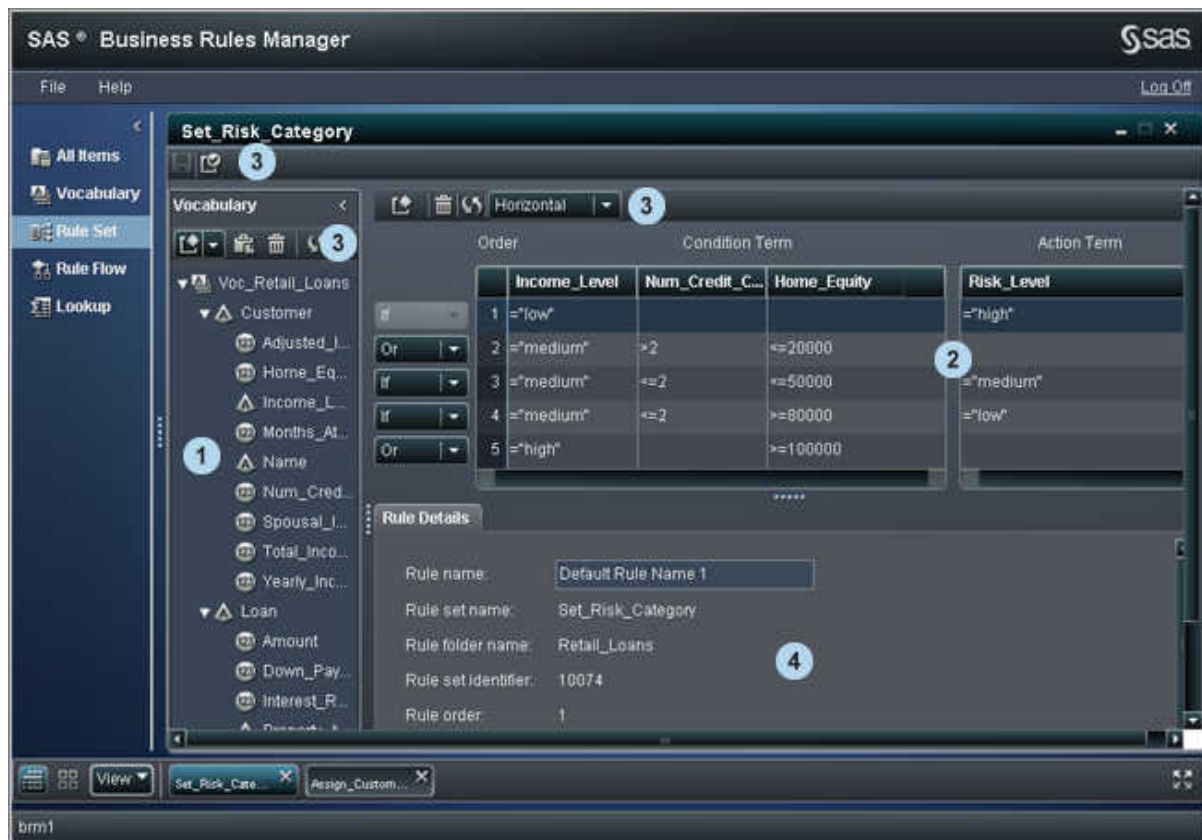
Figure 1.1 All Items Category Pane



- 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 **X** 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 identity that is logged on to the metadata server.
- 5 Panes and tiles can have their own toolbars, so you might see multiple toolbars when the window contains multiple tiles.
- 6 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 “Controlling the View in the Category Pane” on page 6 for information about controlling the display of information in the category pane.

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 three 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.

Controlling the View in the 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 .

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.

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**.

Folder Sort Order

controls the default order in which folders are sorted. If you change this preference, you must click  to refresh the view.

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 folders where you want to save the business rules. See “[Creating Folders](#)” for instructions.
2. Create vocabularies. See “[Create a Vocabulary](#)” for more information.

3. Create entities and terms. See “Create New Entities” on page 12 and “Create New Terms” on page 13 for more information.
4. Create rule sets and rules. See Chapter 4, “Managing Rules and Rule Sets,” on page 19 for more information.
5. Create rule flows. See “Create a Rule Flow” on page 38 for more information.
6. (Optional) Test rule flows. See “Test a Rule Flow” on page 42 for more information.
7. Publish rule flows. See “Publish a Rule Flow” on page 43 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.

Creating Folders

Create New Top-Level Folders


By default, SAS Business Rules Manager defines a top-level folder named **Default folder**. You can rename this folder for your own use.

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.
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.
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 .

Chapter 2

Managing Vocabularies

Introduction to Vocabularies, Entities, and Terms	11
Tips for Creating Entities and Terms	12
Create a Vocabulary	12
Create New Entities	12
Create New Terms	13
Specify Domain Values	13
Edit Existing Vocabularies, Entities, or Terms	14
Delete Vocabularies, Entities, or Terms	14

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 Navigation pane, select either **All Items** or **Vocabulary**.
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. Names can contain only alphanumeric characters and underscores.
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. Names can contain only alphanumeric characters and underscores.
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. Names can contain only alphanumeric characters and underscores.
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 13](#) for more information.
7. (Optional) Select **Exclude from input** to exclude the term from the data that is required to be mapped to an input data set for a rule flow. Terms that are excluded from input cannot be mapped to input table columns in a test or production job.
8. 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 2.1 Examples of Domain Values

Data Type	Domain Type	Examples
String	Discrete	<code>high risk;low risk</code> <code>"d'oscail";"d'fhill"</code>

Data Type	Domain Type	Examples
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 3

Using Lookup Tables And Functions

About Lookup Tables and Functions	15
Create a New Lookup Table	16
Refresh a Lookup Table	17
Delete a Lookup Table	17
Verify Lookup Keys (Lookup Function)	17
Get Lookup Values (LookupValue Function)	18

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 3.1 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
14	LB	Lebanon
15	MG	Madagascar

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	India



In a lookup table, each *lookup key* is associated with a *lookup value*. Lookup keys must be unique within each lookup table.

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 Navigation pane, select either **All Items** or **Lookup**.
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 a folder.
4. (Optional) Enter a description for the new rule set. 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. (Optional) In the Refresh Lookup Table dialog box, edit the table.
3. Click , and select the CSV file that contains the lookup data.
4. Click **OK**.

Delete a Lookup Table

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

To delete a lookup table:

1. In the Navigation pane, select either **All Items** or **Lookup**.
2. Select the lookup table and click . Alternatively, you can right-click on the rule set, 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.

Order	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.

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

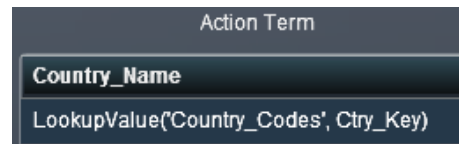
the name of the lookup table that you want to search.

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)
```



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.

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

Chapter 4

Managing Rules and Rule Sets

About Rules and Expressions	20
About Rule Sets	20
Create a New Rule Set	20
Open an Existing Rule Set	21
Adding Rules to the Rule Set	21
Create a New Rule	21
Controlling Which Conditions Are Evaluated	23
Using the Expression Editor	24
Punctuation for Data Values	27
Operators for Use in Expressions	28
Using Functions in Expressions	28
Using SAS Business Rules Manager Macros in Expressions	29
Working with Missing Values	29
Terms and Operators Added by SAS Business Rules Manager	29
Examples of Expressions	29
Create a Rule That Always Executes	31
Validate the Expressions in a Rule Set	32
Change the Order of Rules in a Rule Set	32
Move a Rule to a New Position in a Rule Set	32
Swap Two Rules	33
Copy Rules and Expressions	33
Copy an Entire Rule	33
Copy Text within a Rule	33
Delete Rules and Expressions	34
Delete a Rule	34
Delete or Cut and Paste Text within a Rule	34
Edit the Name and Description for a Rule Set	35
Copy a Rule Set	35
Move a Rule Set to a Different Folder	36
Delete a Rule Set	36
Save a Rule Set	36

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:

Order	Condition Term	Action Term
	<code>customer_debt</code>	<code>app_status</code>
1	<code>>customer_assets</code>	<code>= 'Declined'</code>

A single rule can have multiple terms, conditions, and actions. Action expressions are always assignment statements. Condition expressions and action expressions are required. A rule that does not have a condition expression is not included in the SAS code that is generated and run by SAS Business Rules Manager.

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 23](#) 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 Navigation pane, select either **All Items** or **Rule Set**.

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 **OK**. 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


To open an existing rule set, select either **All Items** or **Rule Set** in the Navigation pane. In the All Items or Rule Set category panes, 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**.

Adding Rules to the Rule Set

Create a New Rule

To create a new rule:

1. In the Navigation pane, select either **All Items** or **Rule Set**.
2. Create a new rule set or open an existing rule set. See [“Create a New Rule Set” on page 20](#) or [“Open an Existing Rule Set” on page 21](#) for more information.
SAS Business Rules Manager displays the decision table for the rule set.
3. If you created a new rule set, go to Step 4. If you opened an existing rule set, follow these steps to create a new rule:
 - a. Click  in the toolbar that is above the decision table.
 - b. In the New Rule dialog box, enter a name for the new rule. Rule names are limited to 100 characters and must be unique within a rule set.

- c. (Optional) Enter a description for the new rule. Descriptions are limited to 256 characters.
- d. (Optional) Change the order of the new rule. The rule order, in addition to the IF, ELSE, or OR keyword (see Step 6), controls the order in which rules are evaluated within the rule set.

Note: You can also change the order of the rules later by right-clicking on a rule and selecting either **Reorder** or **Swap**.

- e. Click **OK**. SAS Business Rules Manager adds a row (or column, in the vertical view) for the new rule to the decision table.
4. Add any terms that are needed by the new rule and that have not already been added to the decision table. You can add 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.


To add several terms at the same time, select the terms, then drag the terms onto the table, or use the right-click menu as you would for a single term. You can select multiple terms at the same time in one of two ways:

- To select a consecutive set of terms, click on the first term, hold down the Shift key, and then 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**.

5. 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.

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 29.) For example, suppose you enter the following rule in the decision table:

Order	Condition Term	Action Term
	Income_Level	Num_Credit_Cards
1	= 'high'	>=2 and <=3
		Risk_Level
		= 'low'

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

```
Condition: (Income_Level = 'high') AND (Num_Credit_Cards >= 2 and Num_Credit_Cards <= 3)
Action: Risk_Level = 'low'
```

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

- [“Using the Expression Editor”](#) on page 24
 - [“Punctuation for Data Values”](#) on page 27
 - [“Operators for Use in Expressions”](#) on page 28
 - Chapter 3, [“Using Lookup Tables And Functions,”](#) on page 15
 - [“Using Functions in Expressions”](#) on page 28
 - [“Using SAS Business Rules Manager Macros in Expressions”](#) on page 29
 - [“Working with Missing Values”](#) on page 29
 - [“Terms and Operators Added by SAS Business Rules Manager”](#) on page 29
 - [“Examples of Expressions”](#) on page 29
6. (Optional) Select the operator for the rule. The default operator is IF. See [“Controlling Which Conditions Are Evaluated”](#) on page 23 for more information.
 7. 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 32 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.

	Order	Condition Term	Action Term
If	1	Order_Quantity <=5	Offer_percent =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, and 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.

Order	Condition Term
	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.

Order	Condition Term
	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 though 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.

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

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 4.1 Expression Tab for a Condition Term



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

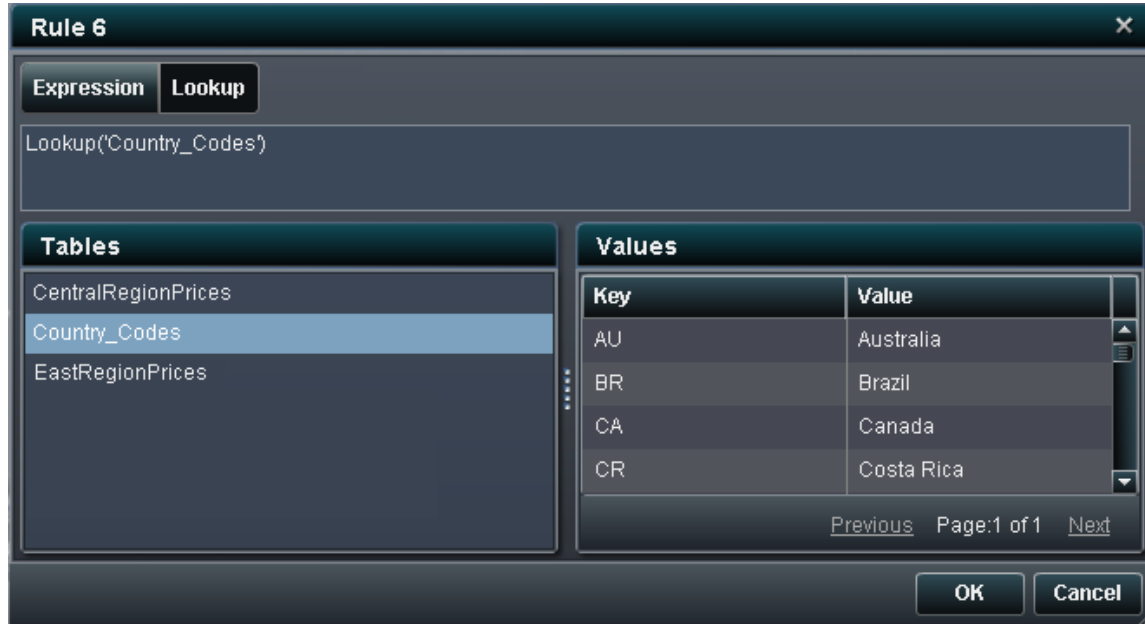
To define expressions that do not use the Lookup or LookupValue functions, enter the expression on the **Expression** tab. Click on the operators, vocabulary terms, and domain values as needed to add them to the expression. The Expression Editor builds the expression in the top field. To add numeric 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 27.](#)) 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 on the **Lookup**, double-click on the lookup table name that you want to specify in the function call, and click **OK**.

Display 4.2 Lookup Tab in the Expression Editor



For more information, see [“Verify Lookup Keys \(Lookup Function\)”](#) on page 17.

Specify LookupValue Function

To use the Expression Editor to enter the LookupValue function:

1. Click on 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**.

Display 4.3 LookupValue Tab in the Expression Editor

For more information, see [“Get Lookup Values \(LookupValue Function\)”](#) on page 18.

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 4.1 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 DDMMYYYY. 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 DDMMYYYY: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 \leq , \geq , or \neq . See “SAS Operators in Expressions” in Chapter 6 of SAS Language Reference: Concepts for more information about the operators shown in the table.

Table 4.2 Operators

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

Using Functions in Expressions

You can specify any function that is callable when the job that is using your business rules is run. This list can vary, depending on which products are installed at your site and whether you have defined your own custom functions. See *SAS Functions and CALL Routines: Reference* for information about all functions that are available with Base SAS.

If you are using lookup tables, then you can also specify the SAS Business Rules Manager Lookup and LookupValue functions.

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 6, “SAS Business Rules Manager Macro Reference,” on page 45](#) 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 4.3 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

Expression As Entered Into the Decision Table For Term X	Resulting Expression	Valid as a Condition Expression	Valid as an Action Expression
=term1	x=term1	Yes	Yes
5 or >100	x=5 or x>100	Yes	No See Note 1 .
^=5 and x<10	x^=5 and x<10	Yes	No See Note 1 .
^=5 or >=(100/4)	x^=5 or x>=(100/4)	Yes	No See Note 1 .
in (10,20,30)	x IN (10,20,30)	Yes	No See Note 1 .
not in ('med','high')	x NOT IN ('MED','HIGH')	Yes	No See Note 1 .
<'10JUN2012'd	x<'10JUN2012'd	Yes	No See Note 1 .
>'10JUN2012:17:00:00'dt	x>'10JUN2012:17:00:00'dt	Yes	No See Note 1 .
=ABS(-10)	x=ABS(-10)	Yes	Yes
=True	x=True	Yes	Yes
False	x=False	Yes	Yes
&myMacroVar	x=&myMacroVar	Yes See Note 2 .	Yes See Note 2 .
%EVAL(&myMacroVar)	x=%EVAL(&myMacroVar)	Yes See Note 2 .	Yes See Note 2 .
term1=5	x=term1=5	Yes See Note 3 .	Yes See Note 4 .
term1=3 or term2=5	x=term1=3 or term2=5	Yes See Note 5 .	Yes See Note 6 .
5 or (x>10 and <20)	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.		
>"mystring"	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.		

Expression As Entered Into the Decision Table For Term X	Resulting Expression	Valid as a Condition Expression	Valid as an Action Expression
+10	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**.

Create a Rule That Always Executes

You might need to create a rule that always executes. For example, you might need a rule that only initializes variables. These types of rules are frequently used with BY-groups. (See “[Simple Rule Flows, Complex Rule Flows, and BY Groups](#)” on page 38.)

To create a rule that always executes:

1. Create a term that you want to use only as a condition term for rules that always execute. The data type can be string, integer, or Boolean. You will use the term to create an expression that always evaluates to true. Consider naming the term according to its purpose. For example, you could name the term **ALWAYS_TRUE**. See “[Create New Terms](#)” on page 13 for more information.
2. When you create the term, select **Exclude from input** in the New Term dialog box.
3. Create a new rule as described in “[Create a New Rule](#)” on page 21. Be sure to make the following specifications:
 - a. For the condition terms, specify only the term that you created in Step 1.
 - b. For the condition expression, specify **=termName**. The expression for the term **ALWAYS_TRUE** would be **=ALWAYS_TRUE**.


- c. Verify that the operator for the rule is IF.
 - d. Add the action terms and expressions for the rule.
4. Click  to save the rule set.

Following this process creates the condition `if ALWAYS_TRUE=ALWAYS_TRUE`, which always evaluates to True, so the rule always executes.

The following image shows a rule that always executes and initializes two variables.

Order	Condition Term	Action Term	
	ALWAYS_TRUE	Count	Num_Records
If	1 =ALWAYS_TRUE	=0	=1

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**.

Order	Condition Term
	Months_At_Cu... Spousal_Income
1	<=3
2	>3 AND <=12
3	
4	=0
5	>0

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

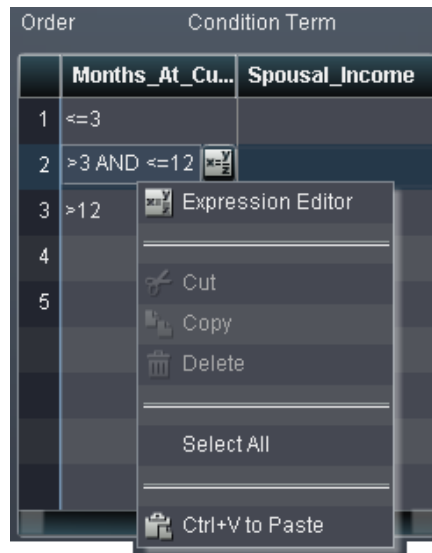
To copy and paste an entire rule:

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**.




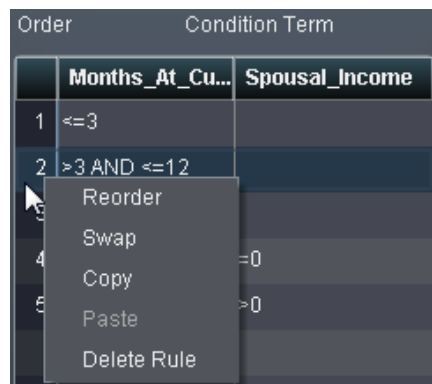
4. Click in the cell where you want to paste the text and press Ctrl+V.

Delete Rules and Expressions

Delete a Rule

You can delete a rule in one of two ways:

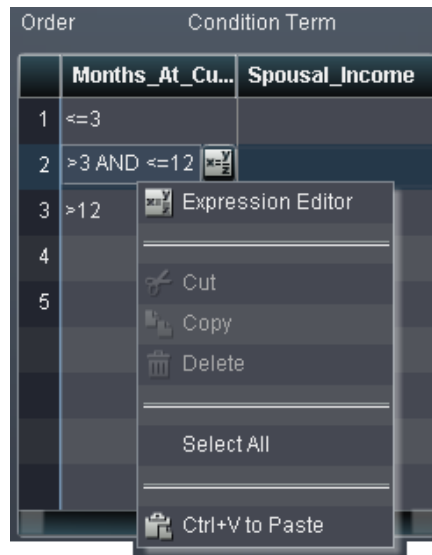
- Click on the order number of the rule that you want to delete, and click .
- Right-click on the order number of the rule, and select **Delete Rule**.



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 for a Rule Set

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

1. In the Navigation pane, select either **All Items** or **Rule Set**.
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.

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 Navigation pane, select either **All Items** or **Rule Set**.
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 Navigation pane, select either **All Items** or **Rule Set**.
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 Navigation pane, select either **All Items** or **Rule Set**.
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 32 for more information.

Chapter 5

Creating and Publishing Rule Flows

Introduction to Rule Flows	37
Simple Rule Flows, Complex Rule Flows, and BY Groups	38
Create a Rule Flow	38
Open an Existing Rule Flow	39
Open Rule Sets from the Rule Flow Editor	39
Change the Order of the Rule Sets	39
View the Terms Used in a Rule Flow	40
Edit Name and Description for a Rule Flow	40
Copy a Rule Flow	40
Move a Rule Flow to a Different Folder	41
Delete a Rule Set from a Rule Flow	41
Delete a Rule Flow	41
Testing a Rule Flow	41
Input Data for Rule Flow Tests	41
Specify Preprocessing Code	42
Test a Rule Flow	42
View Rule Flow Results and Test Output	42
Publish a Rule Flow	43

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 38](#) 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 Navigation pane, select either **All Items** or **Rule Flow**.
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 only alphanumeric characters and underscores. Rule flow names must be unique within a folder.
4. (Optional) Enter a description for the new rule flow. Descriptions are limited to 256 characters.
5. Click **OK**. 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 from the Resources pane 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 Navigation pane, select either **All Items** or **Rule Flow**.
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, select the rule sets that you want to open, and click .


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.
- Click on the order number of the rule set and drag 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 Navigation pane, select either **All Items** or **Rule Flow**.
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.

You can also change the description of a rule flow in the rule flow editor. See [“Open an Existing Rule Flow” on page 39](#) for information about opening the rule flow editor.

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 Navigation pane, select either **All Items** or **Rule Flow**.
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 Navigation pane, select either **All Items** or **Rule Flow**.
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 .

Delete a Rule Flow

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

To delete a rule flow:

1. In the Navigation pane, select either **All Items** or **Rule Flow**.
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 preprocessing code or initialization 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 registered as a SAS Base Library in metadata. Your user ID must have permission to access the data.

Specify Preprocessing Code

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

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

To assign a term of type string to the `&BRMPPrimaryEntityKey` macro variable, specify the variable as follows:

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

If the term contains an integer, specify the variable as follows:

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

The term that you specify for the `&BRMPPrimaryTransactionDTTM` macro should be a term of type datetime. Specify this macro variable as follows:

```
%let brmPrimaryTransactionDttm=termName;
```

Test a Rule Flow

To test a rule flow:

1. Open the rule flow that you want to test.
2. (Optional) On the **Test Preprocessing Code** tab, enter any SAS code, such as initialization code, that you want to run before the rule flow is run. See [“Specify Preprocessing Code” on page 42](#) for more information.
3. Click the **Test Rule Flow** tab.
4. Select the library and data set that contains the input data for the test, and click **Run**. 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 42](#) for more information.
5. (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 that is named according to the rule flow name and your user ID (`ruleFlowName_userID`). SAS Business Rules Manager saves rule-fired data in the data set `dcm_rule_action_fire` and test data such as the number of records that were processed in the data set `dcm_deployment_execution`. This data can then be used as input to other applications.

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 `&BRMPrimaryEntityKey` macro variable in the test preprocessing code.



TRANSACTION_DTTM

the value of the term that was specified with the `&BRMPrimaryTransactionDTTM` 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 [“Specify Preprocessing Code” on page 42](#) for information about the `&BRMPrimaryEntityKey` and `&BRMPrimaryTransactionDTTM` 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**.

Chapter 6

SAS Business Rules Manager Macro Reference

About SAS Business Rules Manager Macros	45
Dictionary	45
%BRM_CONTAINS	45
%BRM_ENDSWITH	46
%BRM_MATCHANY	47
%BRM_PRXMATCH	47
%BRM_STARTSWITH	56

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 6.1 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 6.2 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.
\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”

Metacharacter	Description
\B	Matches a non-word boundary: <ul style="list-style-type: none"> • <code>e\b</code> matches the “er” in “verb” • <code>e\b</code> does not match the “er” in “never”
\cA-\cZ	Matches a control character. For example, <code>\cX</code> matches the control character control-X.
\C	Matches a single byte.
\d	Matches a digit character that is equivalent to <code>[0–9]</code> .
\D	Matches a non-digit character that is equivalent to <code>[^0–9]</code> .
\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 <code>\E</code> 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 (<code>\$num</code>) 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 <code>[\f\n\r\t\v]</code> .
\S	Matches any character that is not a whitespace character and is equivalent to <code>[^\f\n\r\t\v]</code> .
\t	Matches a tab character.
\u	Specifies that the next character is uppercase.
\U	Specifies that the next string of characters, up to the <code>\E</code> 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> .

Metacharacter	Description
\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. For an example of how you can use these metacharacters, see “Replacing Text” in Chapter 1 of SAS Functions and CALL Routines: Reference.

For a description of these metacharacters, see Table A1.2, “Basic Perl Metacharacters and Their Descriptions,” in SAS Functions and CALL Routines: Reference.

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 6.3 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.

Metacharacter	Description
period (.)	Matches any single character except newline. To match any character including newline, use a pattern such as <code>[\n]</code> .
(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 <code>CALL PRXPOSN</code> . To retrieve the value of the capture buffer, use the <code>PRXPOSN</code> function. To match parentheses characters, use <code>\(</code> or <code>\)</code> .

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 6.4 Greedy Repetition Factors

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

The following table lists the lazy repetition metacharacters.

Table 6.5 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.
{ <i>n</i> }?	Matches exactly <i>n</i> times.
{ <i>n</i> ,}?	Matches a pattern at least <i>n</i> times.
{ <i>n</i> , <i>m</i> }?	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 6.6 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.

Metacharacter	Description
<code>[:ascii:]</code>	Matches an ASCII character. Equivalent to <code>[\0-177]</code> .
<code>[:^ascii:]</code>	Matches a non-ASCII character. Equivalent to <code>[\^0-177]</code> .
<code>[:blank:]</code>	Matches a blank character.
<code>[:^blank:]</code>	Matches a non-blank character.
<code>[:cntrl:]</code>	Matches a control character.
<code>[:^cntrl:]</code>	Matches a character that is not a control character.
<code>[:digit:]</code>	Matches a digit. Equivalent to <code>\d</code> .
<code>[:^digit:]</code>	Matches a non-digit character. Equivalent to <code>\D</code> .
<code>[:graph:]</code>	Is a visible character, excluding the space character. Equivalent to <code>[:alnum:][:punct:]</code> .
<code>[:^graph:]</code>	Is not a visible character. Equivalent to <code>[^\[:alnum:][:punct:]]</code> .
<code>[:lower:]</code>	Matches lowercase characters.
<code>[:^lower:]</code>	Does not match lowercase characters.
<code>[:print:]</code>	Prints a string of characters.
<code>[:^print:]</code>	Does not print a string of characters.
<code>[:punct:]</code>	Matches a punctuation character or a visible character that is not a space or alphanumeric.
<code>[:^punct:]</code>	Does not match a punctuation character or a visible character that is not a space or alphanumeric.
<code>[:space:]</code>	Matches a space. Equivalent to <code>\s</code> .
<code>[:^space:]</code>	Does not match a space. Equivalent to <code>\S</code> .
<code>[:upper:]</code>	matches uppercase characters.
<code>[:^upper:]</code>	does not match uppercase characters.
<code>[:word:]</code>	Matches a word. Equivalent to <code>\w</code> .
<code>[:^word:]</code>	Does not match a word. Equivalent to <code>\W</code> .
<code>[:xdigit:]</code>	Matches a hexadecimal character.
<code>[:^xdigit:]</code>	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 6.7 Look-Ahead and Look-Behind Metacharacters

Metacharacter	Description
(?=...)	Specifies a zero-width, positive, look-ahead assertion. For example, in the expression <i>regex1</i> (?!= <i>regex2</i>), a match is found if both <i>regex1</i> and <i>regex2</i> match. The expression <i>regex2</i> is not included in the final match.
(?!...)	Specifies a zero-width, negative, look-ahead assertion. For example, in the expression <i>regex1</i> (?! <i>regex2</i>), a match is found if <i>regex1</i> matches and <i>regex2</i> does not match. The expression <i>regex2</i> is not included in the final match.
(?<=...)	Specifies a zero-width, positive, look-behind assertion. For example, in the expression (?<= <i>regex1</i>) <i>regex2</i> , a match is found if both <i>regex1</i> and <i>regex2</i> match. <i>regex1</i> 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 6.8 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 6.9 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 T. Similarly, the ordering of two matches for T is the same as for S.
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.
(?=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")
```


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
42, 43

&BRMPPrimaryTransactionDTTM macro
variable 42, 43

%BRM_CONTAINS macro 45

%BRM_ENDSWITH macro 46

%BRM_MATCHANY macro 47

%BRM_PRXMATCH macro 47

%BRM_STARTSWITH macro 56

A

action expressions 20

B

Boolean values
contrast with string values 12
entering 27

business rules
conditional processing 23
creating 21
deleting rules 34
deleting text in a rule 34
description of 20
example 20
executing 8
reordering in a rule set 32
rules that always execute 31
swapping positions of two rules 33

BY groups 38

BY-group
rules that always execute 31

C

category pane 3

category pane, managing columns 6

colors, application window preference 6

condition expressions 20

conditional processing 23

copying
rules 33
text within a rule 33

D

date values, entering 27

datetime values, entering 27

decision table 5

domain values 13

E

ELSE operator 23

entities
creating 12
deleting 14
description of 11
editing 14

excluding terms from input data 13

Expression Editor 24

expressions
action, description of 20
condition, description of 20
deleting 34
entering data values in 27
examples 29
missing values in 29
operators added by SAS Business Rules
Manager 29
operators for use in 28
terms added by SAS Business Rules
Manager 29
using functions in 28
using macros in 29
validating 32

F

Final, rule flow section 38

focus indicator, preference 7
 folders
 creating 8
 deleting 9
 functions in expressions 28

G

Group End, rule flow section 38
 Group Start, rule flow section 38

I

IF operator 23
 Init, rule flow section 38

L

locale, preference 6
 Lookup function 17
 in the Expression Editor 26
 lookup key 16
 verifying 17
 lookup table
 creating 16
 deleting 17
 description 15
 refreshing 17
 lookup value 16
 retrieving 18
 LookupValue function 18
 in the Expression Editor 26

M

macro variables
 &BRMPPrimaryEntityKey 42, 43
 &BRMPPrimaryTransactionDTTM 42,
 43
 macros
 %BRM_CONTAINS 45
 %BRM_ENDSWITH 46
 %BRM_MATCHANY 47
 %BRM_PRXMATCH 47
 %BRM_STARTSWITH 56
 searching term values 45
 macros in expressions 29
 Main, rule flow section 38
 metacharacters
 Perl regular expressions 48
 missing values in expressions 29

N

navigation pane 3

O

operators 23
 operators in expressions 28
 OR operator 23
 order of rules, changing 32

P

pattern matching
 See macros
 Perl regular expressions
 class groupings 52
 combining operators 54
 comments 54
 general constructs 48
 inline modifiers 54
 look-ahead metacharacters 53
 look-behind metacharacters 53
 metacharacters 48
 repetition factors 51
 preferences
 global 6
 SAS Business Rules Manager
 preferences 7
 setting 6
 PRX
 See Perl regular expressions
 publishing rule flows 43

R

rule flows
 BY groups 38
 changing name or description 40
 complex 38
 copying 40
 creating 38
 deleteing rule sets from 41
 deleting 41
 description of 37
 displaying rule sets used in 39
 moving 41
 opening 39
 publishing 43
 reordering rule sets in 39
 sections, in complex rule flows 38
 simple 38
 viewing terms in 40
 rule flows, running a test 42
 rule flows, testing 41
 input data 41
 preprocessing code 42
 viewing test results 42
 rule sets
 changing name or description 35
 changing rule order 32

- copying 35
 - creating 20
 - deleting 36
 - deleting from rule flows 41
 - description of 20
 - moving 36
 - opening 21
 - opening multiple 39
 - reordering in rule flows 39
 - saving 36
 - validating 32
- 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 27
-
- T**
- terms
- creating 13
 - deleting 14
 - description of 11
 - domain values 13
 - editing 14
 - temporary 13
 - used in rule flows 40
- theme, preference 6
- tile pane 3
-
- V**
- validating expressions 32
- vocabularies
- creating 12
 - deleting 14
 - description of 11
 - editing 14
 - tips for creating 12
- vocabulary pane 5

