Informatica® Cloud Data Quality
July 2024

Rule specification assets

Informatica Cloud Data Quality Rule specification assets
July 2024
July 2024

© Copyright Informatica LLC 1998, 2024

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, Informatica Cloud, Informatica Intelligent Cloud Services, and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at https://www.informatica.com/trademarks.html. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2024-07-11

Table of Contents

Preface 6

Informatica Resources 6

- Informatica Documentation 6
- Informatica Intelligent Cloud Services web site 6
- Informatica Intelligent Cloud Services Communities 6
- Informatica Intelligent Cloud Services Marketplace 6
- Informatica Knowledge Base 7
- Informatica Intelligent Cloud Services Trust Center 7
- Informatica Global Customer Support 7

Chapter 1: Introduction to rule specifications 8

Rule specification modes 8

Rule specification structure 8

- Configuration tab options in basic mode 9
- Configuration tab options in advanced mode 9
- Rule sets 10
- Inputs and outputs 11
- Rule statements 12

Rule specifications and exception management 13

Rule specifications and mappings 13

Rule specifications and dimensions 13

Chapter 2: Rule specification configuration 15

Prerequisites to rule specification configuration 15

- Verifying the business rule requirements 15
- Verifying the business data properties 16
- Determine the sequence of the steps in the rule 16

Configuring a rule specification in basic mode 16

Configuring a rule specification in advanced mode 17

- Configuring a new rule specification in advanced mode 18
- Converting a saved rule specification from basic mode 18

Creating an input in basic mode 19

Creating an input and output in advanced mode 19

Rules and guidelines for inputs 20

Rule specification data properties in basic mode 20

Rule specification functions 20

Dictionaries 23

Chapter 3: Rule set configuration 25

Rules and guidelines about rule sets 25

Rule set configuration examples . . . 26
Rule sets that use a single input. . . . 26
Rule sets with dependent conditions. . . . 26
Rule sets with independent conditions. . . . 27
Rule sets in parent and child relationships. . . . 27
Rule set properties . . . 28
Adding a rule set. . . . 28
Copying and moving rule sets . . . 29
Copying or moving a rule set within a rule specification. . . . 29
Copying or moving a rule set to another rule specification . . . 29

Chapter 4: Rule statement configuration . . . 31

Conditions. . . . 32
Rules and guidelines for mapplets in conditions and actions. . . . 33
Function configuration in conditions. . . . 33
Operators . . . 35
Actions. . . . 36
Function configuration in actions. . . . 38
Exception data properties. . . . 42
Function descriptions . . . 45
Rule statement menu options . . . 47
Configuring a rule statement. . . . 48
Configuring a rule statement that reads the result of another rule statement . . . 49
Configuring a rule statement with multiple conditions . . . 49
Adding a rule specification to a rule statement . . . 50
Cut and paste operations and rule statements . . . 51
Copying a rule statement within a rule set . . . 51
Copying a rule statement to another rule specification. . . . 51

Chapter 5: Rule specification configuration in advanced mode . . . 53

Rule specification logic in advanced mode. . . . 53
Rule logic configuration examples. . . . 54
Specifying a rule specification in an IF element. . . . 54
Specifying a rule specification in a THEN element. . . . 54
Specifying a mapplet in an IF element. . . . 55
Specifying a mapplet in a THEN element. . . . 56
Specifying a dictionary in the expression logic. . . . 56
Specifying exception data options in the expression logic. . . . 57
Rule specification functions . . . 58
Rule specification operators . . . 59

Chapter 6: Business rules and rule statements. . . . 60

Verifying the accuracy of business data . . . 61

Identifying errors in business data . . . 62
Standardizing project data values . . . 63
Improving the usability of business data . . . 64
Deriving information about business users . . . 65
Verifying that data values conform to business regulations . . . 66
Adding internal data to a data set . . . 67
Finding a value in a range of values . . . 68
Identifying data values that contain key terms or character strings . . . 68
Generating status data for a profile . . . 69
Using status data to identify exception records . . . 70
Resolving hidden data issues . . . 71
Finding null values . . . 72
Finding values that contain character spaces only . . . 72
Combining data values in advanced mode . . . 73
Standardizing character case in advanced mode . . . 74
Eliminating redundant character spaces in advanced mode . . . 75
Standardizing data values using a dictionary in advanced mode . . . 75
Writing the IF statement . . . 76
Writing the THEN statement . . . 76
Writing the ENDIF statement . . . 76
Adding optional parameters to the THEN statement function . . . 76
Rule logic example . . . 77
Standardizing custom data values in advanced mode . . . 77

Chapter 7: Validation and testing . . . 78

Rules and guidelines for test and validation operations . . . 78
Validating a rule specification . . . 78
Testing a rule specification in basic mode . . . 79
Testing a rule set . . . 80
Testing a rule specification in advanced mode . . . 80

Index . . . 81

Preface

Refer to *Rule specifications* to learn how to create and manage rule specifications in Informatica Intelligent Cloud Services ℠ Data Quality. A rule specification is an asset that represents the data requirements of a business rule in logical form. You create a rule specification in Data Quality and add the rule specification to a mapping in Data Integration.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

You can contact a Global Support Center through the Informatica Network or by telephone.

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to rule specifications

A rule specification is an asset that represents the data requirements of a business rule in logical form.

A rule specification breaks down the logic in a business rule into the following tasks:

- Define the types of data that a business data set contains.
- Define a set of conditions that the business data must satisfy.
- Define the actions to take when the data satisfies the conditions of the business rule.
- Define the actions to take when the data fails to satisfy the conditions of the business rule.

You create and test a rule specification in Data Quality. You add a rule specification to a Rule Specification transformation in a mapping in Data Integration. When the mapping runs, the Rule Specification transformation applies the conditions and actions that you define to the values in each row in the input data. The outputs from the transformation indicate whether the row data meets the requirements that you set in the business rule.

Rule specification modes

You can configure a rule specification in basic mode or advanced mode. By default, a rule specification displays the configuration options in basic mode.

Configure a rule specification in basic mode to define business rule logic using the set of graphical elements that the asset provides. Use basic mode if you are not familiar with text-based conditional rule logic. You might also use basic mode if the business rule that the rule specification will represent is easily transferrable into a graphical format.

Configure a rule specification in advanced mode to represent the business rule in expression logic in a text editor. You can use simple expressions, you can use IF-THEN-ELSE syntax to build the rule.

To switch from basic mode to advanced mode, click the **Convert to Advanced** option on the asset toolbar.

Note: You cannot switch to basic mode from advanced mode.

Rule specification structure

A rule specification asset contains options on a **Definition** tab and a **Configuration** tab. Use the **Definition** tab options to enter a name for the asset, optionally enter a description for the asset, and to select the folder in which to store the asset. Use the **Configuration** tab options to configure the rule specification logic.

Configuration tab options in basic mode

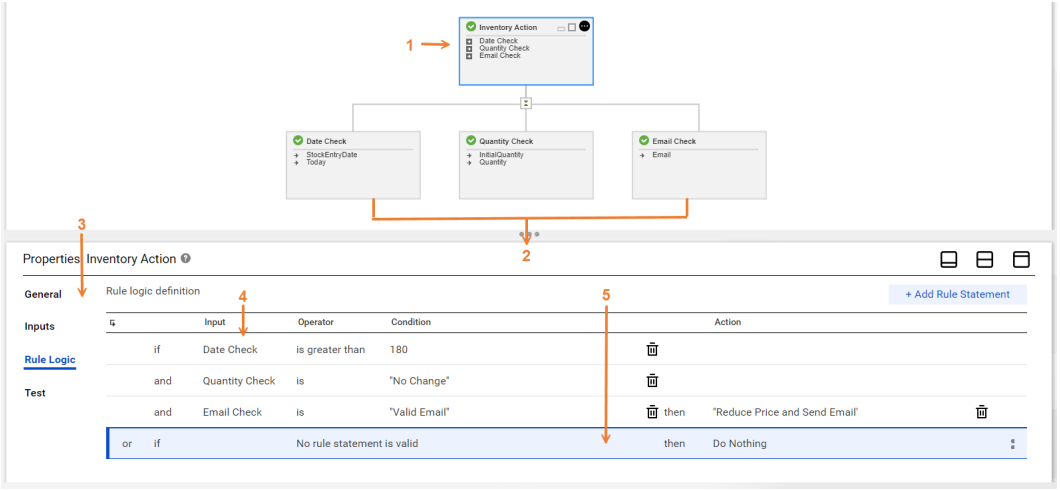
In basic mode, the **Configuration** tab contains the **Design** panel and the **Properties** panel. You can expand or collapse the **Design** panel to show or hide the rule sets that you add for configuration. You might collapse the **Design** panel when you configure a rule set with a rule statement, so that the rule set configuration options are easier to view.

When you create a rule specification, you configure a series of shapes in the **Configuration** tab. The shapes are called rule sets. Each rule set describes an aspect of the business rule.

A rule set contains one or more rule statements that define the business rule requirements at a low level. Each rule statement reads a column of input data and verifies that the input data meets the conditions that you specify.

The top-level rule set is the primary rule set. The primary rule set summarizes the business rule. The output from the primary rule set specifies whether each row of input data meets the overall requirement of the business rule.

The following image shows a rule specification in the **Configuration** tab in basic mode:



The rule specification contains the following elements:

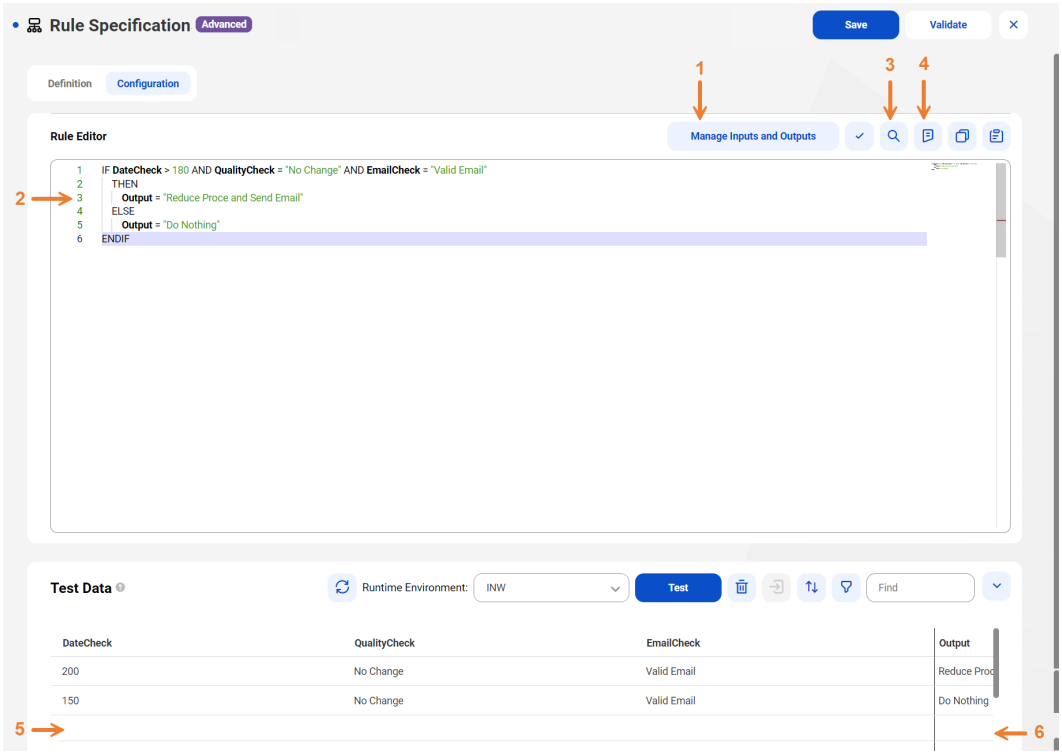
1. Primary rule set.
2. Child rule sets of the primary rule set.
3. Property options on the rule set that you select.
4. Rule statements in the rule set that you select.
5. Rule statement that the system defines.

Configuration tab options in advanced mode

In advanced mode, use the **Configuration** tab to write rule logic in simple expressions or in IF-THEN-ELSE syntax.

The **Configuration** tab in advanced mode contains an editor that allows you to directly write your business rule in expression logic.

The following image shows a rule specification in the **Configuration** tab in advanced mode:



The Configuration tab includes the following options:

1. Manage Inputs and Outputs.
   Opens the input and output configuration options. Use the options to add one or more inputs and outputs.
2. Rule Editor.
   Provides a canvas in which you can write the rule logic.
3. Search and replace.
   Searches and replaces a value that you specify.
4. Select all.
   Selects the entire script in the editor panel.
5. Test input field.
   Contains the input data that the Secure Agent uses to test the expression logic that you enter.
6. Test output field.
   Contains the result of the test.

Rule sets

Rule sets define the logical flow of data through the rule specification. You organize the rule logic in one or more rule sets when you configure a rule specification in basic mode. Data flows upward through the rule specification from the lowest rule set to the primary rule set. The number of rule sets in the rule specification depends on the logical requirements of the business rule. A valid rule specification might have a single rule set.

You can add a rule set below any rule set in the rule specification. The rule sets have a parent-to-child relationship. When you add a rule set, the output of the child rule set becomes an input to the parent rule set.

Note: You must select the output from any child rule set as an input in a rule statement in the parent rule set.

Rule sets contain the rule statements that analyze and update the input data. You can configure a rule set with a single rule statement, or you can add multiple rule statements to the rule set. Within a rule set, data flows from the first rule statement to the final rule statement.

You can copy or move a rule set to another location in a rule specification, and you can copy or move a rule set to another rule specification.

Inputs and outputs

An input describes a field of data that a rule statement can analyze in basic mode or that expression logic can analyze in advanced mode. An output represents the result of an operation within the rule specification. Inputs and outputs are specific to the rule specification in which you create them.

Define an input to represent a column in a data set that the rule specification will analyze at run time. Define an output to represent a column of data that the rule specification will generate at run time. In basic mode, a rule set generates an output. You define the outputs in advanced mode. The outputs that you define in advanced mode correspond to the results of a rule set in basic mode.

When you define an input or output, you can set the following properties:

- The input or output name.
- The data type of the data that the input or output represents. You can create an input with a string, date/time, float, or integer data type. You can additionally create an input of double data type in advanced mode.
- The maximum number of characters that a value in the input or output field can contain.

You can specify an integer data type for numbers in the range -2147483648 through 2147483647. To read numbers that are outside the integer range, use the float or double data type.

You can set the scale, or the number of digits that can follow a decimal point, in basic mode. In a float value, the default scale is 4. In a date/time value, the scale is preset to 9. You cannot change the scale in a date/time value.

The properties in basic mode also include a Usage value. The Usage value indicates the number of times that the input appears in a rule statement in the rule specification.

To view the inputs that a rule set uses in basic mode, select the rule set on the Configuration tab.

Rules and guidelines for inputs

Consider the following rules and guidelines for rule specification inputs:

- Create an input in basic mode to add the input to a rule statement.
- When you create a child rule set, the output from the rule set becomes an input to the parent rule set. You must use the input in a rule statement in the parent rule set.
- An input in a rule specification asset does not store information about business data, such as the name of a source column, table, or file.
- When you add a rule specification asset to a Rule Specification transformation in a mapping, you can connect a rule specification input to any column of data that is compatible with the input properties.

Input data types and Amazon S3 connections

You might use a Rule Specification transformation in a mapping that connects to a file source over an Amazon S3 V2 connection. For example, you might run such a mapping in Data Integration Elastic.

Consider the following factors before you configure the rule specification asset that the Rule Specification transformation will use:

- A mapping that uses an Amazon S3 V2 connector cannot process a date/time input on a rule specification.
- A mapping that reads a source file over an Amazon S3 V2 connector can read a string data type without additional configuration, as the string data type corresponds to the Amazon S3 STRING data type.

For more information on connectors, see the documentation for Connectors in the Data Integration online help.

Rule statements

In basic mode, a rule statement is a set of operators, conditions, and actions that analyze a column of data and generate an output based on the result of the analysis. You add a rule statement to a rule set.

A condition is a data operation that determines a single fact about a data value. You can add multiple conditions to a rule statement. An action is a data operation that generates a potential output from the rule set. An action generates data when the input that you add to the rule statement satisfies the conditions that you define.

A rule specification reads the rule statements in a rule set from top to bottom. For a given row of input data, the rule set accepts the output from the first rule statement that generates output data.

Each rule set contains a system-defined rule statement that specifies the action to take if no other rule statement generates output data. The system-defined rule statement is the final rule statement in the rule set. You can edit the action in the system-defined rule statement. By default, the rule statement specifies that the rule set does not generate any output data if the other rule statements do not generate output data.

In advanced mode, you can write the equivalent rule in expression logic.

Status values in rule statements

The actions that a rule statement can perform include the generation of status values. A status value is a predefined value that a rule statement can generate as the output from an action. You can configure an action to generate a status value as an alternative to user-defined value.

A rule statement can return *Valid* or *Invalid* as status values. You cannot modify the status values.

You can use status values to achieve the following objectives:

Provide data to profiles

Scorecards in Data Profiling can recognize status values. When a rule specification returns a status value as an output, the scorecard can report the number of status values as a data quality category. To enable the scorecard to read the rule specification output, add the rule specification as a rule to the profile that generates the scorecard.

Provide information to downstream users about exception records

You can configure a rule specification to identify a record as an exception. An exception is a record that contains unresolved data quality issues.

To identify records as exceptions in basic mode, configure a rule statement to return the status value *Invalid*. Define the exception properties on a rule statement in the primary rule set.

To identify records as exceptions in advanced mode, configure exception indicators for one or more status values that you define in the rule logic that you write. In advanced mode, you can define custom status values that more closely describe the data quality issue in each exception record. You can associate each status value with a corresponding data quality issue. You can configure a different set of exception properties for each status value.

Rule specifications and exception management

An exception is a record that contains unresolved data quality issues. You can configure a rule specification to find exception records as part of an exception management process. In an exception management process, you add a rule specification that you configured for exception management to a profiling task. You then create an exception task from the profiling task.

A rule specification can add data to each exception record that describes the type of data quality issue that the record represents. The exception data properties include an output field for status values that can summarize the data quality issue that the rule logic identifies in a record. The properties also include an output for values that indicate the priority of each issue and an output to contain a description of each issue. The output values allow a downstream user to find exception records and to understand the reasons why a record is an exception.

For more information about the exception data properties, see "Exception data properties" on page 42.

Rule specifications and mappings

The rule specifications that you create are available to the users who create mappings in Data Integration. You or another user add a Rule Specification transformation to a mapping and then add a rule specification asset to the transformation. The Rule Specification transformation applies the logic in the asset to the input fields that you specify.

After the mapping runs, you can evaluate the data output to determine if the data conforms to the business rule that the rule specification describes.

Rules and guidelines for rule specifications and mappings

Consider the following rules and guidelines when you work with mappings and rule specifications:

- A rule specification is a reusable object. A Data Integration user can add a rule specification to transformations in multiple mappings. A Rule Specification transformation is a nonreusable object. You select a Rule Specification transformation when you configure a mapping.
- A rule specification is a read-only object in Data Integration. The rule specification does not change unless you update it in Data Quality.
- To view the list of rule specifications that you created in Data Quality, select the **Explore** option.
- If you rename a rule specification that you previously added to a transformation, you must synchronize the transformation with the rule specification.

  To synchronize the transformation, open the transformation in the mapping and select the rule settings. Data Integration displays a message that prompts you to synchronize the transformation with the rule specification.

Rule specifications and dimensions

The data quality issues that you may find in your data can fall into a range of common categories. Data Quality assets can identify the categories as dimensions. When you configure an asset in Data Quality, you can use the **Dimension** property to indicate the type of data quality issue that you want the asset to examine.

Find the Dimension property on the **Definition** tab of the asset.

For more information about data quality dimensions, see the *Introduction* in Data Quality documentation.

Dimensions and scorecards

A rule specification that specifies a dimension is a key element in a scorecard. If you will use a rule specification in a scorecard, set a dimension on the scorecard.

A scorecard is a graphical representation of the values across the columns in a profile. You can use scorecards to monitor changes in profile data over time. You create scorecards in Data Profiling.

CHAPTER 2

Rule specification configuration

When you configure a rule specification, you define a series of inputs, rule sets, and rule statements. Each element in the rule specification represents a data operation that interacts with the business data to validate a business rule.

Before you configure the rule specification, verify that you understand the properties of the business data and the requirements of the business rule.

Prerequisites to rule specification configuration

When you configure a rule specification, you translate the requirements of a business rule into one or more rule statements. The rule statements represent the logic that determines whether a data set conforms to the business rule.

Before you configure the rule specification, perform the following steps:

1. Verify the business rule requirements.
2. Verify the business data properties.
3. Determine the sequence of the steps in the rule.

Note: When you verify the business rule requirements, you may determine that the business rule addresses a key dimension of data quality in your data. You can optionally select the data quality dimension that your business rule represents when you configure the rule specification. For more information about data quality dimensions, see "Rule specifications and dimensions" on page 13.

Verifying the business rule requirements

Before you configure a rule specification, discuss the business rule with the data owners in the organization. Verify that the business rule is valid and ready to apply to the business data.

1. Identify the business rule that the rule specification represents.
2. Identify the business data set that the business rule validates.
3. List the business rule requirements that apply to the data inputs.

   The business rule requirements indicate the types of rule logic that you define in the rule specification.
4. Identify the information types that the business rule applies to.

   The information types indicate the data types of the inputs that you create in the rule specification.

Verifying the business data properties

Before you create a rule specification, identify the business data that the rule specification can apply to. Work with a developer or data steward to identify the data source.

1. Identify one or more data sets that you or another Data Integration user can select in a mapping that reads the rule specification. For example, identify the database and the table that contain the mapping source data.
2. Verify the data types of the data columns that the rule specification will analyze. You specify the data types when you create the inputs in the rule specification.

   Note: You might not create an input for every column in the data set.

Determine the sequence of the steps in the rule

At a high level, a business rule defines a single objective that the business data must satisfy. To represent the objective in the rule specification, you'll define at least one rule statement in basic mode or you'll write at least one expression in advanced mode. If the business rule defines more than one data outcome, you might define more complex rule logic.

In basic mode, you might add rule sets and rule statements below the primary rule set to validate the data that the primary rule set analyzes. In advanced mode, you might nest multiple IF-THEN-ELSE conditions within the primary condition that you define.

1. Create the inputs that the rule statements analyze. The inputs are a key prerequisite for the rule statements. The inputs represent the columns in the business data set that the business rule applies to.
2. Identify the business rule conditions that the business data must satisfy. Plan a rule statement or an IF-THEN-ELSE condition for each condition.
3. Determine the sequence in which the rules must analyze the data.

   Start from the lowest dependencies in the business rule, and plan a rule set or IF-THEN-ELSE condition for each dependency.

Configuring a rule specification in basic mode

In basic mode, you configure inputs, rule sets, and rule specifications to represent the data that the business rule will read and the steps that the business rule must perform. Basic mode is the default mode in the asset.

Before you create a rule specification, read the "Prerequisites to rule specification configuration" on page 15.

1. Click **New** > **Rule Specification**.

   Data Quality opens the **Rule Specification** page.
2. On the Definition tab, enter a name for the rule specification.
3. Optionally, enter a description.

   Tip: Enter a summary of the underlying business rule as the description.
4. Select the location in which to save the rule specification.
5. Optionally, select a data quality dimension to represent the type of data quality issue that you want the asset to examine.

   For more information about dimensions, see "Rule specifications and dimensions" on page 13.

6. Save the rule specification.

   The **Asset References** panel appears. The **Asset References** panel shows additional assets that are referenced by Data Quality assets to perform transformation tasks. When you create a new asset, there are no referenced assets.

7. Optionally, add a tag to the rule specification. You can search for assets with a common tag on the Explore page.
8. Select the **Configuration** tab.

   Data Quality displays the configuration workspace for the rule specification. The **Configuration** tab contains the **Design** panel and the **Properties** panel.

9. Expand the **Design** panel, and configure a rule statement in the primary rule set.

   The primary rule set defines the data output from the rule specification.

   The primary rule set contains a preconfigured rule statement and reads a preconfigured input by default. Based on your business rule requirements, you can modify the default components.

10. Optionally, update the primary rule set name.
    - Select the primary rule set, and select **General**.
    - Update the rule set name.
11. Add any rule set that the rule specification requires.

    You add a rule set below another rule set.

12. Add one or more inputs to each rule set.

    Use the **Input Management** dialog box to create the inputs.

    You can modify the properties on the default input.

13. Add any rule statement that the rule specification requires.

    You can add a rule statement to the primary rule set or to another rule set. For more information on rule statements, see Chapter 4, “Rule statement configuration ” on page 31.

14. Save the rule specification.

After you configure the rule specification, test the rule specification with sample data.

Configuring a rule specification in advanced mode

In advanced mode, you define the inputs and outputs for the business rule, and you write expression logic that represents the steps that the business rule must perform.

To configure a rule specification in advanced mode, you convert the asset from basic mode. The steps you follow depend on whether you save the rule specification in basic mode before you convert to advanced mode.

If you saved the asset in basic mode, Data Quality converts your asset configuration to advanced mode. Data Quality creates a new asset with the advanced configuration and maintains the original rule specification in basic mode.

If you did not save the asset in basic mode, Data Quality converts the asset to advanced mode in its default configuration.

Before you create a rule specification, read the “Prerequisites to rule specification configuration” on page 15.

Configuring a new rule specification in advanced mode

In this task, you'll create a rule specification from scratch and save it in advanced mode.

1. Create a rule specification asset.
2. On the asset toolbar, click **Convert to Advanced**.
3. On the **Definition** tab, enter a name for the rule specification.
4. Optionally, enter a description.

   Tip: Enter a summary of the underlying business rule as the description.
5. Select the location in which to save the rule specification.
6. Optionally, select a data quality dimension to represent the type of data quality issue that you want the asset to examine.

   For more information about dimensions, see "Rule specifications and dimensions" on page 13.
7. Save the rule specification.

   The **Asset References** panel appears. The **Asset References** panel shows additional assets that are referenced by Data Quality assets to perform transformation tasks. When you create a new asset, there are no referenced assets.
8. Select the **Configuration** tab.

   Data Quality displays the configuration workspace for the rule specification.
9. Click **Manage Inputs and Outputs**.

   The **Input and Output Management** dialog box appears.

   The asset reads a preconfigured input and output by default. You can modify the properties on the default input and output and add any other input or output that the rule specification requires.
10. In the **Rule Editor** panel, write the rule logic. Build the rule logic in an IF... THEN... ENDIF structure.

    The asset adds a preconfigured rule to the editor when you begin to type. You can modify or delete the default rule.
11. Save the rule specification.

After you configure the rule specification, test the rule specification with sample data.

Converting a saved rule specification from basic mode

In this task, you'll convert a rule specification that you saved in basic mode.

1. Open a rule specification that you configured in basic mode.
2. On the asset toolbar, click **Convert to Advanced**.

   Data Quality creates a copy of the rule specification in advanced mode.
3. Save the rule specification in advanced mode with a new name.

   Review the description, location, and other properties on the **Definition** tab.
4. Verify or update the configuration of the asset to ensure that the inputs, outputs, and expression logic meet your requirements.
5. Save the rule specification.

Creating an input in basic mode

When you create an input, you can add the input to any rule set in the rule specification.

1. On the Explore page, open the rule specification.
2. In the toolbar, select **Manage Inputs**.

   The **Input Management** dialog box opens. The dialog box lists the current inputs in the rule specification.
3. Click **Add Input**.
4. Set the following properties on the input:
   - The input name. You can enter alphanumeric characters, underscore characters, and spaces.
   - The data type of the column in the business data set that the input represents. You can select a date/time, float, integer, or string data type.
   - The maximum character length of the data values in the input data column.
   - The number of digits that can follow a decimal point in a float value.
   - A text description of the input. The description is optional.
5. To save the input and close the dialog box, click **OK**.

Note: The **Input Management** dialog box includes a Usage column. The Usage column indicates the number of times that each input appears in a rule statement in the rule specification.

Creating an input and output in advanced mode

When you create an input and output, you can add the input and output to the rule logic that you write to an editor in the rule specification.

1. On the **Explore** page, open the rule specification.
2. In the toolbar, select **Manage Inputs and Outputs**.

   The **Input and Output Management** dialog box opens. The dialog box lists the current inputs and outputs in the rule specification.
3. Select the **Inputs** or **Outputs** tab.
   - To create inputs to use in the rule logic, select the **Inputs** tab.
   - To create outputs to use in the rule logic, select the **Outputs** tab.
4. Click **Add Field**.
5. Set the following properties on the field:
   - The input or output name. You can enter alphanumeric characters and underscore characters.
   - The data type of the column in the business data set that the input represents. You can select a date/time, float, integer, double or string data type.
   - The maximum character length of the data values in the input data column.
   - The number of digits that can follow a decimal point in a float value.
6. Click **OK**.

Rules and guidelines for inputs

Consider the following rules and guidelines when you create an input:

- Where possible, use input names that match to the names of the input fields that the rule specification will read in a mapping. When you add the rule specification to a Rule Specification transformation in a mapping, you can use an Automatch option to connect the rule specification inputs to transformation input fields with similar names. You can also manually connect any rule specification input to any input field on the transformation.
- When you perform a computation that involves a string data type and an integer data type in advanced mode, the asset can cast the digits in the string as an integer. If the string contains non-numeric characters in the computation with an integer, the asset assigns a value of zero to the non-numeric characters in the string. For example, if you specify a string input value of ABC and an integer input value of 10 in the expression `Output = String_Value + Integer_Value`, the asset generates an output value of 10.

  If you specify a string input value of 10 and an integer input value of 10 in the expression `Output = String_Value + Integer_Value`, the asset generates an output value of 20.

Rule specification data properties in basic mode

The rule specification properties define the data characteristics for the outputs on each rule set in the rule specification, including the primary rule set.

To view or edit the properties, select **Rule Properties** from the Actions menu on the toolbar.

You can define the following properties:

- Maximum string length. Determines the maximum length for text data. Default is 100.
- Maximum number length. Determines the maximum length for numeric data. Default is 10.
- Number of decimal places. Determines the maximum number of decimal places to include in the output data. Default is 4.

Rule specification functions

The rule specification asset includes a range of functions that you can use in rule logic in basic and advanced modes.

You can use the following functions:

- ABS
- ADD_TO_DATE
- ASCII
- CEIL
- CHOOSE
- CHR
- CHRCODE

- CONCAT
- CONTAINS
- CONVERT_BASE
- COS
- COSH
- CRC32
- CUME
- CURDATE
- DATE_COMPARE
- DATE_DIFF
- DATECONVERT
- DECODE
- EXP
- FLOOR
- FV
- GET_DATE_PART
- GREATEST
- IFF
- IN
- INDEXOF
- INITCAP
- INSTR
- IS_DATE
- IS_NUMBER
- IS_SPACES
- ISNULL
- LAST_DAY
- LEAST
- LEFTSTR
- LENGTH
- LN
- LOG
- LOWER
- LPAD
- LTRIM
- MAKE_DATE_TIME
- MAX
- MD5
- METAPHONE

- MIN
- MOD
- MONTHCOMPARE
- MOVINGAVG
- MOVINGSUM
- NPER
- PMT
- POWER
- PV
- RAND
- RATE
- REG_EXTRACT
- REG_MATCH
- REG_REPLACE
- REPLACECHR
- REPLACESTR
- REVERSE
- RIGHTSTR
- ROUND
- RPAD
- RTRIM
- SET_DATE_PART
- SIGN
- SIN
- SINH
- SOUNDEX
- SQRT
- SUBSTR
- SYSTIMESTAMP
- TAN
- TANH
- TIMECOMPARE
- TO_CHAR
- TO_DATE
- TO_DECIMAL
- TO_FLOAT
- TO_INTEGER
- TRUNC
- UPPER
- XOR

Dictionaries

A dictionary is a data set that contains one or more columns of reference data. You can specify a dictionary in a rule statement condition or action in basic mode or in expression logic in advanced mode. For example, add a dictionary to a rule statement condition or an IF statement to verify that the input data contains the data values that you expect. Add a dictionary to rule statement action or a THEN statement to return a value that corresponds to the input data value.

A dictionary might contain public terms, such as telephone area codes or address abbreviations. Or, a dictionary might contain values that are specific to an organization, such as employee codes or product codes.

At least one dictionary column must contain a set of standard or required values as defined by a business rule. The other columns can contain alternative versions of the values. The column that contains the correct or required values is called the valid column. By default, the first or left-most column in a dictionary is the valid column. You can populate a dictionary with any combination of values that suits your business rule logic. The data in each column does not need to be formally correct.

When you add a dictionary to a condition or IF statement, the rule specification compares the input data to the values in each column in the dictionary and passes the results to the action in the rule statement.

When you add a dictionary to an action or THEN statement, the rule specification compares the input data to the values in each column in the dictionary and returns a value according to the operation that you define. For example, the action might return the value in the valid column on the same row in the dictionary.

Note: In a search and replace operation, a rule specification searches every column in the dictionary except the valid column that the asset specifies. If you want the search operation to include the valid column data, add a copy of the valid column to the dictionary. By default, the valid column is the first or left-most column in a dictionary. You can create a dictionary that contains two identical columns of data.

You can change the valid column that an action uses in the **Define Actions** dialog box in basic mode. To change the valid column that you use in advanced mode, add the column name when you identify the dictionary in the rule logic.

You can use any dictionary in a condition regardless of the number of columns that the dictionary contains. Some of the dictionary options in the **Define Actions** dialog box require a dictionary with two or more columns.

Dictionary example

The following table contains rows from a dictionary of city names in India:

Name1	Name2
Mumbai	Bombay
Chennai	Madras
Bengaluru	Bangalore
Puducherry	Pondicherry
Kadapa	Cuddapah
Kochi	Cochin
Kolkata	Calcutta

The Name1 column contains the current names of cities in India. The Name2 column contains the previous version of each city name. You might use the dictionary in an action to replace the older city names with the current city names.

You might also create a dictionary with a single column of data that contains the older city names. You can use the dictionary in a condition to determine if an input data set contains the older names.

CHAPTER 3

Rule set configuration

You configure one or more rule sets on a rule specification in basic mode. The number of rule sets that you configure depends on the facts that you need to verify about the business data.

A fact is a piece of information that you derive from a data value when you measure it or compare it to other data values. For example, a date value might indicate a birthday, or it might indicate the date that you sent an invoice to a customer. You might use a date to determine if a person is an adult or if a customer account is overdue. You can use the facts that you determine about the person or about the customer to make business decisions. A rule specification can determine a single fact, or it can determine multiple facts that contribute to an overall result for a row of business data.

When one fact about the business data depends on multiple other facts, create rule sets in a parent and child relationship. Use the child rule sets to generate inputs that the parent rule set can verify. If a fact relates to a single item of data, you can configure rule statements to analyze the data in a single rule set. If a fact depends on multiple independent facts about the business data, create child rule sets in parallel to determine each fact.

Rules and guidelines about rule sets

When you configure rule sets, you define the flow of data through the rule specification. The position of the rule sets and the order of the rule statements in the rule sets determine the data flow.

Consider the following rules and guidelines when you configure rule sets:

- Data flows upwards through a rule specification from the lowest rule set to the primary rule set. When the rule specification includes parallel rule sets, the rule specification runs the parallel rule sets concurrently.
- A rule set generates a single output for a row of input data. The output from the first rule statement to generate an action becomes the rule set output for the row.
- The output from the primary rule set is the primary fact that the rule specification generates. The primary rule set output represents the result of the business rule analysis of the data row.
- The output from a child rule set is an input to a parent rule set. Every rule set that you add to a rule specification is a child of the primary rule set or a child of another rule set.

  You must add the output from a child rule set to a rule statement in the parent rule set.
- The input properties on a rule set display the inputs that you select in the rule set and the inputs from any child rule set. To view all of the inputs in the rule specification, open the **Input Management** dialog box.
- The rule statements in a rule set must generate outputs of the same data type.

Rule set configuration examples

The number of rule sets that you configure in a rule specification depends on the facts that you need to determine about the input data.

You can configure rule sets in the following ways:

- Configure rule sets that read a single input.
- Configure rule sets that use rule statement conditions in a dependent relationship.
- Configure rule sets that use rule statements in an independent relationship.
- Configure a parent rule set and child rule sets.

Rule sets that use a single input

Configure a rule set to read a single data input when you want to analyze a range of data values in an input column. Define rule statements that apply the same condition logic to the input, and define a different action for the outcome of each condition.

Single input example

An airline sells airplane tickets online. The airline sells tickets at different prices for travelers in different age groups. The airline defines a business rule that requires the customer database to identify the age group of each traveler. You configure an input that represents the date-of-birth data in a customer data set. You create a series of rule statements that compare the data values in the input data to the current date.

You configure a rule set with the following rule statements:

```
    IF AGE >= 65 THEN SENIOR
OR IF AGE >= 18 THEN ADULT
OR IF AGE < 18 THEN STUDENT
```

The rule set returns data from the first rule statement that can generate an action. Therefore, the order of the rule statements is relevant. If the rule set reads the second rule statement first, the rule set cannot identify any traveler in the senior age category. When the rule set reads the second rule statement first, the rule set identifies all traveler who are 18 years or older as adults.

Rule sets with dependent conditions

Configure a rule set with rule statement conditions in a dependent relationship when the business rule requires multiple conditions to be true. Define a rule statement that applies a set of related conditions to the inputs, and define a single action for the conditions.

Use an AND operator in the rule statement to join the conditions.

Dependent conditions example

A city bus company wants to hire bus drivers. The bus company requires all drivers to pass an eyesight test and to hold a license to drive a bus. The company defines a business rule that requires the employee database to verify the eyesight test status and the license status for each driver. You create inputs that represent the data for the eyesight test and the license. You configure a rule statement with a condition for each input. You configure the default rule set to return a string when an input to the earlier rule statement is not valid.

You configure a rule set with the following rule statements:

```
    IF EYESIGHT TEST IS YES
    AND DRIVERS_LICENSE IS YES THEN OK
OR IF NO RULE STATEMENT IS VALID THEN NOT_OK
```

Rule sets with independent conditions

Configure a rule set with rule statements in an independent relationship when the business rule requires one of multiple conditions to be true.

Independent conditions example

A bank wants to offer credit cards to customers. The bank decides that any customer with a credit balance greater than $5,000 or a salary greater than $50,000 qualifies for a credit card. The bank defines a business rule that requires the customer database to identify the eligible customers. You configure inputs that represent the customer salary and the customer balance. You configure a rule statement to test each input.

You configure a rule set with the following rule statements:

```
   IF CUSTOMER SALARY >= 50000 THEN OFFER_CARD
OR IF CUSTOMER BALANCE >= 5000 THEN OFFER_CARD
```

The rule set returns data from the first rule statement that can generate an action. However, the order of the rule statements is not relevant in the example. If the data on any input is valid, the input row is valid.

Rule sets in parent and child relationships

When a business rule describes a data requirement that depends on other data requirements, configure rule sets in a parent-and-child relationship.

Parent and child example

A hardware store sells wood preserver in multiple colors and styles. The store must verify that the product database tracks the inventory of wood preserver in the warehouse. The store defines a business rule that requires the product database to maintain accurate data for the colors, styles, and units of wood preserver. You configure child rule sets with rule statements that verify the data for each type of wood preserver. You configure the parent rule set to verify the output from the child rule sets.

You configure the following child rule sets:

Rule set name: Color

```
IF COLOR = (PINE, OAK, CHESTNUT) THEN GOOD_COLOR
```

Rule set name: Size

```
IF SIZE = (250, 500, 1000) THEN GOOD_SIZE
```

Rule set name: Style

```
IF STYLE = (GLOSS, MATT) THEN GOOD_STYLE
```

You configure the following parent rule set:

Rule set name: Wood Preserver

```
IF COLOR = GOOD_COLOR
AND SIZE = GOOD_SIZE
AND STYLE = GOOD_STYLE THEN WOOD PRESERVER IS GOOD_PRODUCT
```

To configure a condition to compare an input to a range of values, select the *Input: List of Values* option in a function condition. You can configure similar parent rule sets and child rule sets for other products in the inventory.

Note: You might also create dictionaries that contain lists of the acceptable colors and styles. Configure the conditions to compare the input values to the dictionaries.

Rule set properties

To view the properties on a rule set, select a rule set in the Configuration tab. Use the properties to describe the rule set, to add inputs and rule statements, and to test the rule set.

The rule set properties contain the following views:

General

Displays the rule set name and any description that you add to the rule set. You can update the name and the description.

Enter a name that summarizes the type of data operations that the rule statements perform in the rule set. The rule set name must be unique in the rule specification.

Inputs

Lists the inputs that the current rule set uses. If a rule set is a child to a parent rule set, the parent rule set reads the child rule set name as an input name.

To view all of the inputs in the rule specification or to add an input to the rule specification, open the **Input Management** dialog box.

Rule Logic

Displays the rule statements in the rule set. You can add and update a rule statement.

Test

Displays the rule set inputs in a table that you can use to test the rule set logic. The table includes the inputs from any child rule set below the current rule set.

You can enter sample data values and verify that the rule set generates the results that you expect. To test the complete rule specification, select the primary rule set.

Adding a rule set

By default, a rule specification contains a primary rule set. Add a rule set under the primary rule set or under another rule set in the rule specification.

1. On the Explore page, open the rule specification.
2. On the Configuration tab, select a rule set as a parent for the rule set that you create.
3. Click the *Add rule set* button [+] at the base of the rule set.

   Or, right-click the rule set and select *Add rule set*.

   Data Quality adds a rule set below the parent rule set.

Copying and moving rule sets

Copy a rule set to reuse the rule statements in the rule set. Move a rule set when you determine that the rule set belongs in another location.

You might copy a rule set to another rule specification to reuse the rule statements that the rule set contains. You might copy a rule set within a rule specification to create rule statements that are similar to the rule statements in the rule set.

You might move a rule set when you determine that the rule set is a dependency for a different business rule requirement.

You copy or move a rule set to a position below another rule set. The rule set that you copy or move becomes a child rule set of the destination rule set. When you copy or move a rule set to another rule specification, you also copy the inputs that the rule set uses. You cannot move a primary rule set.

The copy and cut options are disabled on a primary rule set when you configure exception management on the rule set.

Copying or moving a rule set within a rule specification

To create multiple similar rule sets in a rule specification, copy the rule set. Move a rule set when you determine that the rule set belongs in another location.

After you copy or move a rule set, update the rule statements to reflect the business requirements that the rule set represents.

1. Open the rule specification.
2. Select a rule set, and right-click it to open the menu.
   - To copy the rule set, select **Copy**.
   - To move the rule set, select **Cut**.
3. Select a parent rule set for the rule set that you copy or move. Right-click the parent rule set to open the menu.
4. Click **Paste**.
5. The rule set that you copy or move appears below the rule set that you select.

   Data Quality adds the string _COPY to the rule set name.
6. Save the rule specification.

Copying or moving a rule set to another rule specification

If you create multiple rule specifications with similar logic, you can copy the rule sets from one rule specification to another to save time. Move a rule set when you determine that the rule set belongs in another location.

After you copy or move a rule set, you might update it to reflect the business requirements of the current rule specification. Or, you might apply the rule set to different inputs.

1. Open the rule specification.
2. Select a rule set, and right-click it to open the menu.
   - To copy the rule set, select **Copy**.
   - To move the rule set, select **Cut**.
3. Click **Explore**.

4. Select a destination rule specification for the rule set.

   The rule specification opens.
5. Select a parent rule set for the rule set that you copy or move. Right-click the parent rule set to open the menu.
6. Click **Paste**.
7. The rule set that you copy or move appears below the rule set that you select.
8. Save both rule specifications.

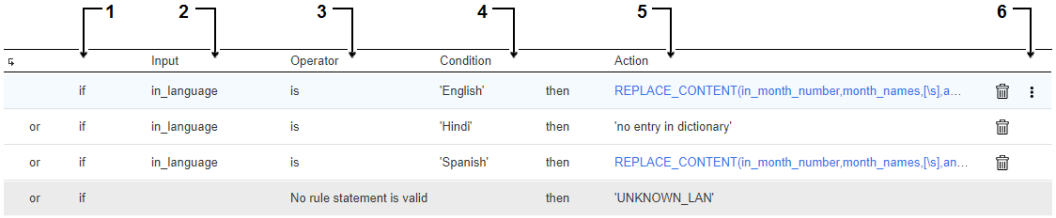
CHAPTER 4

Rule statement configuration

Rule statements define the data analysis and data transformation operations that a rule specification applies to input data in basic mode. Rule statements contain inputs, operators, conditions, and actions.

A condition defines the type of analysis that the rule statement performs on the input data. An operator is a mathematical function that determines how the rule statement processes the result of the condition analysis. An action defines the output that the rule statement generates if the input data satisfies the condition.

The following image shows a series of rule statements on the Configuration tab:



The rule statements have the following elements:

1. Condition type.
   Indicates the condition or conditions in which the input data can be valid. When you create a rule statement, you create an IF condition. When you add a condition to a rule statement, you create an AND relationship between the conditions in the statement.
2. Input.
   Identifies the data to analyze or update. You can select an input from the current rule set, or you can select the output from a child rule set.
3. Operator.
   Identifies the type of validation operation that the condition applies to the input data.
4. Condition.
   Describes a fact that the rule statement verifies about the input data.
5. Action.
   Describes the output that the rule statement generates when the input data satisfies the condition. If you configure multiple conditions in the rule statement, the input data must satisfy all conditions to generate an action.

   If a function in a condition or an action reads a dictionary, the condition or action description includes the dictionary name. Click the dictionary name to open the dictionary.
6. Menu options.
   Displays a list of options for the rule statement.

An input, operator, and condition correspond to an IF statement in the logic that you define in advanced mode. An action corresponds to a THEN statement in the logic that you define in advanced mode. The final rule statement in a rule set describes the action to take if no previous rule statement can generate an action. The final rule statement corresponds to an ELSE statement in the logic that you define in advanced mode.

Conditions

A condition is a data operation that a rule statement specifies for one or more input data values. Every rule statement contains at least one condition.

When a condition validates an input data value, the rule specification performs the action in the rule statement. The action represents a potential output from the rule set. If a condition cannot validate an input data value, the rule specification does not perform the action.

Configure a condition to perform one of the following types of analysis:

Compare the input data values to a single value

The rule statement compares the values in the input column to the value that you enter.

Compare the input data values to the values on another input

The rule statement compares the values in the input column to the values on the same row in another input column that you specify.

Compare the input data values to the current date and time

The rule statement compares the values in the input column to the current date and time.

Search for null values

The rule statement searches the input column for null or empty strings.

Compare the input data values to a range of values

The rule statement compares the values in the input column to a range of values that you enter.

Compare the input data values to the values in a dictionary

The rule statement compares the values in the input column to the values in a dictionary. The rule statement returns a value from the dictionary or a value that you enter.

Compare the input data values to the output from another rule specification

The rule statement compares the values in the input column to the output from a rule specification that you select. The rule specification runs within the rule statement logic.

Specify at least one input for the rule specification. You can select any input available to the current rule specification, including the current condition input, or you can enter a value as a rule specification input. You also select the field that the condition reads as the rule specification output.

Compare the input data values to the output from a mapplet

The rule statement compares the values in the input column to the output from a mapplet that you select. The mapplet runs within the rule statement logic.

Specify at least one input for the mapplet. You can select any input available to the current rule specification, including the current condition input, or you can enter a value as a mapplet input. You also select the field that the condition reads as the mapplet output.

Note: A rule specification can read mapplets that contain passive transformations.

Apply a function expression to the input data values

The rule statement applies a function expression that you select to the values in one or more input columns.

Apply a custom expression to the input data values

The rule statement applies a custom expression that you define to the values in one or more input columns.

Rules and guidelines for mapplets in conditions and actions

You can optionally add a mapplet to a function in a rule statement condition or action. When you add a mapplet, you add the logic of any asset or transformation in the mapplet to the condition or action.

The mapplet can contain any passive data quality transformation and can additionally contain a passive Java transformation and Expression transformation.

Consider the following rules and guidelines when you add a mapplet to a condition or action:

Passive transformations

You can use the following transformations in a mapplet that you add to a condition or action:

- Parse transformation
- Cleanse transformation
- Labeler transformation
- Rule specification transformation
- Verifier transformation
- Expression transformation
- Java transformation in passive mode

Note: Do not define an in-out parameter on a mapplet that you add to a condition or action. Before you add a mapplet with a Java transformation to a condition or action, compile the transformation in Data Integration. For information about how to use a Java transformation that consumes external Java packages in a mapplet, see the Knowledge Base article KB 000227229.

Nested mapplets

You can configure a condition or action with up to four nested mapplets and rule specification assets in an alternating manner.

Consider the following example:

- You select the Mapplet function in a condition.
- You configure the condition to read a mapplet that contains a Rule Specification transformation named `RuleSpec1`.
- The Rule Specification transformation reads a rule specification asset with a condition that also specifies a Mapplet function.
- The mapplet that you specify in this condition contains a Rule Specification transformation named `RuleSpec2`, which reads an additional rule specification asset.

In this way, you can create four levels of mapplets and rule specifications beneath a condition or action.

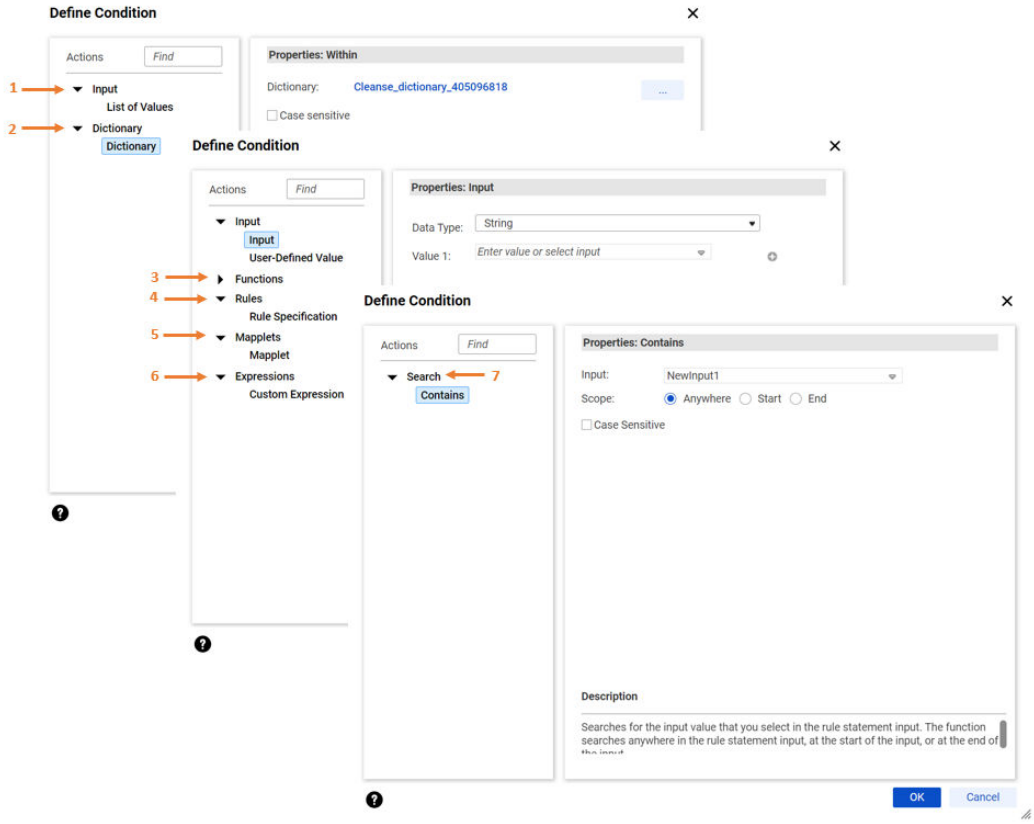
Function configuration in conditions

You can configure the data operation in a condition as a function expression. You can also select a function expression from a list of expressions in the rule specification. Use the function options in the condition to configure or select the expression.

You can use the function options as an alternative to the inline condition options that the rule statement displays in the workspace. You can configure a function to read multiple inputs. You can also use the function options to add a dictionary or another rule specification to the condition.

Use the **Define Condition** dialog box to configure or select a function expression. To open the dialog box, select *function* as the condition type in the rule statement. The function expressions that you can use depend on the operator that you select.

The following image shows the options on the **Define Condition** dialog box for different operators:



The dialog box contains the following options:

1. Input
   Use the Input options to compare an input data value to other data values in the following ways:
   - Compare the rule statement input with an input from another column on the same row.
   - Compare the rule statement input with a value that you enter.
   - Compare the rule statement input with a range of values that you enter.

   Select the Inputs option with the following operators: *is, is not, is greater than, is greater than or equal to, is less than, is less than or equal to, is within,* and *is not within*.
2. Dictionary
   Use the Dictionary options to compare an input data value with the values in a dictionary.

   Select the Dictionary option with the following operators: *is within* and *is not within*.
3. Functions
   Use the Functions options to apply an expression to the input data. Select an expression from the list in the dialog box.

   Select the Functions option with the following operators: *is, is not, is greater than, is greater than or equal to, is less than* and *is less than or equal to*.
4. Rules
   Use the Rules options to compare an input data value to the output from another rule specification. The rule specification that you select runs as part of the rule statement at run time.

   Select the Rules option with the following operators: *is, is not, is greater than, is greater than or equal to, is less than* and *is less than or equal to*.

5. Mapplets
   Use the Mapplets options to compare an input data value to the output from mapplet. The mapplet runs as part of the rule statement at run time.

   Select the Mapplets option with the following operators: *is, is not, is greater than, is greater than or equal to, is less than* and *is less than or equal to*.

   For more information on mapplet configuration, see "Mapplet configuration options" on page 40.
6. Expressions
   Use the Expressions options to define and apply a custom expression to the input data. You can enter an expression as text, or you can select from a range of built-in expressions.

   Select the Expressions option with the following operators: *is, is not, is greater than, is greater than or equal to, is less than* and *is less than or equal to*.
7. Search
   Use the Search options to find an input data value in another input column. The rule statement searches all of the values in the column for the current rule statement input value.

   Select the Search option with the *contains* operator.

Operators

An operator is a mathematical function that determines the outcome of a condition.

You can select one of the following operators:

is

Verifies that the input data matches the data that the condition specifies.

is not

Verifies that the input data does not match the data that the condition specifies.

is within

Verifies that the input data matches a value in a dictionary or in a list of values that you enter.

is not within

Verifies that the input data does not match a value in a dictionary or in a list of values that you enter.

contains

Verifies that the input data contains a sequence of characters that you specify.

is less than

Verifies that the input data value is less than the value that the condition specifies. Select the operator with numeric input data.

is less than or equal to

Verifies that the input data value is less than or equal to the value that the condition specifies. Select the operator with numeric input data.

is greater than

Verifies that the input data value is greater than the value that the condition specifies. Select the operator with numeric input data.

is greater than or equal to

Verifies that the input data value is greater than or equal to the value that the condition specifies. Select the operator with numeric input data.

Actions

An action specifies the output from a rule statement. An action generates an output value when the conditions in the rule statement validate an input value.

Configure an action to generate an output in one of the following ways:

Return a value that you enter

Configure an action to return a data value that you specify.

For example, you might configure an action to return the word VALID when the input data matches a dictionary value.

Return a value from an input column

Configure an action to return a data value from an input in the rule specification. The action reads the data value on the corresponding row of the input that you select.

For example, a business might define a rule that requires all product records to use Quick Response (QR) codes in place of bar codes. You configure an action to replace an input column of bar code values with a corresponding column of QR code values.

Return a value from a dictionary

Configure an action to return a data value from a dictionary. The action compares the input data to the values in the dictionary that you specify. If the action finds the input data in the dictionary, the action returns the dictionary value that corresponds to the input data.

For example, you might configure an action to compare first name data to a dictionary that identifies the gender of the first name. The action might read the word "JOHN" from the input data and return the word "MALE" from the dictionary.

Return a value that you enter when an input data value matches a dictionary value

Configure an action to return a data value from a dictionary. The action compares the input data to the values in the dictionary that you specify. If the action finds the input data in the dictionary, the action returns the string that you entered. if the action does not find the input data in the dictionary, the action returns the input value.

For example, you might configure an action to compare employee code data to a dictionary that contains the valid employee codes in an organization. You define a function that returns the word "VALID" when an employee code matches a value in the dictionary. If the function does not find an employee code in the dictionary, the action returns the employee code from the input column.

Return the current date and time

Configure an action to return the current date and time. The rule statement returns the date and time values in a date/time data type.

Remove data values from the input data

Configure an action to remove a data value that you specify.

For example, the input data might include a product name that the business no longer uses. Configure the action to return a version of the input data that omits the product name.

Remove dictionary values from the data

Configure an action to remove dictionary values from an input.

The action compares the input data to the values in a dictionary that you specify. If the action finds an input data value in the dictionary, the action returns a version of the input without the data value.

For example, you might configure an action to compare an input string to a dictionary of salutation terms. The action might read the input string "MR JOHN SMITH" and return the string "JOHN SMITH" as the output.

Remove character spaces from the input data

Configure an action to delete character spaces from the input data. Remove character spaces when the input data includes redundant character spaces.

For example, you might configure an action to remove character spaces from the following telephone number:

```
212 555 1234
```

Use a function expression to determine the output value

Configure an action to apply a function expression to the input data.

For example, you might configure the action to return the character length of the input data.

Use a custom expression to determine the output value

Configure an action to apply an expression that you define to the input data.

For example, you might configure the action to concatenate two input data values.

Merge values from multiple inputs

Configure an action to merge data from two or more inputs.

For example, you might configure an action to merge a column of first name data and a column of second name data. The action returns a single field that contains the first name and second name.

Change the character case of the input data

Configure an action to convert the character case of an input that you select.

For example, a rule statement might read a data set that stores data values in sentence case. Configure an action to return the data in uppercase.

Return an output form a rule specification

Configure an action to run a rule specification on one or more inputs that you specify. Select at least one input on the rule specification, or enter a value as a rule specification input. You can select any input available to the current rule specification as an input. You also select the field that the action reads as the output of the rule specification.

Return an output from a mapplet

Configure an action to run a mapplet on one or more inputs that you select. Select at least one input on the mapplet, or enter a value as a mapplet input. You can select any input available to the current rule specification as an input. You also select the field that the action reads as the mapplet output.

You can configure a condition or an action with up to four nested mapplets and rule specification assets in an alternating manner. For more information, see "Rules and guidelines for mapplets in conditions and actions" on page 33.

Note: A rule specification can read mapplets that contain passive transformations.

Generate a status value

Configure an action to generate a predefined status value when the input data satisfies the condition on the rule statement.

You can configure an action to return the status values *Valid* and *Invalid*. If you select *Invalid* as the status value, you can add data to the record to explain the status and to identify the priority of the issue that the value represents.

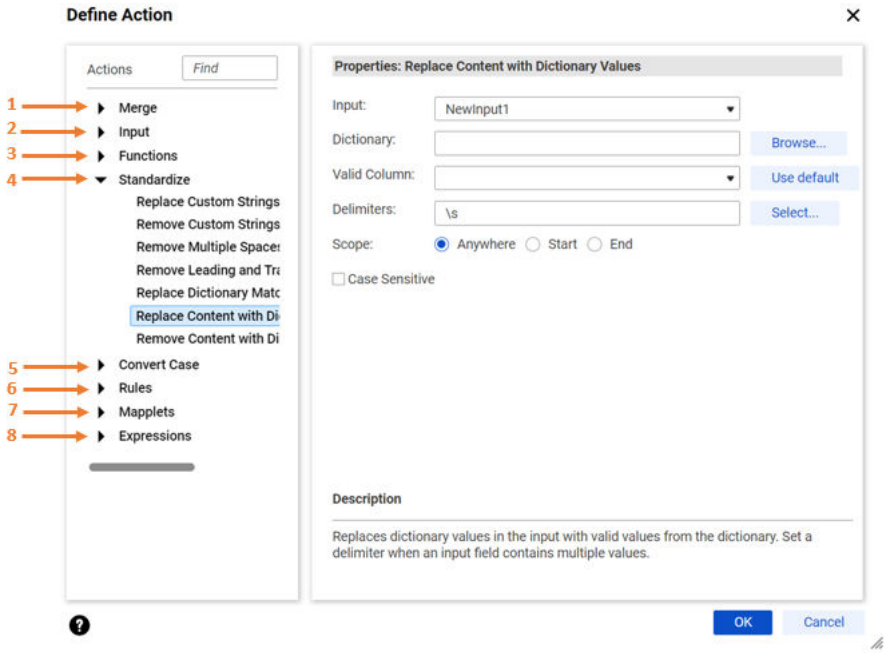
Function configuration in actions

You can configure the data operation in an action as a function expression. You can also select a function expression from a list of expressions in the rule specification. Use the function options in the action to configure or select the expression.

You can use the function options as an alternative to the inline action options that the rule statement displays in the workspace. You can configure a function to read multiple inputs. You can also use the function options to add a dictionary or another rule specification to the condition.

To configure an expression, select the function option from the action menu. When you select the field, you open the **Define Action** dialog box. The dialog box displays a common list of function expressions for all operators.

The following image shows the **Define Action** dialog box:



The dialog box contains the following options:

1. Merge
   Use the Merge options to return data values from multiple inputs as a single output.
2. Input
   Use the Input options to return a data value. You can specify a constant value, or you can select an input. If you select an input, the action returns the value from the input column on the same row as the condition input.

3. Functions
   Use the Functions options to apply a function expression to the input that you select and to return the output of the expression.
4. Standardize
   Use the Standardize options to remove an input value or to update a copy of the input data value and return the copy. The section below describes the standardization options.
5. Convert Case
   Use the Convert Case options to change the character case of the input data values.
6. Rules
   Use the Rules options to return the output of a rule specification that you add to the function.
7. Mapplets
   Use the Mapplets options to return the output of a mapplet that you add to the function.
8. Expressions
   Use the Expressions options to define and apply a custom expression to the input that you select and to return the output of the expression. You can enter an expression as text, or you can select from a range of built-in expressions.

Stardardization options

The Standardization options contain several properties that you can configure to remove or replace data.

You can configure the properties to perform the following actions:

Remove Values

Searches an input field for one or more values that you specify, and removes the values.

You can enter one or more values that the asset will remove. Or, you can use a dictionary to identify the values that the asset will remove.

Remove Spaces

Removes redundant character spaces from an input field.

You can remove leading and trailing spaces, and you can replace multiple spaces with a single space.

Replace Values

Searches an input field for one or more values that you specify, and replaces the values. You can find and replace values in the following ways:

- Enter one or more values that the asset can search for. Enter a single replacement value for every value that the asset finds.
- Use a dictionary to identify multiple strings that the asset can search for. Use the same dictionary to specify a replacement value for each value that the asset finds.
- Use a dictionary to identify multiple strings that the asset can search for. Enter a single replacement value for every value that the asset finds.

When you select a dictionary in an option, you must specify the column in the dictionary that contains the valid values. You must also confirm one or more delimiters for the data values that the action reads.

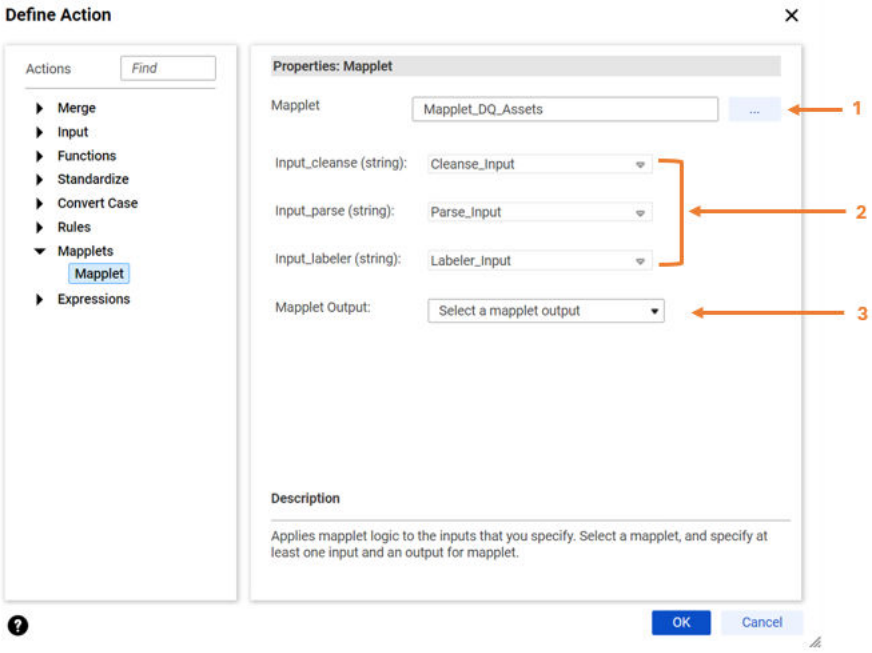
Mapplet configuration options

When you add a mapplet to a condition or action, you must configure at least one input on the mapplet. A mapplet may specify one or multiple inputs, depending on its design. The rule statement additionally reads a single output from the mapping.

Before you configure the inputs, review the design of the mapplet. Verify that your input selections enable the mapplet to generate the data outcomes that the condition or action requires.

For example, you may decide to configure a single input on a mapplet that supports multiple inputs. The data on the field that you specify will flow through the mapplet at run time, provided that the mapplet logic does not require any additional data input. If the input that you select has a dependency on other inputs on the mapplet, configure those inputs also.

The following image shows the options for a sample mapplet in the **Define Action** dialog box:



The dialog box includes the following options:

1. Mapplet name and selection options.

   The options open the **Select Mapplet** dialog box, which displays the mapplets on the system.

2. Mapplet inputs.

   The number of inputs depends on the mapping design. You can assign an input from the rule specification to a mapplet input, or you can enter a value as the input.

3. Mapplet output.

   The rule statement reads a single output from the mapplet as a condition to evaluate or as the data outcome of an action.

Custom expression configuration options

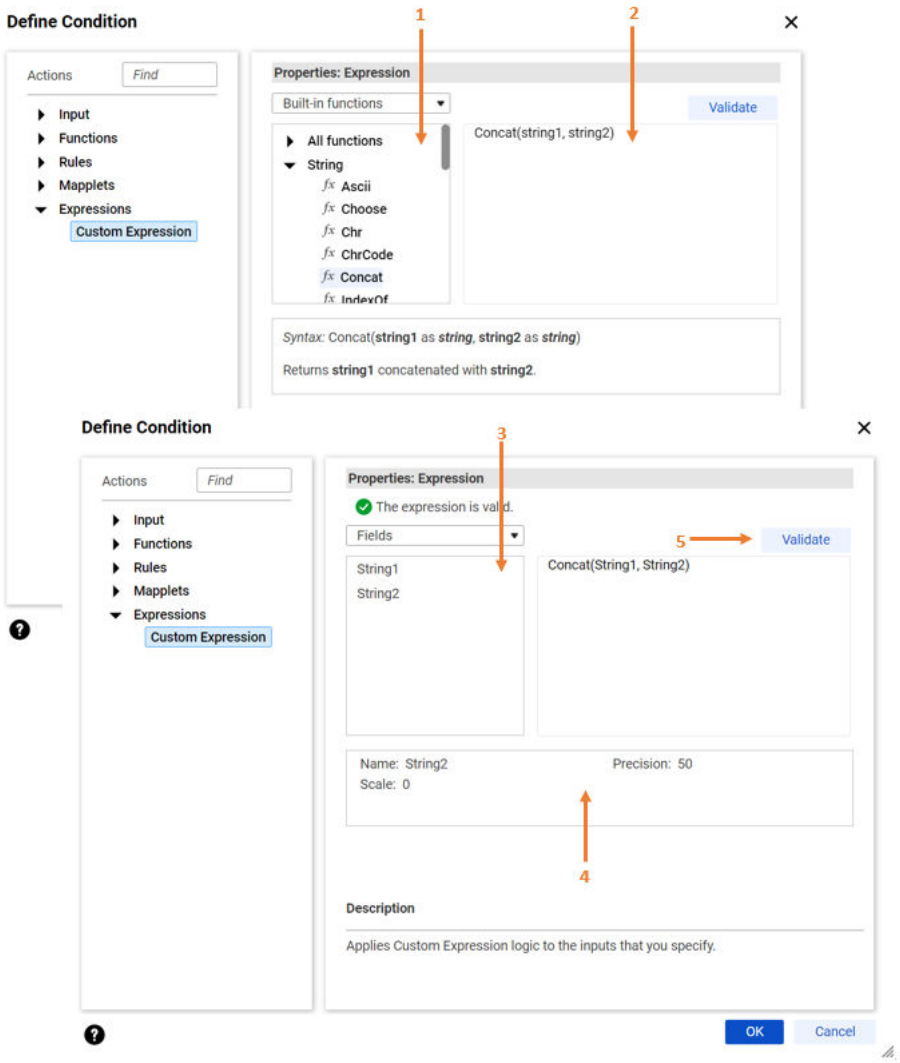
A custom expression is a logical element in a condition or an action. When you configure a custom expression, you enter or select the expression logic and you select the fields that the expression will apply to.

When you add a custom expression to a condition, the custom expression applies the expression logic that you specify to the input data that you select and returns the result to the condition. The condition uses the rule statement operator to determine the next step. When you add a custom expression to an action, the action applies the expression logic to the input data that you select and returns the result as the rule statement output.

To add a custom expression to a condition, use the Custom expression option in the **Define Condition** dialog box. To add a custom expression to an action, use the Custom expression options in the **Define Action** dialog box.

Before you add the custom expression to a condition or action, you can validate the expression.

The following image shows the options for a sample custom expression in the **Define Condition** dialog box:



The dialog box includes the following options:

1. Built-in functions option.

   Displays a library of custom expressions that you can apply to the input data.

2. Expression field.
   Displays the expression logic.

3. Fields option.
   Displays the input fields that you can select.

   Click **Add** next to the field that you want to use.
4. Properties field.
   Displays the details of the input field or the built-in function that you select.
5. Validate option.
   Validates the custom expression.

Note: You select the Built-in functions option and the Fields option from the same menu.

Exception data properties

You can configure a rule specification to add exception data indicators to a record. Use the exception data properties to add outputs for the indicators to the rule specification.

When you configure the exception data properties, you enable an output field for status values that can summarize the data quality issue that the rule logic identifies in a record. You also enable an output for values that indicate the priority of each issue and an output to contain a description of each issue. The output values allow a downstream user to find exception records and to understand the reasons why a record is an exception.

A rule specification asset can specify a single set of output fields for exception indicators. In basic mode, configure the exception indicators as outputs from a rule statement in the primary rule set. If you add a rule specification within a rule statement in basic mode, take care that the child rule specification does not create outputs for exception indicators.

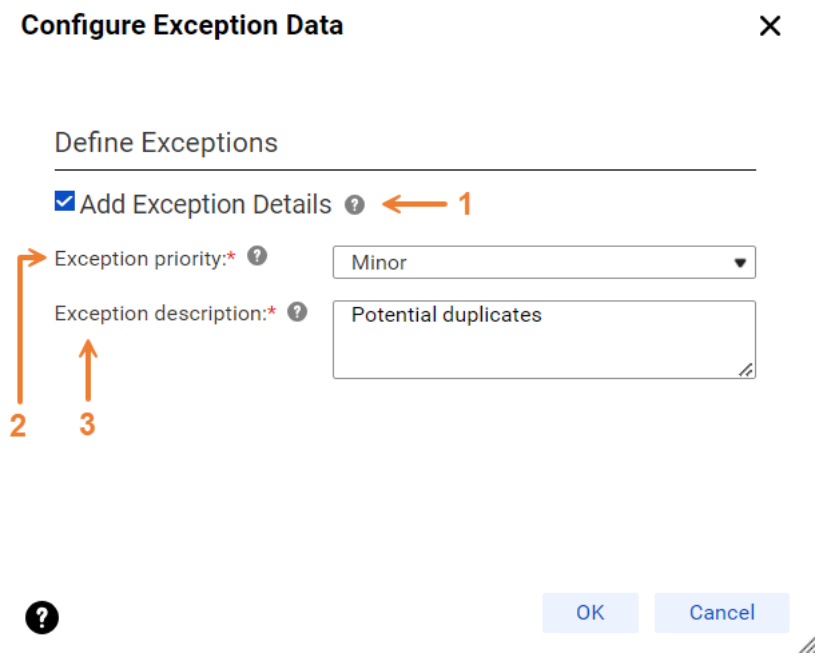
Note: Configure the options and add the rule specification to a profiling task before you create the exception task that reads the profiling task. If a profiling task reads a rule specification in which the exception data options are not configured, configure the exception options and refresh the rule specification in the profiling task before you create the exception task.

Exception data options in basic mode

To add exception indicators to a record, activate the exception indicators in a rule statement that specifies the status value *Invalid* as an action.

Use the **Configure Exception Data** dialog box options to define the indicators. To open the dialog box, click the **+ Exception** option beside the action properties.

The following image shows the options on the **Configure Exception Data** dialog box:



The dialog box contains the following options:

1. Add Exception Details.
   Indicates whether the rule statement adds exception indicators to records. The option is cleared by default.

   The rule statement can add indicators to a record if it assigns *Invalid* as a status value to the record. You specify the indicators in the **Exception priority** and **Exception description** fields.
2. Exception priority.
   Indicates the priority of the data quality issue that defines the record as an exception. The rule statement adds a field to the output records for the priority indicators.

   You can select one of the following options:

   - Minor
   - Major
   - Critical
3. Exception description.
   Describes the data quality issue that defines the record as an exception. The rule statement adds a field to the output records for the description.

Configure all of the dialog box options if you will use the rule specification in a profiling task from which you create an exception task.

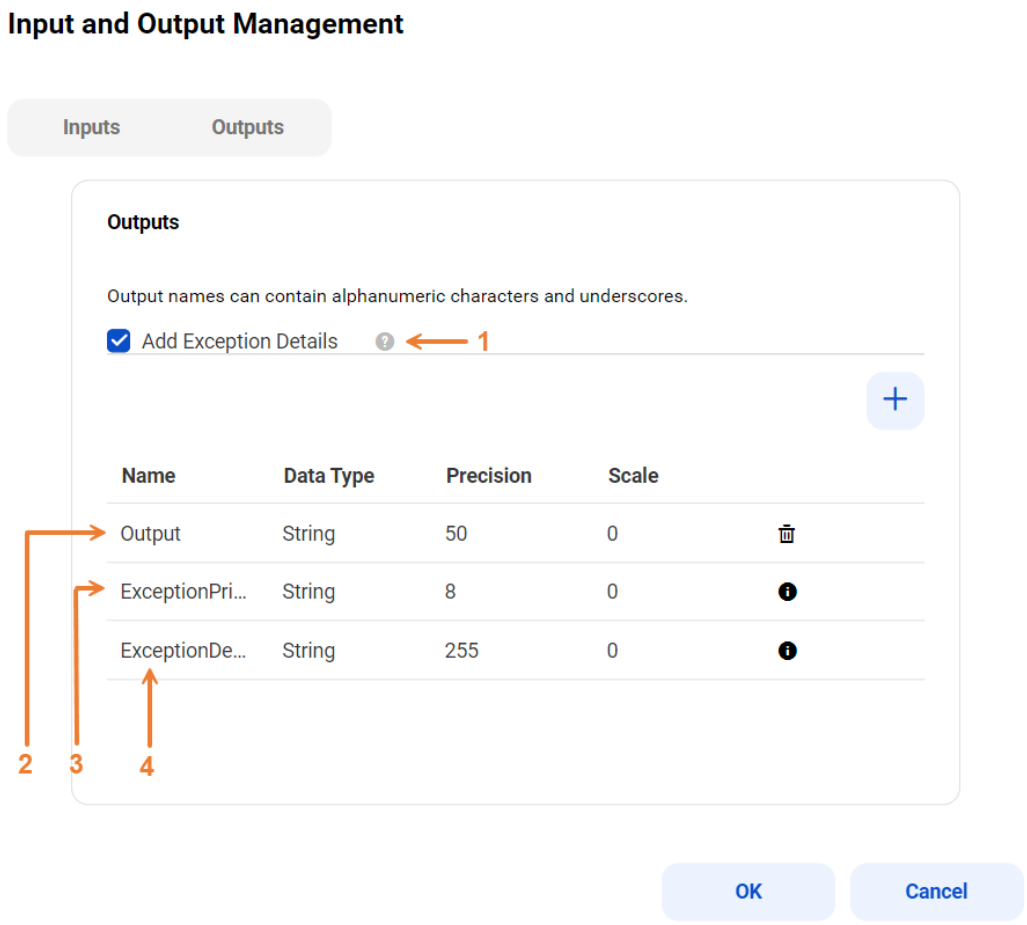
Note: The dialog box does not identify an output field for status values. You activate the output for status values when you configure the Invalid option in the rule statement action.

Exception data options in advanced mode

To add exception indicators to a record, activate exception indicators in the rule specification. You can include the exception indicators in the rule logic that you write.

Use the **Input and Output Management** dialog box options to activate the exception indicators. To open the dialog box, click **Manage Inputs and Outputs**.

The following image shows the options on the **Outputs** tab in the **Input and Output Management** dialog box:



The dialog box contains the following options:

1. Add Exception Details.
   Indicates whether the expression logic can add exception indicators to records. The option is cleared by default.
2. Output field for status values.
   Identifies an output field that the rule specification can use for status values. You can rename the status field in advanced mode.
3. Exception priority field.
   The name of the output field that indicates the priority of the data quality issue in the output field. The statement adds the ExceptionPriority field to the output data.

   You can specify one of the following options when you configure the exception indicator:

   - Minor
   - Major
   - Critical
4. Exception description field.
   The name of the output field that describes the data quality issue in the output field. The statement adds the ExceptionDescription field to the output data.

Function descriptions

A function expression is a logical element in a condition or an action. A function expression reads the input data that you specify, performs a calculation on the input data, and returns a result.

When you add a function expression to a condition, the function expression performs the calculation and returns the result to the condition. The condition uses the rule statement operator to determine the next step. When you add a function expression to an action, the action performs the calculation and returns the result as the rule statement output.

To add a function expression to a condition, use the Function options in the **Define Condition** dialog box. To add a function expression to an action, use the Function options in the **Define Action** dialog box.

You can add the following function expressions to a condition or an action:

Add to Date

Reads a date value and adds a numeric value to an element in the date. Enter the value to add to the element, or select an input that contains the value to add. Select the function expression in a condition or an action.

Choose

Reads input data that contains multiple values and selects the value at the position that you specify. Select the function expression in a condition or an action.

For example, the following string contains values at four positions:

```
Q1, Q2, Q3, Q4
```

You can enter a position, or you can select an input to specify a position.

Concatenate

Reads two inputs that you select and concatenates the values from the inputs. Select the function expression in a condition or an action.

Convert to Date

Converts a value in a string data type to a date/time data type. You identify the input data format when you configure the function expression. Select the function expression in a condition or an action.

Convert to Float

Converts a value in a string data type to a float data type. Select the function expression in a condition or an action.

Convert to Integer

Converts a string or numeric value to an integer data type. Select the function expression in a condition or an action.

You can set the function expression to truncate an input value at the decimal point or to round an input value to the nearest integer.

The function expression treats an alphanumeric value that begins with an alphabetical character as a null value and returns 0 as the output. The function expression treats an alphanumeric string that begins with a digit as a numeric value and truncates the value at the first alphabetical character.

Date Difference

Reads date values from two inputs that you select and calculates the numeric difference between the dates. You select the date element that the function expression uses to measure the difference. Enter the date element, or select an input that contains the date element. Select the function expression in a condition or an action.

Date Part

Reads a date from an input and identifies the numeric value of a date element that you specify. Enter the date element, or select an input that contains the date element. Select the function expression in a condition or an action.

Date Time

Uses multiple values to specify a date and time. Enter a value for each date element, or select an input to add a value to each element. Select the function expression in a condition or an action.

Greatest

Reads two or more values and identifies the highest value. Enter the values, or select rule set inputs to add each value. You can specify numeric values, alphabetic values, or date values. Select the function expression in a condition or an action.

When you specify numeric values, the function expression identifies the highest number.

When you specify alphabetic values, the function expression identifies the value that is last in alphabetic order.

When you specify date values, the function expression identifies the latest date.

Is Date

Returns a value that indicates whether the input string is valid date. Select the function expression in a condition or an action.

Is Number

Returns a value that indicates whether the input string is valid number. Select the function expression in a condition or an action.

Is Spaces

Returns a value that indicates whether the input string contains character spaces only. Select the function expression in a condition or an action.

Last Day

Reads a date value from an input and identifies the last day in the month that the date specifies. The function expression identifies the last day as a numeric value. Select the function expression in a condition or an action.

Least

Reads two or more values and identifies the lowest value. Enter the values, or select rule set inputs to add each value. You can specify numeric values, alphabetic values, or date values. Select the function expression in a condition or an action.

When you specify numeric values, the function expression identifies the lowest number.

When you specify alphabetic values, the function expression identifies the value that is first in alphabetic order.

When you specify date values, the function expression identifies the earliest date.

Length

Reads an input value and calculates the number of characters in the value. Select the function expression in a condition or an action.

Lower

Reads an input string and returns the lowercase version of the characters in the string. Select the function expression in a condition.

Mod

Returns the remainder of a division operation. Select the function expression in a condition or an action.

Null

Reads an input value and determines if the value is null. Select the function expression in a condition or an action.

Replace Character

Reads an input value and replaces a sequence of characters in the value with a sequence that you specify. Select the function expression in a condition.

Replace String

Reads an input value and replaces the value with a value that you specify. Select the function expression in a condition.

Reverse

Reads an input value and reverses the order of the characters in the value. Select the function expression in a condition or an action.

Substring

Returns a substring from an input value of a length that you specify, beginning at a position that you specify. Select a string or numeric data type as an input. Select the function expression in a condition or an action.

To Char

Converts a numeric value or a date value to a string value. Select the function expression in a condition or an action.

Truncate

Reads an input value and truncates the value to the number of places that you specify. The function expression reads a numeric value. Enter the value, or select an input to add the value. Select the function expression in a condition or an action.

Upper

Reads an input string and returns the uppercase version of the characters in the string. Select the function expression in a condition.

Rule statement menu options

You can open a set of menu options for each rule statement. Use the menu options to organize the rule statements in the rule set.

The menu has the following options:

Add Condition

Add a condition to the current rule statement. When you configure multiple conditions in a rule statement, you create an AND relationship between the conditions.

Add Rule Statement

Add an empty rule statement to the rule set.

Add Rule Statement Using Result

Add a rule statement to the rule set below the current rule statement. The rule statement that you add uses the output from the current rule statement as an input.

Copy Rule Statement

Copy the current rule statement. You can paste the rule statement to the current rule set or to another rule set.

Cut Rule Statement

Remove the current rule statement from the rule set. You can paste the rule statement to another rule set.

Move Down

Reorder the rule statements so that the current rule statement changes places with the rule statement below it.

Move to Bottom

Move the current rule statement to the bottom of the rule set.

Move to Top

Move the current rule statement to the top of the rule set.

Move Up

Reorder the rule statements so that the current rule statement changes places with the rule statement above it.

Paste Rule Statement

Paste a rule statement that you copied or moved in an earlier action.

Configuring a rule statement

Before you configure the rule statement, verify that the rule specification contains the inputs that the rule statement requires.

1. Open the rule specification.
2. Select the rule set to contain the rule statement.
3. Click **Rule Logic**.
4. Click **Add Rule Statement**.
5. Add an input to the rule statement. The condition reads the input that you add.
6. Select an operator. The operator specifies the type of comparison operation that a condition performs.
7. Configure the condition.

   Identify one or more data values that the condition can compare to the rule statement input. Or, configure a data operation to apply to the input.
8. Configure the action to perform when the condition validates the input data.

   Identify the input data or enter a data value that the action can return as the rule statement output. Or, configure a data operation that defines the output. Use the function options to configure the operation.
9. Save the rule specification.

Configuring a rule statement that reads the result of another rule statement

You can create a chain of rule statements in which each statement reads the result of the statement above it. Use the rule statements to derive information from the relationship between multiple inputs.

1. Open the rule specification.
2. Select the rule set.
3. Click **Rule Logic**.
4. Select the rule statement that generates the result that another rule statement must read.

   Or, configure a rule statement.
5. From the rule statement menu, select **Add Rule Statement Using Result**.

   Data Quality adds an empty rule statement under the rule statement that you select.
6. Configure and save the rule statement.

Note: Data passes through a chain of rule statements when all of the logical operations in the statements are valid for the inputs that you select. If a rule statement in the chain does not generate a result for a data row, the chain performs no further analysis on the row.

Configuring a rule statement with multiple conditions

You can configure multiple conditions in a single rule statement. When you use multiple conditions, you create an AND relationship between the conditions.

You configure a single action in the rule statement.

1. Open the rule specification, and click **Edit**.
2. Select the rule set to contain the rule statement.
3. Click **Rule Logic**.
4. Click **Add Rule Statement**.
5. Add an input to the rule statement. The condition reads the input that you add.
6. Select an operator. The operator specifies the type of comparison operation that the condition performs.
7. Configure the condition.

   Identify one or more data values that the condition can compare to the rule statement input. Or, configure a data operation to apply to the input.
8. From the rule statement menu, select **Add Condition**.

   Data Quality adds a condition to the workspace.
9. Configure the condition.

   You can add multiple conditions.
10. Configure the action for the rule statement.

    Data Quality adds the action after the final condition in the rule statement.
11. Save the rule specification.

Adding a rule specification to a rule statement

You can reference a rule specification in a condition or an action in a rule statement. Add a rule specification to a rule statement when the rule specification contains function logic that you want to reuse in the rule statement. Use the rule specification so that you do not have to define the logic in the current rule specification.

Configuring a condition to run a rule specification

1. Create or select a rule statement in a rule set.
2. Select one of the following operators:

   `is, is not, is greater than, is greater than or equal to, is less than` or `is less than or equal to`.

   The condition name changes to function.
3. Open the **Define Condition** dialog box.
4. Select **Rules** > **Rule Specification**.
5. In the **Select Rule Specification** dialog box, browse to and select a rule specification to add to the condition.
6. The **Define Condition** dialog box shows the inputs on the rule specification that you select.

   Select an input in the current rule set to map to each input in the selected rule specification. The data types of the inputs must be compatible in each case.
7. The **Define Condition** dialog box also shows an output field for the rule specification that you select. The output field identifies the rule set that generates an output for the condition.

   To provide an output from the rule specification that the condition can use, select a rule set name.
8. Click **OK**.

Configuring an action to run a rule specification

1. Create or select a rule statement in a rule set.
2. Select the function option on the action.
3. Open the **Define Action** dialog box.
4. Select **Rules** > **Rule Specification**.
5. In the **Select Rule Specification** dialog box, browse to and select a rule specification to add to the action.
6. The **Define Action** dialog box shows the inputs on the rule specification that you select.

   Select an input in the current rule set to map to each input in the selected rule specification. The data types of the inputs must be compatible in each case.
7. The **Define Action** dialog box also shows an output field for the rule specification that you select. The output field identifies the rule set that generates an output for the condition.

   To provide an output from the rule specification that the action can use, select a rule set name.
8. Click **OK**.

Cut and paste operations and rule statements

Copy a rule statement to reuse the rule statement logic in the same rule set or in another rule set. Move a rule statement that belongs in another rule set. You can copy or move a rule statement to a rule set in the same rule specification or in another rule specification.

You might copy a rule statement to create a rule statement based on the logic of the rule statement that you copy. You might move a rule statement when you determine that the rule statement applies to a different business rule requirement.

When you copy or move a rule statement to another rule specification, you also copy the inputs that the rule statement uses. The rule statement that you copy must be valid within the destination rule set. For example, the rule statement and the inputs on the rule set must be compatible.

Copying a rule statement within a rule set

To create multiple similar rule statements in a rule specification, copy the rule statement. You can copy and paste a rule statement within the same rule set or to another rule set. After you copy the rule statement, update it to reflect the business rule logic.

Move a rule statement when you determine that the rule statement belongs in another rule set.

1. Open the rule specification.
2. Select the rule set that contains the rule statement to copy or move.
3. Select **Rule Logic**.
4. Select the rule statement. If the rule statement contains more than one condition, select the first line of the rule statement.
5. Open the actions menu.
   - To copy the rule statement, select **Copy Rule Statement**.
   - To move the rule set, select **Cut Rule Statement**.

   Note: Find the actions menu option beside the Action options.
6. Select a destination rule set for the rule statement.

   You can create a copy of a rule statement in the current rule set or in another rule set.
7. Select a rule statement in the rule set.
8. Open the rule statement context menu, and select **Paste Rule Statement**.

   The rule statement that you copied or moved appears in the rule set.
9. Verify the position of the rule statement in the rule set.

   If necessary, move the rule statement up or down in the list of list of rule statements.
10. Save the rule specification.

Copying a rule statement to another rule specification

When you create multiple rule specifications with similar logic, you can copy the rule statements from one rule specification to another to save time. After you copy the rule statement, you might update it to reflect the business requirements of the current rule specification.

Move a rule specification when you determine that the rule set belongs in another location.

1. Open the rule specification.
2. Select the rule set that contains the rule statement to copy or move.

3. Select **Rule Logic**.
4. Select the rule statement. If the rule statement contains more than one condition, select the first line of the rule statement.
5. Open the actions menu.
   - To copy the rule statement, select **Copy Rule Statement**.
   - To move the rule set, select **Cut Rule Statement**.

   Note: Find the actions menu option beside the Action options.
6. Click **Explore**.
7. Select a destination rule specification for the rule set.

   The rule specification opens in the Configuration tab.
8. Select a rule set, and select **Rule Logic**.
9. Select a rule statement in the rule set.
10. Open the actions menu on the rule statement, and select **Paste Rule Statement**.

    The rule statement that you copied or moved appears in the rule set.
11. Verify the position of the rule statement in the rule set.

    If necessary, move the rule statement up or down in the list of list of rule statements.
12. Save the rule specification.

    If you moved a rule statement from another rule specification, save both rule specifications.

CHAPTER 5

Rule specification configuration in advanced mode

A rule specification asset in advanced mode provides a rule editor in which you can write rule logic in simple expressions or in IF-THEN-ELSE syntax.

You can nest an IF-THEN-ELSE statement within another statement. You can specify other rule specification assets, dictionaries, mapplets, and complex expressions in the logic. You receive suggestions as you type in the editor, such as input or output field names, functions, and mapplets.

You can define the same business rule logic in basic mode and advanced mode.

Rule specification logic in advanced mode

In advanced mode, you write your business rule in expression logic. Within an IF-THEN-ELSE conditional statement, you can use ELSEIF conditions and nest additional IF-THEN-ELSE statements.

An IF-THEN-ELSE statement uses the following format:

```
// Primary condition
IF  <Boolean expression>
THEN <Rule Block>
// Optional - Multiple ELSEIF conditions
ELSEIF  <Boolean expression>
THEN  <Rule Block>
// Optional ELSE condition
ELSE  <Rule Block>
ENDIF
```

Before you write the logic, verify that you understand the business data and the requirements of the business rule.

You receive suggestions as you type in the editor. Enter a character space after you type each element in the editor.

Rule logic configuration examples

The elements that you include in your rule logic depend on the facts that you need to determine about the input data.

You might can write the rule logic to include the following elements:

- Other rule specifications
- Mapplets
- Dictionaries
- Exception data options

Specifying a rule specification in an IF element

You can configure an IF element in your rule specification to read a rule specification that you identify:

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.
3. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement.
4. Type the name of a rule specification that you want to add.

   The editor can suggest rule specification names from the set of assets in your organization folders or projects based on the text that you type.
5. Map each input in the current rule specification to an input in the rule specification that you select above. Append the input names from the rule specification that you selected as parameters to the rule specification name.

   Note: The data types of the inputs that you map must be compatible.
6. Enter an operator value. For example, enter the `=` value to verify whether the input value matches the value that you specify.
7. Enter the value to compare to the input value. For example, enter `VALID`.
8. Type the `THEN` element.
9. To specify the output, type the output name.
10. Enter an operator value. For example, enter the `=` value to write a value for the data that satisfies the rule logic.
11. Enter the value that the output returns. For example, enter `Input not null`.
12. Type the `ENDIF` element.

The following example shows rule logic that contains a rule specification:

```
IF rule:Auto_Project1.TC593_InnerRule_AdvancedRulespec(Input) = 'VALID'
THEN
Output = 'Input not null'
ENDIF
```

The italicized text represents the rule specification name.

Specifying a rule specification in a THEN element

You can configure a THEN element in your rule specification to read a rule specification that you identify:

1. Open a rule specification in advanced mode.

2. Verify that the rule specification contains the inputs and outputs that you require.
3. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement.
4. To specify the input, type the input name.
5. Enter an operator value. For example, enter the `!=` value to write a value for the data that does not satisfy the rule logic.
6. Specify a condition for the operator. For example, enter `NULL`.
7. Type the `THEN` element.
8. To specify the output, type the output name.
9. Enter an operator value. For example, enter the `=` value to write a value for the data that satisfies the rule logic.
10. Type the name of a rule specification that you want to add.

    The editor can suggest rule specification names from the set of assets in your organization folders or projects based on the text that you type.
11. Map each output in the current rule specification to an output in the rule specification that you select above. Append the output names from the rule specification that you selected as parameters to the rule specification name.

    Note: The data types of the outputs that you map must be compatible.
12. Type the `ENDIF` element.

The following example shows rule logic that contains a rule specification:

```
IF Input != NULL
    THEN
    Output = rule:Default.rule 1 for TC53(Input1)
ENDIF
```

The italicized text represents the rule specification name.

Specifying a mapplet in an IF element

You can configure an IF element in your rule specification to read a mapplet that you identify:

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.
3. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement.
4. Type the name of a mapplet that you want to add.

    The editor can suggest rule specification names from the set of assets in your organization folders or projects based on the text that you type.
5. Map each input in the current rule specification to an input in the mapplet that you select above. Append the input names from the rule specification that you selected as parameters to the mapplet name.

    Note: The data types of the inputs that you map must be compatible.
6. Enter an operator value. For example, enter the `=` value to verify whether the input value matches the value that you specify.
7. Enter the value to compare to the input value. For example, enter `DIGIT`.
8. Type the `THEN` element.
9. To specify the output, type the output name.
10. Enter an operator value. For example, enter the `=` value to write a value to the data that satisfies the rule logic.

11. Enter the value that the output returns. For example, enter `Input given is DIGIT`.
12. Type the `ENDIF` element.

The following example shows rule logic that contains a mapplet:

```
IF mapplet:Validation_RuleSpec_1.Mapplet_RuleSpecVal(Input)='DIGIT'
THEN
Output= 'Input given is  DIGIT'
ENDIF
```

The italicized text represents the mapplet that you use in the rule specification.

Specifying a mapplet in a THEN element

You can configure a THEN element in your rule specification to read a mapplet that you identify:

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.
3. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement.
4. To specify the input, type the input name.
5. Enter an operator value. For example, enter the `!=` value to write a value for the data that does not satisfy the rule logic.
6. Specify a condition for the operator. For example, enter `NULL`.
7. Type the `THEN` element.
8. To specify the output, type the output name.
9. Enter an operator value. For example, enter the `=` value to write a value to the data that satisfies the rule logic.
10. Type the name of a mapplet that you want to add.

    The editor can suggest rule specification names from the set of assets in your organization folders or projects based on the text that you type.
11. Map each output in the current rule specification to an output in the mapplet that you select above. Append the output names from the rule specification that you selected as parameters to the rule mapplet name.

    Note: The data types of the outputs that you map must be compatible.
12. Type the `ENDIF` element.

The following example shows rule logic that contains a mapplet:

```
IF Input != NULL
    THEN
    Output = mapplet:ER_11227.MappletRS(Input1)
ENDIF
```

The italicized text represents the mapplet that you use in the rule specification.

Specifying a dictionary in the expression logic

You can configure an IF or THEN element in your rule specification to read a dictionary that you identify.

The following task describes the steps to add a dictionary to an IF element.

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.

3. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement. To specify the input, type the input name.
4. Enter an operator value. For example, enter the `in` value to identify input values that match the dictionary values.
5. To apply a dictionary to the input data, type the name of the dictionary that you created.

   The editor can suggest dictionary names from the set of assets in your organization projects or folders based on the text that you type.
6. Type the `THEN` element.
7. To specify the output, type the output name.
8. Enter an operator value. For example, enter the `=` value to write a value for the data that satisfies the rule logic.
9. Enter the value that the output returns. For example, enter `Input present`.
10. Type the `ELSE` element.
11. To specify the output, type the output name.
12. Enter an operator value. For example, enter the `=` value to write a value for the data that does not satisfy the rule logic.
13. Enter the value that the output returns. For example, enter `Input not present`.
14. Type the `ENDIF` element.

The following example shows rule logic that contains a dictionary:

```
IF Input in dictionary:Validation_RuleSpec_1.Dictionary_RuleSpec
THEN
Output = 'Input present'
ELSE
Output = 'Input not present'
ENDIF
```

The italicized text represents the dictionary that you use in the rule specification.

Specifying exception data options in the expression logic

You can configure exception properties in the expression logic to identify exception records.

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.
3. Select the **Add Exception Details** option on the **Outputs** tab in the **Input and Output Management** dialog box.

   The option activates the exception priority, exception description, and status indicators that you will use in the rule logic.
4. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement.
5. To specify the input, type the input name.
6. Enter the operator value.
7. Specify a condition for the operator.
8. Type the `THEN` element.
9. Define a status indicator:
   - Define an `Output`.

- Enter the = operator.
- Enter a status valuethat reflects the data quality issue that the rule specification finds in the record.

10. Add the `ExceptionPriority` indicator.

    Enter the = operator value to write a priority value.

11. Enter the priority value for the data quality issue that defines the record as an exception.

    You can enter one of the following priority values: Minor, Major, Critical.

12. Add the `ExceptionDescription` indicator.

    Enter the = operator value to write a descriptive text.

13. Enter a description of the data quality issue that defines the record as an exception.
14. To identify records that contain another data quality issue, add an `ELSEIF` element. Write the element logic in the same manner as the IF-THEN sequence in the earlier steps.
15. Type the `ENDIF` element.

The following example shows exception rule logic that identifies individuals who may be old enough to obtain a driver's license:

```
IF Input < 17
THEN
Output = 'Child'
ELSEIF Input = 17
THEN
Output = 'Verify age'
ExceptionPriority = 'Major'
ExceptionDescription = 'Driver may qualify this year'
ELSEIF Input > 17
THEN
Output = 'Adult'
ELSE
Output = 'Not known'
ENDIF
```

The italicized text represents the exception data options.

Rule specification functions

Advanced mode supports a range of built-in functions that you can use in the rule logic. The built-in functions match the functions that you can use in basic mode in the asset.

You receive suggestions as you type function names in the editor. When you select a function in the logic, the asset displays the function signature.

To view the list of functions that you can use in the expression logic, see “Rule specification functions” on page 20.

Rule specification operators

Use the rule specification operators when you write business rule logic.

You can use the following operators:

Operator	Description
=	Verifies that the input value matches the value that the condition specifies.
!=	Verifies that the input value does not match the value that the condition specifies.
in	Verifies that the input value matches a value in a dictionary or in a list of values that you enter.
not in	Verifies that the input value does not match a value in a dictionary or in a list of values that you enter.
contains	Verifies that the input value contains a sequence of characters that the condition specifies.
<	Verifies that the input value is less than the value that the condition specifies.
<=	Verifies that the input value is less than or equal to the value that the condition specifies.
>	Verifies that the input value is greater than the value that you specify.
>=	Verifies that the input value is greater than or equal to the value that you specify.

CHAPTER 6

Business rules and rule statements

You can define all or part of a business rule in a rule statement. Define a rule statement for each fact that a business rule must discover. The type of rule statement that you define depends on the business rule requirements.

Use rule statements to verify and update the following business facts:

Verify the accuracy of business data

Define a rule statement that compares input data to a dictionary.

Use the rule statement to verify that the business data is accurate. For example, define a rule statement to verify that a product description table uses the current product codes.

Find obsolete business data

Define a rule statement to find values that are no longer valid or relevant to the business.

Use the rule statement to return a user-defined value that identifies any row that contains an obsolete term.

Standardize project data values

Define a rule statement that searches for values that you specify and replaces them with other values.

Use the rule statement to verify that the project data follows a standard format or reflects the current project terminology. For example, define a rule statement to verify that network machinery names reflect a change from the Development stage to the Production stage in a project. The rule statement can replace instances of Development with Production in the data.

Improve the usability of business data

Define a rule statement that removes redundant characters and redundant strings.

Use the rule statement to ensure that users and software applications can read the business data correctly. For example, define a rule statement to delete character spaces at the start and end of a data field.

Use data values to discover information about business users

Define a rule statement that correlates information from different inputs. Use different conditions to analyze the inputs, and connect the conditions with AND logic. For example, define conditions to link different facts about customers and to derive information that can improve customer service.

Use data values to verify a business policy

Define a rule statement that verifies that the business follows a stated policy.

Use the rule statement to define a function expression that represents the policy.

Verify that you applied a business rule to a data set

Define a rule statement that adds the current date and time to a data set.

Use the rule statement to add a time stamp to the data set that indicates when the mapping ran.

Use a list of values to find data in a column

Define a rule statement that looks for the values that you enter in a column of data.

Use the rule statement to define a list of data values that a dictionary does not contain.

Identify data values that contain key words or character strings

Define a rule statement that finds a data value or a character string in fields that contain multiple values or longer character strings.

Use the rule statement to find values that contain similar data.

Generate status values

Define a rule statement that generates a predefined status value when the data in an input field satisfies a given condition. For example, define the rule statement to return the value of *Valid* when an input field is not null.

If you define a status value of *Invalid* in the rule statement, you can add data to describe the status of the record.

Combine data values in advanced mode

Write rule logic that combines two or more data values in your data set. For example, write logic to concatenate the data values that you require into a single string. Use AND operator to connect two or more inputs.

Standardize character case in advanced mode

Write rule logic to convert the character case of the input data.

Eliminate redundant character spaces in advanced mode

Write rule logic that removes redundant character spaces.

Standardize data values using a dictionary in advanced mode

Write rule logic that searches for instances of values that appear in a dictionary and replaces each value with a corresponding valid value from the dictionary.

Standardize custom data values in advanced mode

Write rule logic that searches for instances of one or more values that you specify and replaces the values with a single new value.

Verifying the accuracy of business data

To verify the accuracy of a column of business data, configure a rule statement that compares the column to a dictionary. For example, you might configure a rule statement to verify that a data column contains the current product codes for the organization.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.

4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.
5. Select an operator to validate the results of the condition analysis.
   - To identify input values that match the dictionary values, select the following operator:

     ```
     is within
     ```
6. Select the type of condition to apply to the input.
   - To use a dictionary in a condition, select the following condition type:

     ```
     function
     ```
7. Configure the condition to apply a dictionary to the input data.
   - Open the **Define Condition** dialog box.
   - Select **Dictionary**.
   - Browse to the dictionary that contains the standard versions of the business data. For example, select a dictionary that contains a set of product codes.
   - Select or clear the option to perform a case-sensitive search of the dictionary.
   - Click **OK**.
8. Select the type of action to apply to the data that satisfies the condition logic.
   - To return a string value, select the following action type:

     ```
     string value
     ```
9. Enter the value that the action returns. For example, enter VALID.
10. Save the rule specification.

Identifying errors in business data

To find values that are no longer valid in a data set, configure a rule statement to read a dictionary that contains known data errors. For example, a national organization updates the codes that identify branch offices. The organization changes the code structure from eight digits to ten digits. You might configure a rule statement to read a dictionary that contains the obsolete eight-digit codes.

If the rule statement finds an obsolete branch code in the input data, it writes a value that you specify as output.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the branch code data.
5. Select an operator to validate the results of the condition analysis.
   - To specify that the input data must match the data in the condition, select the following operator:

     ```
     is
     ```
6. Select the type of condition to apply to the input.

- To compare the input data to the data from another input, select the following condition type:

  ```
  input
  ```

7. Select the input that you selected in step 4.

   The rule statement compares the input to itself. Therefore, the rule statement passes all of the input data values from the condition to the action.

8. Configure the action to search the input data for obsolete branch codes.
   - Open the **Define Action** dialog box.
   - Select **Standardize**, and select **Replace Dictionary Matches with Custom Strings**.
   - Select the input that you specified in the condition.
   - Select the dictionary that contains the obsolete values. You must also select or verify the valid value column in the dictionary.
   - Enter the data value that the action returns when an input data value matches a dictionary value. For example, enter OBSOLETE_CODE.
9. Click **OK**.
10. Optionally, update the default rule statement to return the input data value if the action does not find a dictionary match with the value. Select *input* as the action type, and then select the input column that you selected in step 4.

    When you update the default rule statement in this way, the action can write an output for each row of input data.
11. Save the rule specification.

Standardizing project data values

To standardize data values, configure a rule statement that replaces an input value with a value that you require. For example, you might configure a rule statement to replace terms that identify a Development environment with terms that identify a Production environment.

To standardize multiple values, you can create additional rule statements. Alternatively, use a dictionary that identifies the values.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.
5. Select an operator to validate the results of the condition analysis.
   - To specify that the input data must match the data that you enter, select the following operator:

     ```
     is
     ```
6. Select the type of condition to apply to the input.
   - To compare the input data to a string value, select the following condition type:

     ```
     string value
     ```
7. Enter the value to compare to the input data. For example, enter DEV.

8. Select the type of action to apply to the data that satisfies the condition logic.
   - To return a string value, select the following action type:
     ```
     string value
     ```
9. Enter the value that the action returns. For example, enter PROD.
10. Save the rule specification.

Improving the usability of business data

To improve the usability of data values, you can delete character spaces at the start and at the end of the values. The redundant spaces might introduce errors when you sort or compare the values.

Configure a rule statement that deletes the redundant spaces. To delete longer character strings, create a rule statement that uses the "Replace Character" function.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.
5. Select an operator to validate the results of the condition analysis.
   - To specify that the condition does not apply to null data, select the following operator:
     ```
     is not
     ```
6. Select the type of condition to apply to the input.
   - To verify that the input values are not null, select the following condition type:
     ```
     null value
     ```
7. Select the type of action to apply to the data that satisfies the condition logic.
   - To standardize the data, select the following action type:
     ```
     function
     ```
8. Configure the action to remove the redundant spaces from the input data.
   - Open the **Define Action** dialog box.
   - Select **Standardize**, and select **Remove Leading and Trailing Spaces**.
   - Select the input that you specified in the condition.
   - Click **OK**.
9. Save the rule specification.

Deriving information about business users

To discover information about business users, configure a rule statement that measures a fact about the users. When you combine multiple conditions to measure related facts, you discover additional information.

For example, a financial institution decides to offer a credit card to all customers with high salaries and credit account balances. You might configure a rule statement to identify customers who hold credit balances. You might also configure a rule statement to identify customers who earn a high salary.

Use the AND feature to combine the rule statements.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.

     Note: The condition analyzes salary data.
5. Select an operator to validate the results of the condition analysis.
   - To identify input values that indicate a minimum salary of $100,000, select the following operator:

     ```
     is greater than or equal to
     ```
6. Select the type of condition to apply to the input.
   - To compare the input data to an integer, select the following condition type:

     ```
     integer value
     ```
7. Enter the value to compare to the input data. For example, enter 100000.
8. Select **Add Condition** in the rule statement menu.

   The rule specification adds a condition under the current condition and creates an AND relationship between the conditions.
9. Select an input for the condition that you added.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.

     Note: The condition analyzes the customer account balances.
10. Select an operator to validate the results of the condition analysis.
    - To identify input values that indicate a minimum balance of $1,000, select the following operator:

      ```
      is greater than or equal to
      ```
11. Select the type of condition to apply to the input.
    - To compare the input data to an integer, select the following condition type:

      ```
      integer value
      ```
12. Enter the value to compare to the input data. For example, enter 1000.
13. Select the type of action to apply to the data rows that meet all of the conditions in the rule statement.
    - To return a string value, select the following action type:

      ```
      string value
      ```
14. Enter the value that the action returns. For example, enter HIGH.
15. Save the rule specification.

Verifying that data values conform to business regulations

To verify that the business follows a stated policy, define a rule statement that applies an expression to key data columns.

For example, you might configure a rule statement that validates mortgage loan applications for a mortgage broker. The mortgage broker refuses any loan application that requires the applicant to repay more than twenty-five percent of their salary each month. The rule statement tests the monthly repayment amount against an expression that calculates the salary percentage.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.

     Note: The input represents the monthly repayment amount on the mortgage loan.
5. Select an operator to validate the results of the condition analysis.
   - To identify input values that are greater than twenty-five percent of another value, select the following operator:

     ```
     is greater than
     ```
6. Select the type of condition to apply to the input.
   - To define an expression to test the input data, select the following condition type:

     ```
     function
     ```
7. Configure the condition to apply an expression to the input data. The expression uses an input that represents the monthly salary of the loan applicant.
   - Open the **Define Condition** dialog box, and select **Input**.
   - Select the float or the integer data type.
   - In the Value 1 field, select the input column that represents the monthly salary data.
   - Add a value field under the current value.
   - Enter the number 4 in the Value 2 field.
   - Select the division sign [/] to indicate that the expression divides the first value by the second value.
   - Click **OK**.
8. Select the type of action to apply to the data that satisfies the condition logic.
   - To return a string value, select the following action type:

     ```
     string value
     ```
9. Enter the value that the action returns. For example, enter DENY_LOAN.
10. Save the rule specification.

Adding internal data to a data set

To demonstrate that you applied a business rule to a data set, you might add a date and time stamp to each record. The date and time stamp indicate the time at which Data Quality ran a mapping to validate the business rule on the data set.

For example, you might define a business rule that states that primary key columns in a data set cannot contain null values. You define a parallel business rule that sets a monthly schedule to review the primary key column data. You configure a rule statement that analyzes the primary key columns and that adds a date and time stamp to each record to indicate the time of the analysis.

Add a condition for each primary key column. Use the AND feature to combine the conditions.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the primary key data.
5. Select an operator to validate the results of the condition analysis.
   - To specify that the condition does not apply to null data, select the following operator:

     `is not`
6. Select the type of condition to apply to the input.
   - To verify that the input values are not null, select the following condition type:

     `null value`
7. Optionally, select **Add Condition** in the rule statement menu.

   The rule specification adds a condition under the current condition and creates an AND relationship between the conditions.
8. Repeat steps four through six to configure a condition for any additional primary key column in the data set.
9. Select the type of action to apply to the data rows that meet all of the conditions in the rule statement.
   - To return the date and time at which the mapping runs, select the following action type:

     `current time stamp`
10. Optionally, update the default rule statement to return a value if any condition in the earlier rule statement finds a null value. Select `string value` as the action type, and then enter a value such as NULL.

    When you update the default rule statement in this way, the action can write an output for each row of input data.
11. Save the rule specification.

Finding a value in a range of values

To determine if any value in a list of values appears in a column of business data, add the values on the list to a condition. The rule statement compares the values in the input that you select to the values in the list.

Add a list of values to a condition in the following scenario:

- The business rule requirement that the condition represents applies to multiple values.
- The organization does not maintain a dictionary of the values.

For example, you might work for a manufacturing company that must recall a small number of products because of a product defect. You know the serial numbers of the products that you must recall. You configure a rule statement with an input that represents the column of serial number data. You add the list of serial numbers to recall to the rule statement condition.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.
5. Select an operator to validate the results of the condition analysis.
   - To identify input values that match the dictionary values, select the following operator:

     ```
     is within
     ```
6. Select the type of condition to apply to the input.
   - To enter a list of values, select the following condition type:

     ```
     function
     ```
7. Configure the condition to apply a list of values to the input data.
   - Open the **Define Condition** dialog box.
   - Select **Input**, and select **List of Values**.
   - Enter a list of values. By default, the dialog box displays a single data field. Add a data field for each value on the list.
   - Click **OK**.
8. Select the type of action to apply to the data that satisfies the condition logic.
   - To return a string value, select the following action type:

     ```
     string value
     ```
9. Enter the value that the action returns. For example, enter RECALL.
10. Save the rule specification.

Identifying data values that contain key terms or character strings

To find a word or a character string in a column of input data, configure a condition to use the "contains" operator. You might use the operator to find a single data value when each field in the input contains multiple

values. Or, you might use the operator to find input data values that contain a character string within a longer string.

For example, you might work for an organization that stores inventory items in a warehouse. You want to measure the number of items in inventory for a range of products. The product names are West Wood, West Star, and West Land. You design a condition to find product names that contain the string "West."

The rule statement that you configure might represent a single element in a business rule. For example, the business rule might describe a policy to reduce the sale price of products that spend a long time in inventory. The current rule statement generates an output that a rule statement in a parent rule set can read as an input.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.
5. Select an operator to validate the results of the condition analysis.

   When you use the `contains` operator, the rule statement prompts you to use the `function` condition type.
6. In the **Define Condition** dialog box, select Contains.
7. Enter the value to compare to the input data in the Input field. For example, enter "West."

   The function searches the rule statement input for the string value that you enter.

   By default, the function looks for the value anywhere in the input string. You can also configure the function to look for the value at the start or at the end of the input string.

   Note: You can also configure the Contains function to compare the condition input to a value on another input.
8. Select the type of action to apply to the data that satisfies the condition logic. For example, you might configure an action that returns the inventory numbers for each product.
   - To return the data values from another column on the same row, select the following action type:

     `input`
9. Select the input that represents the inventory levels for each product in the warehouse.
10. Save the rule specification.

Generating status data for a profile

To generate status values, configure an action with the *status value* option.

For example, you might work for a bank that monitors the account status of its account holders on a daily basis. You define a rule statement that returns the predefined status value *Invalid* for accounts that are overdrawn and returns the *Valid* status value for other accounts. You can add the rule specification to a profile and configure the profile to generate metrics for the status values.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.

4. Select an input for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.
5. Select an operator to validate the results of the condition analysis.
   - To evaluate whether the account balances are below zero, select the following operator:

   `is greater than or equal to`
6. Enter the value to compare to the data in the Input field. In this case, enter the digit 0.

   The condition evaluates whether the input field contains positive or negative values.
7. Select the type of action to apply to the data that satisfies the condition logic.
   - To return a status value that represents the result of the condition, select the following action type:

     `status value`
   - Select the status value to return when the input data satisfies the condition. In this case, select *Valid*.
8. Define an action to perform when no rule statement is valid.
   - To return a status value that represents the result of the condition, select the following action type:

     `status value`
   - Select the status value to return when the input data does not satisfy the earlier condition. In this case, select *Invalid*.
9. Save the rule specification.

Using status data to identify exception records

To generate status values that identify exceptions, configure an action with a status value of *Invalid* and use the exception properties to indicate why the records meet your exception criteria. Users who analyze the data in downstream processes can use the data that you add to find the exception records and to understand why they qualify as exceptions.

For example, you might decide to examine the output from a mapping that performed duplicate analysis, on the basis that the output may contain additional duplicate records. Let's say that the deduplicate asset in the mapping applied a match threshold value of 90% to the records.

You decide to identify the records that retuned a match threshold value of 89% in the deduplicate output. You define a rule statement that reads the Out_LinkScore field from the output and updates any records with a score between 89% and 90% as Invalid. Additionally, you define exception properties to add `Major` as the priority value and `Potential Duplicates` as the descriptive text.

1. Select the primary rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input of type float for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.

     Note: The input represents the Out_LinkScore value.
5. Select an operator to validate the results of the condition analysis.

- To identify input records with a score of 89% and above, select the following operator:

  ```
  is greater than or equal to
  ```

6. Select the type of condition to apply to the input.
   - To compare the input data to a float, select the following condition type:

     ```
     float value
     ```
7. Enter the value to compare to the input data. In this case, enter 0.89.
8. Select **Add Condition** in the rule statement menu.

   The rule specification adds a condition under the current condition and creates an AND relationship between the conditions.
9. Select the input that you added in step 4.
10. Select an operator to validate the results of the condition analysis.
    - To identify input records with a score of below 90%, select the following operator:

      ```
      is less than
      ```
11. Select the type of condition to apply to the input.
    - To compare the input data to a float, select the following condition type:

      ```
      float value
      ```
12. Enter the value to compare to the input data. In this case, enter 0.9.
13. Select the type of action to apply to the data that satisfies the condition logic.
    - To return a status value that represents the result of the condition, select the following action type:

      ```
      status value
      ```
    - Select the status value to return when the input data satisfies the condition. In this case, select *Invalid*.
14. Configure the exception properties to add data that describe the status of the record.
    - Click the **+ Exception** option to open the **Configure Exception Data** dialog box.
    - Select the **Add Exception Details** option.
    - Add a priority value. For example, Minor.
    - Enter a text description.
    - Click **OK**.
15. Save the rule specification.

Resolving hidden data issues

Your data may contain issues that are not immediately evident. For example, you may define a rule specification to find null values in a data column. However, the rule specification does not find values that contain character spaces exclusively. Such values can appear to be null when you view them onscreen.

To find the values, you define a rule specification with a series of rule statements. One rule statement searches for null values. Two additional rule statements search for fields that contain character spaces exclusively. To find the fields that contain character spaces only, you configure one of the rule statements to read the output from the condition of the rule statement that precedes it.

Finding null values

In this task, you define a rule statement that searches for null or empty strings. To find fields that contain null or empty strings, set the condition type to `null value` and use the `is` operator to determine if the input value is null.

1. Select a rule set in a rule specification.
2. In the rule set properties, click **Rule Logic**.
3. Click **Add Rule Statement**.
4. Select an input of type string for the rule statement.
   - If the rule specification does not contain an input that you can use, create an input. Configure the input properties to represent the type of column that contains the business data.
5. Select an operator to validate the results of the condition analysis.
   - To specify that the input data must match the data in the condition, select the following operator:

     `is`
6. Select the type of condition to apply to the input.
   - To verify that the input values are null, select the following condition type:

     `null value`
7. Select the type of action to apply to the data that satisfies the condition logic.
   - To return an integer value, select the following action type:

     `integer value`
8. Enter the value that the action returns. For example, enter 2.

   Note: Enter a value other than 1 or 0. The values 1 and 0 represent the true and false outcomes from the function that identifies character spaces in the next task.

Finding values that contain character spaces only

In this task, you define rule statements that search for fields that exclusively contain character spaces. You configure two rule statements, where one rule statement reads the result of another rule statement. To find fields that contain character spaces only, you create a rule statement that uses the `Is Spaces` function.

1. Click **Add Rule Statement**.
2. Select the input that you added in the earlier task.
3. Select an operator to validate the results of the condition analysis.
   - To specify that the condition does not apply to null data, select the following operator:

     `is not`
4. Select the type of condition to apply to the input.
   - To identify input values that are not null, select the following condition type:

     `null value`
5. Select the type of action to apply to the data that satisfies the condition logic.
   - To return an integer value, select the following action type:

     `integer value`
6. Enter the value that the action returns. For example, enter 3.

   Note: Enter a value other than 1 or 0 and other than the value that you specified in the earlier task.
7. Select **Add Rule Statement Using Result** from the actions menu on the current rule statement.

The rule specification adds an empty rule statement under the current rule statement. The rule statement reads the result of the rule statement above it.

8. In the rule statement that you added in step 7, select an operator to validate the results of the condition analysis.
   - To specify that the condition reads the output from the earlier rule statement, select the following operator:

     ```
     is
     ```
9. Select the type of condition to apply to the input.
   - To verify the output from the earlier rule statement, select the following condition type:

     ```
     integer value
     ```
10. Enter the value that you added in step 6 in the current task.

    You configure the condition to search for values that indicate that the input string is not null. For example, enter 3 if the earlier rule statement returns the value 3 when an input string is not null.
11. Select the type of action to apply to the data that satisfies the condition logic.
    - To use a function expression in an action, select the following action type:

      ```
      function
      ```
12. Configure the action to search the input data for character spaces only.
    - Open the **Define Condition** dialog box.
    - Select **Functions**, and select **Is Spaces**.
    - Select the string input that you specified in the earlier rule statements.
    - Click **OK**.
13. Save the rule specification.

The rule specification searches first for null strings and then for strings that contain character spaces only. The `Is Spaces` function returns 1 for any string that exclusively contains character spaces and 0 for any other string that it reads.

Combining data values in advanced mode

To combine two or more data values in your data set, use the || operator. The operator combines the data values that you require into a single string.

For example, your organization might combine values to generate unique identifiers or keys. You might concatenate relevant attributes in your data set, such as a customer ID and an order ID to create a composite key that ensures unique identification of a data value.

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.
   - If the rule specification does not contain an input and an output that you can use, create an input and output. Configure the input and output properties to represent the type of column that contains the business data.
3. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement.
4. To specify the input, type the input name.
5. Enter the input value. For example, enter `12345`.

6. Enter an operator value. For example, enter the `AND` value to connect the inputs.
7. To specify another input, type the input name.
8. Enter the input value. For example, enter `1111`.
9. Type the `THEN` element.
10. To specify the output, type the output name.
11. Enter an operator value. For example, enter the `=` value to write a value for the data that satisfies the rule logic.
12. Enter the input names and use the `||` operator between the inputs to combine the values in the inputs.
13. Type the `ENDIF` element.

The following example shows the use of concatenate operator:

```
IF (Input = "12345" AND Input1 = "1111")
    THEN
        Output = (Input || Input1)
ENDIF
```

Standardizing character case in advanced mode

To convert your input data to a consistent case format, write rule logic that changes the character case of the data values. For example, your organization might want to store data values in a specific case format to ensure consistency in the data values.

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.
   - If the rule specification does not contain an input and an output that you can use, create an input and output. Configure the input and output properties to represent the type of column that contains the business data.
3. In the rule editor, type the output name.
4. Enter an operator value. For example, enter the `=` value to write a value for the data that satisfies the rule logic.
5. To change the character case to sentence case, select the `convertToSentenceCase` function in the operation.

   You can select the function from the suggestion list.
6. Type the name of an input that represents the column of data that the rule specification will search at run time.

The following example shows rule logic that contains a case convert operation:

```
Output = CONVERTCASE("convertToSentenceCase",Input)
```

Eliminating redundant character spaces in advanced mode

To eliminate redundant spaces in a data set, write rule logic to replace multiple character spaces with a single space.

For example, you organization might decide to identify duplicate records in a data set. When you perform deduplication operation on your data set, the redundant spaces in the data values might lead to incorrect results. If you do not eliminate the redundant character spaces, the deduplication operation might not correctly identify the records that are identical as duplicates, which leads to duplicate entries in the data set.

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.
   - If the rule specification does not contain an input and an output that you can use, create an input and output. Configure the input and output properties to represent the type of column that contains the business data.
3. In the rule editor, type the output name.
4. Enter an operator value. For example, enter the = value to write a value for the data that satisfies the rule logic.
5. To remove redundant characters, select the `removeMultipleSpaces` function in the operation.

   You can select the function from the suggestion list.
6. Type the name of an input that represents the column of data that the rule specification will search at run time.

The following example shows rule logic that contains a standardize operation:

```
Output = STANDARDIZE("removeMultipleSpaces",Input);
```

Standardizing data values using a dictionary in advanced mode

To standardize data values, you can write rule logic that searches for instances of values that appear in a dictionary and replaces each value with a corresponding valid value from the dictionary.

For example, a bank might decide to standardize address terms in customer address data so that the data is consistent and uniform. The bank wants to use abbreviated street descriptors that the postal carrier might prefer, such as St for Street, Rd for Road, and Ave for Avenue.

Your data quality assets include a dictionary of address terms. You write a rule specification that standardizes any value in the input data that matches a value in the dictionary.

At a high level, you'll complete the following steps:

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.

   If the rule specification does not contain an input and an output that you can use, create an input and output. Configure the input and output properties to represent the type of column that contains the business data.
3. Write an IF statement to define the condition in which the rule will standardize data.

4. Write a THEN statement to define the action to take when the data satisfies the IF condition.

   Include a `replaceWithDictionary` function in the THEN statement. You can optionally add parameters to the function.

5. Add an ENDIF element.

Writing the IF statement

1. In the rule editor, type the `IF` element of the IF-THEN-ELSE statement.
2. Type an input name.
3. Add an `in` value to associate the input name with the dictionary name.
4. Type the name of the dictionary that contains the address terms.

   The rule editor can suggest available dictionaries based on the text that you type.

Writing the THEN statement

1. In the rule editor, type the `THEN` element.
2. Type an output name.
3. Add an = operator value to associate the output with the function that you'll add.
4. Define a function that returns the dictionary values that match input values.

   To define the function, use the STANDARDIZE command with a `replaceWithDictionary` function, add the input name, and specify the dictionary.

   Use the dictionary that you selected in the IF statement. The rule editor can suggest available dictionaries and function names based on the text that you type.

Writing the ENDIF statement

- To complete the rule logic, type `ENDIF`.

Adding optional parameters to the THEN statement function

- Optionally, add one or more of the following parameters to the `replaceWithDictionary` function:
  - Specify the dictionary column that contains the preferred versions of the address terms. If you do not identify a valid column, the rule specification uses the first column in the dictionary.
  - Enter the delimiter that the rule specification will use to recognize discrete values in the input field. Specify the delimiters within square brackets. If you do not specify a delimiter, the rule specification uses a space as the delimiter.
  - Enter the scope for the search operation. You can set the following options:
    - `Anywhere`. Replaces any instance of the value that occurs in the input field.
    - `Start`. Replaces any instance of the value that is not preceded by another value in the input field.
    - `End`. Replaces any instance of the value that is not followed by another value in the input field.
  - Enter `NULL` to indicate that you are not replacing input values with a custom value.
  - Enter `TRUE` to indicate that the function will perform a case-sensitive comparison between the dictionary data and the input data.

Rule logic example

The following example shows rule logic that represents the standardize operation:

```
IF Input in dictionary:MRS.Merge_Cleanse_Strategy.AddressDictionary
THEN Output =
STANDARDIZE("replaceWithDictionary",Input,dictionary:MRS.Merge_Cleanse_Strategy.AddressDi
ctionary, "Column 1",[" "],"Anywhere","NULL", TRUE)
ENDIF
```

Standardizing custom data values in advanced mode

To standardize data values, you can write rule logic that searches for instances of one or more values that you specify and replaces the values with a single new value.

For example, your organization might decide to standardize a list of department names to a common format. The organization decides to replace the terms `Information Technology` and `Information Tech` with the term `IT` in a column of department name data.

1. Open a rule specification in advanced mode.
2. Verify that the rule specification contains the inputs and outputs that you require.

   If the rule specification does not contain an input and an output that you can use, create an input and output. Configure the input and output properties to represent the type of column that contains the business data.
3. In the rule editor, type an output name.
4. Add an = operator value to associate the output with the function that you'll add.
5. Define a function that replaces the string values that you specify with a new value.

   To define the function, use the STANDARDIZE command with a `replaceCustomString` function, add the input name, and enter the string values that the rule specification will search for and replace.

   Specify the list of custom string values within square brackets.

   The rule editor can suggest available function names based on the text that you type.
6. Enter the delimiter that the rule specification will use to recognize discrete values in the input field.

   Specify the delimiters within square brackets. If you do not specify a delimiter, the rule specification uses a character space as the delimiter.
7. Enter the scope for the search operation.

   You can set the following options:

   - `Anywhere`. Replaces any instance of the value that occurs in the input field.
   - `Start`. Replaces any instance of the value that is not preceded by another value in the input field.
   - `End`. Replaces any instance of the value that is not followed by another value in the input field.
8. Enter the replacement value.

The following example shows rule logic that represents the standardize operation:

```
Output = STANDARDIZE("replaceCustomString",Input,["Information Technology","Information
Tech"],[","," "],"Anywhere","IT")
```

CHAPTER 7

Validation and testing

Validate a rule specification to verify that the rule specification is valid for use. For example, validate a rule specification before you add it to a Rule Specification transformation in Data Integration.

Test the rule specification to verify that the output satisfies the business rule requirements. To test a rule specification or a rule set, enter sample values in one or more data columns on the Test tab.

You can cut, copy, and paste test data within the test area and between rule sets in the rule specification. If you open more than one rule specification in concurrent browser tabs, you can cut, copy, and paste test data between the rule specifications.

You can test a complete rule specification, and you can test each rule set in the rule specification. If a rule specification is not valid, test the rule sets to find and fix the validation error.

Rules and guidelines for test and validation operations

Before you add a rule specification to a Rule Specification transformation, validate the rule specification.

Consider the following rules and guidelines when you validate a rule specification:

- A rule specification can fail validation for the following reasons:
  - A condition or an action in a rule statement specifies a data type that differs from the input data type.
  - The rule statements in a rule set generate output data values of different data types.
  - A parent rule set does not read an output from a child rule set.
  - A rule statement is incomplete.
- When you move or copy a rule set to another rule specification, you move or copy any test data that you added for the rule set. You also move or copy any input that the rule set uses.

Validating a rule specification

Validate a rule specification to verify that the rule specification is ready for use in a Rule Specification transformation.

1. Open the rule specification.

2. Click **Validate** on the asset toolbar.

   The rule specification highlights any rule set that is not valid. Click a rule set to read a message that describes the validation error.

Testing a rule specification in basic mode

Test a rule specification to verify that the data flows through the rule specification in the ways that you expect. If possible, use the business data to test the rule specification.

1. Open the rule specification.
2. Select the primary rule set.
3. In the rule specification properties, click **Test**.

   The properties view displays the input columns that you configured in the rule specification.
4. Select a Secure Agent to run the test.

   To refresh the list of active Secure Agents, click the **Refresh** icon.
5. Enter one or more data values in an input column. You can enter data values for any rule set in the rule specification.

   If you enter date data in an input column, use one of the following date formats:

   - yyyy-MM-dd
   - yyyy-MM-dd hh:mm:ss
   - yyyy-MM-dd hh:mm:ss.SSS

   Note: To test a rule statement that reads a list of date values as an input, enter the date data in the format yyyy-MM-dd.
6. Click **Run Test**.

   The Outputs view displays the path of the data values through the rule specification.

   If your rule statement contains the status value *Invalid* and you configured the action to describe exception data, the test results include columns for the exception data options that you specified.
7. Verify that the rule sets read the input values and write the output values that you expect.

   - A data value from a child rule set might generate output values on multiple rule sets. Verify that the test results indicate a data value on each rule set.
   - If a data value does not generate an output on a rule set, the test results show an empty field for the rule set.

Testing a rule set

Test a rule set to verify that the rule statements generate the correct output values. If possible, use the business data to test the rule specification.

To verify the full operation of the rule specification, select the top-level rule shape and test the rule specification.

1. Open the rule specification.
2. Select a rule set.
3. In the rule set properties, click **Test**.

   The properties view displays the input columns that you configured in the rule set.
4. Select a Secure Agent to run the test.

   To refresh the list of active Secure Agents, click the **Refresh** icon.
5. Enter one or more data values in an input column. You can enter data values for any rule set in the rule specification.

   If you enter date data in an input column, use one of the following date formats:

   - yyyy-MM-dd
   - yyyy-MM-dd hh:mm:ss
   - yyyy-MM-dd hh:mm:ss.SSS

   Note: To test a rule statement that reads a list of date values as an input, enter the date data in the format yyyy-MM-dd.
6. Click **Run Test**.

   The Outputs view displays the path of the data values through the rule specification.

Testing a rule specification in advanced mode

Test a rule specification asset to verify that the data flows through the asset in the ways that you expect.

1. Open the rule specification that you created in advanced mode.
2. Select the **Configuration** tab.
3. Select a Secure Agent to run the test.

   To refresh the list of active Secure Agents, click the **Refresh** icon.
4. Enter one or more data values in the test panel.
5. Click **Test**.

   The test runs the rule logic in the asset. The output columns display the test results.

   If you configure exception indicators in the expression logic, the test results include columns for the exception data options that you specified.
6. Verify that the rule logic reads the input values and writes the output values that you expect.

INDEX

B

business rules
  facts 25

C

Cloud Application Integration community
  URL 6
Cloud Developer community
  URL 6
conditions
  Define Action dialog box 38
  Define Condition dialog box 33
copy and paste operations
  rule sets 29
  rule statements 51
creating an input 19

D

Data Integration community
  URL 6

F

facts in business data 25
function expressions
  Add to Date 45
  Choose 45
  Concatenate 45
  Convert to Date 45
  Convert to Float 45
  Convert to Integer 45
  Date Difference 45
  Date Part 45
  Date Time 45
  Greatest 45
  Is Spaces 45
  Last Day 45
  Least 45
  Length 45
  Lower 45
  Null 45
  Replace Character 45
  Replace String 45
  Reverse 45
  Substring 45
  To Char 45
  To Date 45
  Truncate 45
  Upper 45

I

Informatica Global Customer Support
  contact information 7
Informatica Intelligent Cloud Services
  web site 6
Input Management dialog box 19
inputs
  data types 11

M

maintenance outages 7

R

rule sets
  adding a rule set to a rule specification 28
  configuration 25
  configuration rules and guidelines 25
  copy and paste operations 29
  inputs 11
  properties 28
  testing a rule set 80
rule specifications
  and dimensions 13
  and mappings 13
  and rule occurrences 13
  and scorecards 13
  configuration prerequisites 15
  creating a rule specification 16
  dimensions 15
  inputs 11
  output properties 20
  rule sets 10, 25
  rule statements 12, 31
  status values 12
  testing a rule specification 79
  verify business data properties 16
  verify business rule requirements 15
rule statements
  actions 12, 36
  add rule statement using result 49
  AND relationships 65
  conditions 12, 32
  configuration 31
  copy and paste operations 51
  current date analysis 32
  default rule statement 12
  discovering business information 65
  function expression descriptions 45
  improving data usability 64
  List of Values function 27
  menu options 47

rule statements *(continued)*
nested mapplets 33
null value analysis 32
operators 12, 35
searching a list of values 68
standardizing data 63
steps to configure 48
verify data against business regulations 66
verifying data accuracy 61

S

status
Informatica Intelligent Cloud Services 7
system status 7

T

trust site
description 7

U

upgrade notifications 7

W

web site 6