# Contents

**Chapter 1 / Introduction to SAS Intelligent Decisioning** ........................................... 1  
- Enterprise Decision Management Systems ................................................................. 1  
- About Business Rules ..................................................................................................... 2  
- SAS Intelligent Decisioning Features ........................................................................... 2  
- Support for Models in Decisions ................................................................................... 3  
- Workflow for Creating and Publishing Decisions ...................................................... 3  
- Sign in to SAS Intelligent Decisioning .......................................................................... 4  

**Chapter 2 / Working with Business Rules** ................................................................. 5  
- About Rules, Assignment Rule Sets, and Filtering Rule Sets ........................................ 6  
- About Condition and Action Expressions ..................................................................... 7  
- Create a New Rule Set ................................................................................................... 8  
- Managing the Variables in a Rule Set ........................................................................... 9  
- Defining New Rules in a Rule Set ................................................................................ 13  
- Defining Expressions in Rules and Assignment Statements ......................................... 18  
- Managing Rules ........................................................................................................ 26  
- Copy a Rule Set URL .................................................................................................. 28  
- Generate PDF Documentation for a Rule Set ............................................................ 28  
- Managing Rule Sets .................................................................................................... 29  
- Managing Versions of Rule Sets ................................................................................ 30  
- Testing a Rule Set ....................................................................................................... 32  
- Publishing and Validating a Rule Set .......................................................................... 38  
- Executing Published Content ..................................................................................... 42  
- Importing and Exporting Rule Sets ........................................................................... 44  

**Chapter 3 / Working with Treatments and Treatment Groups** .................................... 47  
- About Treatments and Decisions ................................................................................ 48  
- Example: A Decision That Includes a Treatment Group ............................................. 49  
- About Attributes, Eligibility Rules, and Effective Dates ............................................ 51  
- Arbitrating Treatments ............................................................................................... 52  
- About Channels ......................................................................................................... 52  
- Define a Treatment ..................................................................................................... 53  
- Define a Treatment Group ......................................................................................... 54  
- Activate a Treatment Group ....................................................................................... 55  
- Managing Attributes and Aliases .............................................................................. 56  
- Generate PDF Documentation for a Treatment Group ............................................. 58  
- Delete Treatments from a Treatment Group .............................................................. 59  
- Managing Eligibility Rules for a Treatment .............................................................. 59  
- Managing Treatments and Treatment Groups ............................................................ 60  
- Managing Versions of Treatments and Treatment Groups ......................................... 62  

**Chapter 4 / Using Lookup Tables and Functions** ..................................................... 65  
- About Lookup Tables and Functions ......................................................................... 66  
- Predefined Lookup Tables ......................................................................................... 67  
- Create a New Lookup Table ...................................................................................... 67  
- Importing and Exporting Lookup Tables .................................................................... 68  
- Add Lookup Table Entries ....................................................................................... 69  
- Edit Lookup Table Entries ....................................................................................... 69
Enterprise Decision Management Systems

Enterprise decision management systems can transform how businesses make decisions. They enable businesses to use the information they already have to make better decisions—decisions that are based on predictive analytics rather than on past history. Decision management systems automate the process of making decisions, particularly day-to-day operational decisions. They improve the speed, efficiency, and accuracy of routine business processes, in part by reducing the need for human intervention. Automating decisions with SAS Intelligent Decisioning provides a streamlined mechanism for controlling and monitoring the rules and processes used by your organization. By automating decisions, organizations in every industry can improve interactions with customers, partners, suppliers, and employees. In addition, organizations that are highly regulated, such as financial services, health care, and insurance, can more easily achieve compliance as a result of repeatable, traceable decisions. Explicitly defining a decision makes your organization’s decision-making process transparent, and enables you to monitor the process for accuracy.
SAS Intelligent Decisioning helps organizations manage data, business rules, analytical models, and optimization techniques. Rule management, model management, and data management are integrated into a consistent interface for easier accessibility.

About Business Rules

Business rules capture the logic of business decisions and are a core component of decision management systems. Business rules enable you to codify the decision-making process used by your organization. Business rules make the decision-making process transparent and adaptable, enabling organizations to respond quickly to new information about customers and markets. They enable organizations to identify and deal with fraud, avoid unnecessary risk, and find opportunities hidden in customer data.

SAS Intelligent Decisioning Features

You can use SAS Intelligent Decisioning to create a database of business rules, combine those rules together into decisions, and publish the decisions for use by other applications. SAS Intelligent Decisioning provides the following capabilities:

business rule authoring
A business rule specifies conditions to be evaluated and action to be taken if those conditions are satisfied. For example, you can create a rule that determines whether a particular customer has a mortgage. That same rule can then add the outstanding balance of the mortgage to a running total of the customer’s debt. With SAS Intelligent Decisioning, you define the conditions and actions for each rule.

rule set management and publishing
A rule set is a logical collection of rules. A single rule set can have many rules, but it generally corresponds to a single step in a decision. For example, you can have a rule set that determines a customer’s asset balance and another rule set that determines a customer’s debt level. You can use SAS Intelligent Decisioning to easily create new rule sets, reorder the rules in a rule set, add new rules to existing rule sets, and so on. When a rule set is published, the versioning features of SAS Intelligent Decisioning create a static version of the rule set. This static version helps you enforce integrity and governance over the rules that are put into production.

treatment authoring and management
A treatment is an offer that can be sent to a customer as part of an inbound marketing campaign. For example, when a customer visits your company’s website, a customer service application can capture information about the customer. The application can invoke a decision that includes a set of treatments and pass the customer’s information to the decision. The decision determines which of the treatments (offers) are suitable to present to that customer at that
time. With SAS Intelligent Decisioning, you can author treatments, combine them into treatment groups, and add the groups to decisions.

lookup table authoring and management
Lookup tables are tables of key-value pairs. You can use lookup tables for tasks such as retrieving a part name based on a part number or retrieving a location name based on a ZIP code. SAS Intelligent Decisioning provides two predefined lookup tables for use with treatments: one for treatment channels and one for subject levels.

code file authoring and management
You can define custom code files to do things that are not possible in rules, models, or treatments. For example, you can define a code file that makes HTTP calls to REST APIs, interacts with a database, manipulates files in the file system, or performs custom data transformations. In SAS Intelligent Decisioning, you can easily create, edit, and manage custom data query files, DS2 files, and Python files. You can create and edit data query files by using the SQL editor or by launching SAS Studio.

decision authoring and publishing
SAS Intelligent Decisioning enables you to combine rule sets, analytical models, treatment groups, custom code files, record contacts nodes, and branches (conditional logic) into decisions. You can also add a decision to another decision. You can investigate various scenarios, test and refine the decision logic, and then publish the decisions for use in batch applications and online transactions. After a decision has been published, it is available for use by other applications.

Support for Models in Decisions
Support for a model depends on the tool that you use to create the model and the model’s score code type. For example, decisions that use certain model score code types cannot be published or can be published only to SAS Micro Analytic Service destinations. For more information, see “High-Level Support Matrix for Primary Functions” in SAS Model Manager: User’s Guide.

Workflow for Creating and Publishing Decisions
The following list shows the primary steps for creating a decision flow in SAS Intelligent Decisioning. Technically, all of these steps are optional. The actual steps that are required depend on the content of the decision and how your organization uses rule sets and decisions.

1 Create rule sets.
2 Test rule sets.
3 Publish rule sets.
4 Validate published rule sets.
5 Create treatments and treatment groups.
6 Create decisions.
7 Test decisions.
8 Publish decisions.
9 Validate published decisions.

After you publish a rule set or decision, it is available for use by other applications. In a production environment, these applications map variables in the rule set or decision to columns in the input data. The output that is generated when a decision is executed is written to an output table. The location of the input and output data is specified by the application.

---

Sign in to SAS Intelligent Decisioning

**Note:** If you are already signed in to SAS Drive, you can access SAS Intelligent Decisioning by clicking ✖️ and selecting **Build Decisions**.

To sign in to SAS Intelligent Decisioning:

1 In the address bar of your web browser, enter the URL for SAS Intelligent Decisioning and press **Enter**. The Sign In page appears.

   **Note:** Contact your system administrator if you need the URL for SAS Intelligent Decisioning. The default URL is http://host_name/SASDecisionManager.

2 Enter a user ID and password.

3 Click **Sign In**.

4 (Optional) If you have not previously signed in to SAS Intelligent Decisioning, SAS Intelligent Decisioning displays the Welcome to SAS window in which you can set up a profile. You can choose to enter a profile picture or select a theme. You can also set these properties in the Settings window. For more information, see “Settings” under General Usage in the Help Center.
Working with Business Rules

About Rules, Assignment Rule Sets, and Filtering Rule Sets ........................................ 6
About Condition and Action Expressions ................................................................. 7
Create a New Rule Set ............................................................................................... 8
Managing the Variables in a Rule Set ....................................................................... 9
About Variables ......................................................................................................... 9
Add Variables from a Data Table ............................................................................. 10
Add Variables from a Rule Set or Decision .............................................................. 11
Create a Custom Variable Dynamically ................................................................. 11
Create Custom Variables on the Variables Tab ......................................................... 12
Duplicate a Variable ............................................................................................... 12
Delete Variables ...................................................................................................... 13
Edit Variable Properties .......................................................................................... 13
Edit Metadata for Data Grid Variables .................................................................. 13
Defining New Rules in a Rule Set ............................................................................ 13
Add a Stand-Alone Assignment Statement ............................................................ 13
Add a New Rule in an Assignment Rule Set ............................................................. 14
Add a New Rule in a Filtering Rule Set .................................................................... 15
Copy a Rule From Another Rule Set ...................................................................... 16
Controlling Which Conditions Are Evaluated ....................................................... 17
How Rules Are Evaluated and When Rule-Fired Records Are Generated .......... 18
Defining Expressions in Rules and Assignment Statements ............................... 18
About Defining Expressions ................................................................................... 18
Using the Expression Editor ................................................................................... 19
Enter LOOKUP and LOOKUPVALUE Expressions .................................................. 20
Entering Literal Data Values .................................................................................... 21
Operators for Use in Expressions .......................................................................... 22
Using the LIKE Operator ....................................................................................... 24
Using Functions in Expressions .............................................................................. 25
Working with Missing Values .................................................................................. 26
Delete Expressions or ELSE Rules ........................................................................ 26
Managing Rules ....................................................................................................... 26
Duplicate a Rule ....................................................................................................... 26
Delete a Rule ............................................................................................................ 26
Rename a Rule ........................................................................................................... 27
About Rules, Assignment Rule Sets, and Filtering Rule Sets

A rule specifies conditions to be evaluated, and it can also specify actions to be taken if those conditions are satisfied. Rules are grouped together into rule sets. Rule sets are logical collections of rules that are grouped together because of interactions or dependencies between the rules or because they are processed together after they are published.

A rule set can be a filtering rule set or an assignment rule set.

filtering rule sets

contain rules that correspond to the form:

IF condition_expressions
These rules contain only IF statements. They have conditional expressions but do not have action expressions. Filtering rule sets enable you to select only certain records for processing. Only the records for which the conditions evaluate to True are processed by the remaining objects in the decision.

Filtering rule sets are also used as the eligibility rule set in treatments. In a treatment, the eligibility rule set defines who is eligible to receive the offer that is defined in the treatment.

Assignment rule sets contain rules that correspond to one of the following forms:

- **Assign** variable variable_or_value
- IF condition_expressions THEN action_expressions

These rules are either assignment statements or IF-THEN statements, or IF-THEN-ELSE statements. IF-THEN and IF-THEN-ELSE statements have both conditional expressions and action expressions.

Assignment statements contain only an action expression. Assignment statements always execute unless a RETURN action stops the execution of the rule set before execution reaches the assignment statement. See Step 8 of “Add a New Rule in an Assignment Rule Set” on page 14 for information about the RETURN action.

---

About Condition and Action Expressions

An assignment rule corresponds to this form:

IF condition_expressions THEN action_expressions

For example, suppose you have the following rule:

IF customer_debt > customer_assets THEN approval_status = 'Decline'

In this case, customer_debt > customer_assets is a condition expression, and approval_status = 'Decline' is an action expression.

For example, the following figure shows the rule above as it appears in the rule set editor:

![Rule Editor](image)

Note: Filtering rules do not contain action expressions.

A single assignment rule can contain multiple condition expressions and action expressions. Multiple condition expressions within the same rule are joined together with the AND operand. For example, suppose you define the following rule in SAS Intelligent Decisioning:
Create a New Rule Set

1. Click `_Assign_` to navigate to the Rule Sets category view.

2. Click **New Rule Set**. The New Rule Set window appears.

3. Enter a name for the rule set if you do not want to use the default name. Rule set names are limited to 100 characters and must be unique within a folder.

4. Select the rule set type. See “About Rules, Assignment Rule Sets, and Filtering Rule Sets” on page 6 for descriptions of each rule set type.

   **IMPORTANT** You can publish decisions that include filtering rule sets to SAS Micro Analytic Service destinations, but you cannot publish the filtering rule sets themselves to SAS Micro Analytic Service destinations.

5. (Optional) Enter a description for the new rule set. Descriptions are limited to 1000 characters.

   **TIP** You can edit the description at any time on the **Properties** tab.

6. Click `Folder`, and select the folder where you want to save the rule set.

7. Click **Save**. SAS Intelligent Decisioning opens the new rule set and displays the **Variables** tab.

8. Add variables and rules to the rule set. For more information, see the following topics:

   - “Managing the Variables in a Rule Set” on page 9
   - "Add a Stand-Alone Assignment Statement" on page 13
   - "Defining New Rules in a Rule Set" on page 13
Managing the Variables in a Rule Set

About Variables

The Properties of a Variable

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Variable names must start with a letter or an underscore (_), and they can contain only alphanumeric characters and the underscore. They can be up to 32 characters long and must be unique within a rule set.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> SAS Intelligent Decisioning does not support double-byte character set (DBCS) characters in variable names.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Do not use any of these operators or keywords as variable names: AND, OR, IN, NOT, LIKE, TRUE, or FALSE. Do not use <em>N</em> or any DS2 reserved word as a variable name. See “Reserved Words in the DS2 Language” in SAS DS2 Programmer’s Guide for information about reserved words in the DS2 language.</td>
</tr>
<tr>
<td>Data type</td>
<td>SAS Intelligent Decisioning supports the following data types: Boolean, character, data grid, date, datetime, decimal, and integer.</td>
</tr>
<tr>
<td></td>
<td>For Boolean values, enter True or 1 and False or 0. If you specify True or False, SAS Intelligent Decisioning represents Boolean values using the numbers 1 and 0 in the code that it generates.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For content that will be published to a SAS Micro Analytic Service destination, the values of date and datetime input variables must be numeric. They cannot be formatted values such as 18Jul2019.</td>
</tr>
<tr>
<td>Input and Output</td>
<td>A variable can be an input variable, an output variable, both, or neither (a temporary variable). See “Input Variables, Output Variables, and Temporary Variables” on page 10 for more information.</td>
</tr>
<tr>
<td>Length</td>
<td>For character variables that are input-only variables or that are both input and output variables, the length is derived from the length of the column to which the variable is mapped when the rule set is run.</td>
</tr>
<tr>
<td></td>
<td>The maximum length for character variables is 32767.</td>
</tr>
<tr>
<td></td>
<td>The length for Boolean and numeric variable types is set automatically.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Initial value</td>
<td>You can specify an initial value for all data types except data grids.</td>
</tr>
<tr>
<td></td>
<td>Initial values are used only at run time and only for output-only variables.</td>
</tr>
<tr>
<td>Description</td>
<td>Descriptions are limited to 256 characters.</td>
</tr>
</tbody>
</table>

Input Variables, Output Variables, and Temporary Variables

For each variable used in a rule set, you must specify whether the variable is an input variable, an output variable, both an input and an output variable, or a temporary variable.

- **Input variables** are variables that are present in the input table for a rule set. When a rule set is deployed in a production system, all input variables must be mapped to table columns in input data. When you test a rule set in SAS Intelligent Decisioning, for each input variable, you must either map it to a table column or specify a constant as its input value. When you create or edit a variable, clear the **Input** check box for any variable that you do not want to be mapped to a column in an input table or for which you do not want to specify a value.

- **Output variables** are variables that are written to the output table that is created when a rule set is run. When you create or edit a variable, clear the **Output** check box for any variable that you want to exclude from the output data.

- **Temporary variables** are variables that are not present in the input data and they are not written to the output table. To create a temporary variable for use only while a rule set is executing, clear both the **Input** and **Output** check boxes.

When you create a new variable, it is created as both an input and output variable by default.

Add Variables from a Data Table

1. On the **Variables** tab, click **Add variable** and select **Data table**. The Choose Data window appears, and the list of SAS Cloud Analytic Services (CAS) tables that are loaded into memory is displayed on the **Available** tab.

   If the table that you need does not appear in the list of available tables, try the following solutions:

   - If the table appears on the **Data Sources** tab, right-click on the table, and select **Load** to load the table into memory. If the table does not appear on the **Available** tab, click 🔄.

   - If the table does not appear on the **Data Sources** tab, import the data. The process of importing the data loads it into memory. For information about
importing data from different sources, see "Making Data Available to CAS" in SAS Data Explorer: User's Guide.

2 Select the table from which you want to import variables, and click OK. The Add Variables window appears.

3 Select the variables that you want to import and click ➡. To import all of the variables in the table, click ➩.

4 Click Add to add the select variables, or click Add and replace to replace existing variables that have the same name.

5 On the Variables tab, select or clear the Input and Output check boxes as necessary. See “Input Variables, Output Variables, and Temporary Variables” on page 10 for more information.

---

### Add Variables from a Rule Set or Decision

1 On the Variables tab, click Add variable, and select Rule set or Decision. The Choose an Item window appears.

2 Select the rule set or decision from which you want to import variables, and click OK. The Add Variables window appears.

3 Select the variables that you want to import and click ➡. To import all of the variables in the table, click ➩.

4 Click Add to add the select variables, or click Add and replace to replace existing variables that have the same name.

5 On the Variables tab, select or clear the Input and Output check boxes as necessary. See “Input Variables, Output Variables, and Temporary Variables” on page 10 for more information.

---

### Create a Custom Variable Dynamically

To create a variable in a rule set, you can enter the new variable name in any condition variable field and in the variable field of assignment statements. By default, SAS Intelligent Decisioning creates a variable of type Decimal. To create a variable of a different type, enter the variable name, a space, and then the data type. If you enter the name in a condition variable field, SAS Intelligent Decisioning creates the variable as an input-output variable. If you enter the name in an assignment statement, SAS Intelligent Decisioning creates the variable as an output variable. For example, you can create an input-output variable of type Character that is named address by entering it in the condition variable field:

You can create an output-only variable of type Boolean named approve by entering it in an assignment statement, followed by a space and the data type:
Create Custom Variables on the Variables Tab

Note: For information about data grid variables, see “Defining Data Grid Variables” in SAS Intelligent Decisioning: Using Data Grids.

To create custom variables on the Variables tab:

1. Click Add variable and select Custom variable. The Add Variables window appears.

2. Complete these steps for each variable that you want to add:
   a. Enter the name of the new variable, and select the data type of the variable. See “The Properties of a Variable” on page 9 for additional information.
   b. (Optional) Click Optional to display the Description, Initial value, and Length fields.
   c. (Optional) Enter a length, initial value, and description for the new variable. See “The Properties of a Variable” on page 9 for additional information.
   d. Click Add. SAS Intelligent Decisioning adds the new variable to the table of variables. By default, variables are added to the table as both input and output variables.
   e. (Optional) Clear the check boxes in the Input or Output columns.
      ■ Clear the Input check box for any variable that you do not want to be mapped to a column in an input table or for which you do not want to specify a value.
      ■ Clear the Output check box for any variable that you want to exclude from the output data.
      ■ Clear both the Input and Output check boxes to create a temporary variable.

3. Click OK to add the variables and close the Add Variables window.

Duplicate a Variable

1. On the Variables tab, select the variable that you want to duplicate, click ↓, and select Duplicate. The Duplicate Variable window appears.

2. Enter a new name for the duplicate variable.

3. (Optional) Enter a description for the variable.
1 Click **Duplicate**.

### Delete Variables

On the **Variables** tab, select the check box for the variables that you want to delete, click , and select **Delete**.

**Note:** You cannot delete a variable if it is used in a rule set or decision.

### Edit Variable Properties

On the **Variables** tab, click on the variable name of the variable that you want to edit. The **Edit Variable** window appears. Edit the properties as needed, and then click **OK**. See “The Properties of a Variable” on page 9 for additional information.

### Edit Metadata for Data Grid Variables

For information, see “Editing Data Grid Variable Metadata” in SAS Intelligent Decisioning: Using Data Grids.

### Defining New Rules in a Rule Set

#### Add a Stand-Alone Assignment Statement

Stand-alone assignment statements always execute unless a RETURN action stops the execution of the rule set before execution reaches the assignment statement. Rule-fired data is not generated for standalone assignment statements.

1 On the **Rule Set** tab, click **Add Assignment** if the rule set is empty or, if the rule set contains at least one statement, select **Add ➔ Add assignment**. The application adds an assignment statement to the top of the rule set, below any existing assignment statements.

2 Import or create any variables that are required for the assignment statement that have not already been added to the rule set. You can add or create the variables on the **Variables** tab, or you can define variables dynamically as you
Author the statement. See “Managing the Variables in a Rule Set” on page 9 for more information.

3 Select the variable to which you want to assign a value.

4 Enter the expression for the variable in the expression field. See “About Defining Expressions” on page 18 for additional information.

5 (Optional) Move the assignment statement to a different position in the rule set. To move the statement, click ↑ or ↓.

6 Click ✅ to save the rule set. SAS Intelligent Decisioning validates the syntax of the expressions. If it does not detect any problems, it saves the rule set.

Add a New Rule in an Assignment Rule Set

1 Create or open the rule set. If no variables are defined in the rule set, SAS Intelligent Decisioning displays the Variables tab. Otherwise, it displays the Rule Set tab.

2 Import or create any variables that are required for the new rule that have not already been added to the rule set. You can add or create the variables on the Variables tab, or you can define variables dynamically as you author the rule. See “Managing the Variables in a Rule Set” on page 9 for more information.

3 Click the Rule Set tab.

4 Click Add Rule if the rule set is empty. If the rule set contains at least one statement, click ➕Add Rule or select Add ➔ Add rule. SAS Intelligent Decisioning adds a new IF-THEN rule to the rule set.

5 (Optional) Define the condition expression for the rule. See “About Defining Expressions” on page 18 for additional information.

   To add additional condition expressions to the selected rule, select Add ➔ Condition.

6 Define the action expressions for the rule. See “About Defining Expressions” on page 18 for additional information.

   To add additional action expressions to the selected rule, select Add ➔ Action.

   **TIP** To move condition or action expressions up or down within an IF or ELSE clause, select the expression and click ↑ or ↓.

7 (Optional) Change the rule operator to ELSE. If the rule is the first rule in a rule set, the rule operator must be IF.

   When you change the operator on a rule from IF to ELSE, the condition expression is preserved, and the rule becomes an ELSE clause with an IF condition. For more information, see “Controlling Which Conditions Are Evaluated” on page 17.

8 (Optional) Change the operator on the THEN clause from ASSIGN to RETURN.

   The RETURN action stops the execution of any additional statements in the rule
Defining New Rules in a Rule Set

set. See “Controlling Which Conditions Are Evaluated” on page 17 for more information.

9 (Optional) Select Add ⇒ ELSE rule to add an ELSE clause to the currently selected rule. The ELSE clause does not have a condition, but you can add one by selecting Add ⇒ Condition.

10 (Optional) Define the condition and action expressions for the ELSE clause.

11 (Optional) Change the order of the rules. Rules are evaluated sequentially. To move a rule up or down within a rule set, select the rule and click ↑ or ↓.

12 (Optional) Change the name of the rule. Rule names are limited to 100 characters and must be unique within a rule set. For instructions, see “Rename a Rule” on page 27.

TIP Assigning logical names to the rules makes it easier to determine which rules fired when you review rule-fired data.

13 (Optional) Clear the Record rule-fired data check box if you do not want a rule-fired record to be written each time this rule fires. See “How Rules Are Evaluated and When Rule-Fired Records Are Generated” on page 18 for more information.

14 Click to save the rule set. SAS Intelligent Decisioning validates the syntax of the expressions. If it does not detect any problems, it saves the rule set.

Add a New Rule in a Filtering Rule Set

Note: When SAS Intelligent Decisioning generates code for a filtering rule set, it joins the rules together with the OR operand.

1 Create or open the rule set. If no variables are defined in the rule set, SAS Intelligent Decisioning displays the Variables tab. Otherwise, it displays the Rule Set tab.

2 Import or create any variables that are required for the new rule that have not already been added to the rule set. You can add or create the variables on the Variables tab, or you can define variables dynamically as you author the rule. See “Managing the Variables in a Rule Set” on page 9 for more information.

3 Click the Rule Set tab.

4 Click Add Rule if the rule set is empty or, if the rule set contains at least one statement, select Add ⇒ Add rule or +Add Rule. SAS Intelligent Decisioning adds a new IF rule to the end of the rule set.

5 Define the condition expression for the rule. See “About Defining Expressions” on page 18 for additional information.

To add additional condition expressions to the selected rule, select Add ⇒ Condition.
6 (Optional) Change the order of the rules. Rules are evaluated sequentially. To move a rule up or down within a rule set, select the rule and click ↑ or ↓.

7 (Optional) Change the name of the rule. Rule names are limited to 100 characters and must be unique within a rule set. For instructions, see “Rename a Rule” on page 27.

8 (Optional) Clear the Record rule-fired data check box if you do not want a rule-fired record to be written each time this rule fires. See “How Rules Are Evaluated and When Rule-Fired Records Are Generated” on page 18 for more information.

9 Click to save the rule set. SAS Intelligent Decisioning validates the syntax of the expressions. If it does not detect any problems, it saves the rule set.

Copy a Rule From Another Rule Set

The menu option Copy from rule set appears in the Actions menu ( ) on individual rules and in the Add menu for the rule set. The rule set must contain at least one rule in order for these menu options to appear. Adding a rule from another rule set does not replace any of the rules in the rule set that you are editing.

1 Open the rule set into which you want to add the rule, and click the Rule Set tab.

2 (Optional) If the rule set does not already contain a rule, click Add rule.

3 Click Add and select Copy from rule set. The Choose a Rule Set window appears.

4 Select the rule set from which you want to copy a rule, and click OK. The Choose a Rule window appears.

5 Select the version of the rule set from which you want to copy the rule.

6 Select the name of the rule that you want to copy, and click OK.

Note: If you want to copy both an IF rule and its ELSE rule, you must copy the IF rule first.
Controlling Which Conditions Are Evaluated

By default, rules are assigned the IF rule operator, which means that the rule’s conditions are evaluated regardless of the results of previous rules. In assignment rule sets, you can control whether condition expressions are evaluated by using the RETURN action and the ELSE operator.

The RETURN action stops the execution of any remaining rules in a rule set. If you are executing a single rule set, execution ends. If you are executing a decision, control moves to the next object in the decision. For example, the rule in the following figure stops the execution of any remaining rules in the rule set if the value of the Order_Quantity variable is missing.

If you set a clause’s operator to ELSE, then the clause’s conditions are evaluated only if the previous clause’s conditions evaluated to false. For example, given the rule set shown in the following figure, if Order_Quantity is 9, the condition for the IF clause evaluates to false, and the condition for the first ELSE clause evaluates to true. Therefore, the action for first ELSE clause is executed, and the condition for the last ELSE clause is not evaluated. The value of Offer_Percent is set to 5.
How Rules Are Evaluated and When Rule-Fired Records Are Generated

By default, the condition expressions for all rules in a rule set are evaluated sequentially regardless of the results of previous rules. However, in assignment rule sets, you can use the ELSE operator and the RETURN action to control whether condition expressions are evaluated. See “Controlling Which Conditions Are Evaluated” on page 17 for more information.

If a rule’s condition expressions evaluate to True, SAS Intelligent Decisioning, the rule is said to have fired. In assignment rule sets, SAS Intelligent Decisioning executes the rule’s action expressions.

By default, every time a rule fires, it generates a rule-fired record. You can control when rule-fired records are generated by using the Record rule-fired data check boxes. See Step 13 in “Add a New Rule in an Assignment Rule Set” on page 14.

Note: Stand-alone assignment statements always execute unless a RETURN action stops the execution of the rule set before execution reaches the assignment statement. Rule-fired data is not generated for standalone assignment statements.

Defining Expressions in Rules and Assignment Statements

About Defining Expressions

Expressions can be up to 1024 characters long. They can contain numeric constants, character strings, variables, operators, SAS DS2 functions, data grid functions, and the SAS Intelligent Decisioning LOOKUP and LOOKUPVALUE functions. You can enter expressions directly into the expression fields, or you can use the Expression Editor to create and edit expressions.

TIP Use caution when you test for equality by using scientific notation. Two numbers that appear to be the same might evaluate to different numbers because of the precision involved in scientific notation.

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

- “Using the Expression Editor” on page 19
- “Enter LOOKUP and LOOKUPVALUE Expressions” on page 20
Using the Expression Editor

You can use the Expression Editor to enter expressions that do not use the LOOKUP or LOOKUPVALUE functions. You must use the Expression Editor to enter expressions that use the OR operator, the concatenation (||) operator, or the exponent operator (**).

To open the Expression Editor, select an expression, and click .

You can enter expressions directly into the expression field, or you can use the lists of operators, function names, and variable names to add them to the expression.

- To add an operator, click the operator in the rows above the expression field.
- To add a variable, click the Variables tab, and double-click the variable name.
- To add a function call, click the Functions tab, expand the appropriate function category, select a function name, and click . The Expression Editor adds a basic syntax template for the function to the expression field.

You can click Validate at any time to check the syntax of the expression that you are building. Click Clear to clear the expression field.

When you are finished building the expression, click Save. The Expression Editor adds the expression to the rule set.

**IMPORTANT** After you save an expression in the Expression Editor, the expression field on the Rule Set tab becomes a read-only field. You can edit it only by launching the Expression Editor again. To return to the default view
Enter LOOKUP and LOOKUPVALUE Expressions

Note: You can enter the LOOKUP function only in condition expressions, and you can enter the LOOKUPVALUE function only in action expressions. The LOOKUP function determines whether a lookup key exists in the lookup table, and the LOOKUPVALUE function retrieves the value associated with a key. It is best practice to use the LOOKUP function to verify that a key exists before you try to retrieve the value associated with the key.

To enter a condition expression that uses the LOOKUP function, select **LOOKUP** as the rule operator, and select the lookup table in the expression field.

To enter an action expression that uses the LOOKUPVALUE function, complete these steps:

1. Select **LOOKUPVALUE** as the operator on the THEN clause.
2. In the second field of the THEN clause, select the variable to which you want to assign the value that is retrieved from the lookup table.
3. Click ![button](image), and select the lookup table from which you want to retrieve the lookup value.
4. In the last field, select the variable whose value matches the value of the lookup key, or select the actual value of the lookup key. To select the value of the lookup key:
   a. Select **Select a lookup key**. SAS Intelligent Decisioning displays the list of lookup entries in the specified table.
   b. Select the record whose key you want to use in the expression, and click **Save**.

For example, suppose you have a lookup table in which the lookup keys are ZIP codes and the lookup values are city names. You could use the following rule to test whether the lookup table contains an entry for the ZIP code, and, if the entry exists, retrieve the city name that is associated with the ZIP code.
## Entering Literal Data Values

Depending on whether you use the Expression Editor or enter expressions directly into the expression fields, you must enter some values differently.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>How To Enter Values</th>
<th>Example</th>
</tr>
</thead>
</table>
| Character | Enclose character strings in single quotation marks. For embedded quotation marks, use two single quotation marks. | 'Gold Account'  
|           |                     | 'd''oscail'   |
| Data grid | Enclose data grid JSON strings in single quotation marks.  
Note: The only time you might need to enter JSON strings manually is when you are using the DATAGRID_CREATE function. | See [SAS Intelligent Decisioning: Using Data Grids](https://support.sas.com/documentation/cdl/en/idvprod/65688/HTML/default/viewer.htm#idvprod085.htm) for an explanation of the JSON syntax and a description of the DATAGRID_CREATE function. |
| Date      | In the rule set editor, enter Date values by using the format DDMMYYYY. Enclose each value in single quotation marks followed by `d`.  
In the Expression Editor, use the DS2 function TO_DOUBLE and specify the DATE data type in order to cast the Date value so that it can be compared correctly to other variables. See [SAS DS2 Programmer's Guide](https://support.sas.com/documentation/cdl/en/ds2pg/66056/HTML/default/viewer.htm) for information about date, time, and timestamp values, and see [SAS DS2 Language Reference](https://support.sas.com/documentation/cdl/en/ds2lr/66052/HTML/default/viewer.htm) for information about the TO_DOUBLE function. | '01AUG2017'd  
|           |                     | to_double(date '2017-11-04') |
Data Type | How To Enter Values | Example
--- | --- | ---
Datetime | In the rule set editor, enter Datetime values by using the format DDMMYYYY:HH:MM:SS. Use 24-hour clock notation. Enclose each value in single quotation marks followed by \texttt{dt}.
 | '31AUG2017:15:00:00\texttt{dt}' | 

In the Expression Editor, use the DS2 function TO\_DOUBLE and specify the TIMESTAMP data type in order to cast the Datetime value so that it can be compared correctly to other variables. See \textit{SAS DS2 Programmer's Guide} for information about date, time, and timestamp values, and see \textit{SAS DS2 Language Reference} for information about the TO\_DOUBLE function.
 | \texttt{to\_double(timestamp '2017-11-04 10:54:34.012')} | 

Boolean | In the rule set editor, Boolean values are not enclosed in quotation marks. Enter only the values.
 | True
 | False | 

In the Expression Editor, use numeric values to indicate True or False. Specify 1 for True and 0 for False.
 | 1
 | 0 | 

Operators for Use in Expressions

The following table lists the operators that you can use in an expression. Do not enter a space between the elements of the operators \(\leq\), \(\geq\), or \(^\wedge\). Some mnemonic equivalents for these operators cannot be used in SAS Intelligent Decisioning expressions. See \textit{SAS DS2 Programmer's Guide} for more information about specifying operators in expressions.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Multiply</td>
<td>0.085 * sales</td>
</tr>
<tr>
<td>/</td>
<td>Divide</td>
<td>amount / 5</td>
</tr>
<tr>
<td>Operator</td>
<td>Definition</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>+</td>
<td>Add</td>
<td>num + 3</td>
</tr>
<tr>
<td>-</td>
<td>Subtract</td>
<td>sale - discount</td>
</tr>
<tr>
<td>**</td>
<td>Raises the first operand to the power of the second operand</td>
<td>num1**num2</td>
</tr>
<tr>
<td>=</td>
<td>Equal to</td>
<td>tries = maxTriesAllowed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to</td>
<td>insufficientFunds != True</td>
</tr>
<tr>
<td>^=</td>
<td>Not equal to</td>
<td>balance ^= 'low'</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
<td>daysLate &gt; 5</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
<td>balance &gt;= 1000</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
<td>balance &lt;= 250</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>The maximum of the left and right operands</td>
<td>num1 &lt;&gt; num2</td>
</tr>
<tr>
<td>IN (value-list)</td>
<td>Equal to an item in value-list</td>
<td>risk in ('high','medium','low')</td>
</tr>
<tr>
<td>NOT IN (value-list)</td>
<td>Not equal to an item in value-list</td>
<td>offerPercent not in (10,20,30)</td>
</tr>
<tr>
<td>LIKE 'pattern'</td>
<td>If the variable’s value matches the expression pattern in pattern, the result is true.</td>
<td>like 'HS%PP'</td>
</tr>
<tr>
<td>expression AND expression</td>
<td>If both expressions are true, the result is true.</td>
<td>dateExpired &gt;= '01AUG2015'd AND dateExpired &lt;= '31AUG2015'd</td>
</tr>
<tr>
<td>expression OR expression</td>
<td>If either expression is true, the result is true.</td>
<td>dateEnrolled &gt;= '01JAN2015' OR member = True</td>
</tr>
</tbody>
</table>
Using the LIKE Operator

The LIKE operator determines whether the value of a variable matches a pattern-matching expression. An expression that uses the LIKE operator has the following syntax:

LIKE 'pattern-matching-expression'

If a variable's value matches the pattern that is specified by \textit{pattern-matching-expression}, the expression evaluates to true (1). Otherwise, the expression evaluates to false (0).

There are three classes of pattern-matching characters.

\begin{table}[h]
\centering
\caption{Pattern-Matching Characters}
\begin{tabular}{|l|l|}
\hline
\textbf{Character} & \textbf{Description} \\
\hline
underscore (_)& Matches any single character \\
\hline
percent sign (%) & Matches any sequence of zero or more characters  \\
& \textbf{Note:} Be aware of the effect of trailing blanks. To match values, you might have to use the TRIM function to remove trailing blanks. \\
\hline
any other character & Matches that character \\
\hline
\end{tabular}
\end{table}

The LIKE expression is case sensitive. To search for mixed-case strings, use the UPCASE function to create an uppercase version of the variable that you want to search. You can use a temporary variable to store the results of the UPCASE function. Use the LIKE operator to search the uppercase version of the variable. For example, you can search the variable Part_Number for mixed-case strings that begin with HS and end with PP by using the two rules shown in the following figure.

The following table shows examples of the matches that result if you search a variable that could have these values: Smith, Smooth, Smothers, Smart, Smuggle.
Table 2.3  Examples of LIKE Expressions

<table>
<thead>
<tr>
<th>LIKE Expression Example</th>
<th>Matching Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>like 'Sm%'</td>
<td>Smith, Smooth, Smothers, Smart, Smuggle</td>
</tr>
<tr>
<td>like '%th'</td>
<td>Smith, Smooth</td>
</tr>
<tr>
<td>like 'S__gg%'</td>
<td>Smuggle</td>
</tr>
<tr>
<td>like 'S_o'</td>
<td>(no matches)</td>
</tr>
<tr>
<td>like 'S_o%'</td>
<td>Smooth, Smothers</td>
</tr>
<tr>
<td>like 'S$th'</td>
<td>Smith, Smooth</td>
</tr>
</tbody>
</table>

Using Functions in Expressions

SAS Intelligent Decisioning supports the following functions in expressions:

- LOOKUP and LOOKUPVALUE functions. Condition expressions can contain the LOOKUP function, and action expressions can contain the LOOKUPVALUE function. However, if the expression contains the LOOKUP or LOOKUPVALUE function, then the expression cannot contain anything else. For more information, see “Enter LOOKUP and LOOKUPVALUE Expressions” on page 20, “LOOKUP Function” on page 77, and “LOOKUPVALUE Function” on page 78.

- SAS DS2 functions. For additional information about these functions and additional DS2 functions, see SAS DS2 Language Reference.

- SAS Data Quality Server functions. For information about these functions, see “Functions and CALL Routines” in SAS Data Quality and SAS Data Quality Server: Language Reference. These functions appear in the expression editor only if SAS Data Quality Server is installed.

Note: You cannot publish rule sets that use SAS Data Quality functions to SAS Micro Analytic Service destinations. Also, rule sets that use these functions fail when they are run inside the database.

- Data grid functions. For more information about these functions, see “Data Grid Functions” in SAS Intelligent Decisioning: Using Data Grids.

To enter a function in an expression, use the Expression Editor. See “Using the Expression Editor” on page 19 for more information.
Working with Missing Values

You can use the MISSING function to check for missing values. This function returns 0 (false) or 1 (true). Missing values have a value of false when you use them with logical operators such as AND or OR. You can use the MISSING function to eliminate errors, notes, and warnings in the SAS log that are caused by missing values.

In expressions, you can use the period (.) to denote missing numeric values, and two single quotation marks with no space (the empty string ‘’ ) to denote missing character values.

For information about how DS2 processes nulls and SAS missing values, see SAS DS2 Programmer’s Guide.

Delete Expressions or ELSE Rules

To delete a condition or action expression, click for that expression.

To delete an entire ELSE rule, select the rule, right-click on the rule, and select Delete the selected object.

Managing Rules

Duplicate a Rule

1. Click for the rule, and select Duplicate rule. The Duplicate Rule window appears.

2. Enter a name for the duplicate rule if you do not want to use the default name.

3. If you are duplicating an ELSE rule, select whether you want the duplicate rule to be a new IF rule or an additional ELSE rule, and click Duplicate.

Delete a Rule

1. Click for the rule, and select Delete rule.

2. (Optional) If the rule has an ELSE clause, SAS Intelligent Decisioning asks whether you want to delete both the IF and ELSE rules or only the IF rule. Click
Delete All to delete everything, or click Delete IF to delete only the IF clause. If you select Delete IF, then the ELSE clause becomes an assignment statement if it does not have any condition expressions. If it has a condition expression, it becomes the new IF rule.

**TIP** To delete only the ELSE clause, right-click on the ELSE operator and select **Delete the selected object**.

---

**Rename a Rule**

To rename the IF clause of a rule, select the existing rule name, and enter a new name.

To rename the ELSE clause of a rule:

1. Right-click on the ELSE operator and select **Rename rule**.

2. Enter the new name and click **Rename**.

**TIP** Rule names for ELSE clauses do not appear in the rule set editor.

---

**Reorder Rules**

To move a rule up or down within an IF or ELSE clause, select the rule, and click ↑ or ↓.
Copy a Rule Set URL

To create a link for external documentation that automatically opens a rule set in SAS Intelligent Decisioning, complete these steps:

1. Open the rule set.
2. Click ☰, and select **Copy object URL**. The Copy Rule Set URL window appears.
3. Click **Copy**, and then click **Close**.
   Paste the link into your documentation.

Generate PDF Documentation for a Rule Set

You can generate detailed documentation for a rule set as a PDF document. The PDF includes the rule set properties, details about each of the rules, and a table of the variables that are used in the rule set. You can download additional documents for lookup tables that are used in the rule set.

1. Open the rule set.
2. Click ☰, and select **Create document**. The Create Document window appears.

   **TIP** If this option is disabled, there might be unsaved changes. Click 🗑.

3. (Optional) Enter a name for the document if you do not want to use the default name.
4. (Optional) Select **Choose additional documents to download** to display a window from which you can download additional documents. You can download documents for any lookup tables that are used in the rule set.
5. Click **Create**. SAS Intelligent Decisioning creates the PDF. The result depends on your browser’s default download actions. The browser might automatically download the file and display a notification in the bar at the bottom of the browser window. Alternatively, the browser might ask you to choose a download location.

   If you download additional documents in **Step 4**, the Download Additional Documents window appears.
(Optional) Click Download for each additional document that you want to download.

(Optional) Click Close to close the Download Additional Documents window.

Managing Rule Sets

Duplicate Rule Sets

Note: You cannot duplicate a rule set if it is open.

To duplicate a single rule set:

1. In the Rule Sets view, select the rule set that you want to duplicate.
2. Click ‹ and select Duplicate. The Duplicate Rule Set window appears.
3. Enter a new name for the duplicate rule set
4. (Optional) Enter a description for the rule set.
5. Select the version of the rule set that you want to duplicate.
6. Click ‗ and select the location where you want to save the duplicate rule set.
7. Click Duplicate.

To duplicate multiple rule sets:

1. In the Rule Sets view, select the rule sets that you want to duplicate.
2. Click ‹ and select Duplicate. SAS Intelligent Decisioning duplicates the rule sets and appends _Copy to the names of the duplicate copies. If needed, a number is also appended to the names of the duplicate copies.

Delete Rule Sets

Note: You cannot delete a rule set if it is open.

In the Rule Sets view, select the rule sets that you want to delete, click ″, and select Delete.
Rename Rule Sets

Note: You cannot rename a rule set if it is open.

1. In the Rule Sets view, select the rule set that you want to rename.
2. Click and select Rename. The Rename window appears.
3. Enter a new name for the rule set, and click Rename.

Move Rule Sets to a Different Folder

1. In the Rule Sets view, select the rule sets that you want to move.
2. Click and select Move. The Choose a Location window appears.
3. Select the location to which you want to move the rule sets, and click OK.

Managing Versions of Rule Sets

Set the Displayed Version

The displayed version is the version whose information is displayed on the other tabs, such as the Properties and Rule set tabs. On the Versions tab, a ✓ indicates the displayed version. To change the displayed version, select the version that you want to view, and click Set Version. The displayed version is shown in the title bar.

Create a New Version

Note: The current version of an object is the version with the highest version number. When you create a new version, SAS Intelligent Decisioning locks the current version before it creates the new version.
Note: You cannot save changes to a version that is locked. If you modify a version that is locked and click , SAS Intelligent Decisioning asks you if you want to replace the current unlocked version with your edited version.

**IMPORTANT** You cannot unlock a locked version.

To create a new version:

2. Select the version type: Minor or Major. Version numbers follow the format Major.Minor. If you select Major, the number to the left of the period is incremented. If you select Minor, the number to the right of the period is incremented.
3. (Optional) Enter information about the new version in the Notes field.

**TIP** You can edit these notes at any time on the Versions tab.

4. Click Save.

**Copy the Content of a Version**

You can copy the content of an object's version in the category view or on the Version tab for the object.

1. In the category view, complete these steps:
   a. Select the rule set whose contents you want to copy.
   b. Click ‌, and select Copy version. The Copy Version window appears.
   c. Select the version whose contents you want to copy.

   Alternatively, on the Versions tab of an object:
   a. Select the version whose contents you want to copy.
   b. Click ‌, and select Copy version. The Copy Version window appears.

2. Click ‌, and select the target rule set into which you want to paste the contents of the version. You can paste version contents only into an object of the same type. If the rule set from which you are copying the version is a filtering rule set, then you must paste the contents into another filtering rule set. If it is an assignment rule set, then you must paste the version contents into another assignment rule set.

   When you paste the contents, SAS Intelligent Decisioning creates a new version of the target object.
Select whether you want to create a new major or minor version of the target object.

(Optional) Modify the notes that will be associated with new version of the target object.

Click Paste Version, and then click Yes.

Delete a Version

Note: In order to be able to delete a specific version of a rule set, you must have permission to delete the rule set itself.

On the Versions tab, select the version that you want to delete, click Delete, and select Delete.

You cannot delete the current version.

Upgrade an Object Version That Is Used in a Decision

If you create a new version of an object that is already used in a decision, you can upgrade the decision to use the new version.

1. On the Versions tab for the object, click ..., and select Upgrade decisions. The Upgrade Decisions window appears. This window lists all of the decisions that include the object.

2. In the Version to upgrade to field, select the version of the object to which you want to upgrade the decisions.

3. Select the check boxes for the decisions that you want to upgrade, and click Upgrade Decisions.

Testing a Rule Set

Create and Run a New Test

Testing a rule set is optional, but doing so is a best practice. Testing enables you to discover any problems before the rule set is published and incorporated into a production system.
1. On the **Scoring** tab, click the **Tests** tab.

2. Click **New Test**. The New Test window appears.

3. Enter a name for the test if you do not want to use the default name.

4. (Optional) Enter a description for the test. Descriptions are limited to 1000 characters.

5. (Optional) Click 📌 for the **Location** field, and select the folder where you want to save the test definition and results.

   **TIP** Selecting a location is optional, but it is highly recommended. Storing test definitions and test results in a folder simplifies the tasks of setting permissions and transferring the test files.

6. Click 📊 for the **Input table** field, select the input table for the test, and click **OK**.

7. Verify or change the variable mappings. The input variables in the rule set must be mapped to columns in the input table that you selected for the test.

   SAS Intelligent Decisioning automatically maps the input variables in the rule set to columns in the input table when the names and data types of the variables match those of the table columns. If any input variables cannot be mapped automatically, an error message is displayed.

   ![Input table](image)

   You can change the automatic variable mappings in the Variable Mappings window.

   **To change variable mappings:**

   a. Click **Variables**. The Variable Mappings window appears.

   b. For each input variable, select the table column to which the variable should be mapped. Alternatively, for Decimal, Integer, and Character variables, you can select **Use value** for the table column, and specify a literal value in the **Value** column. When you are entering literal values, remember these rules:

      - Do not enclose character strings in quotation marks.
      - To specify a missing value for character variables, select **Use value** and leave the **Value** column empty. When SAS Intelligent Decisioning generates code, it generates an empty string (""). For numeric values, enter a period (.)

   ![Note](image)

   **Note:** For content that will be published to a SAS Micro Analytic Service destination, the values of date and datetime input variables must be numeric. They cannot be formatted values such as 18jul2019.

   c. Click **OK** to close the Variable Mappings window.

8. (Optional) Click **Advanced** to display the advanced options.
9 (Optional) Click and select the library where you want to write the output of the test.

10 (Optional) Select the version of the rule set that you want to test.

11 (Optional) Select the variable that you want to serve as an input debug variable. You can specify an input-only variable or an input-output variable. The rule set writes the name and value of this variable to the log for each input record that is processed. It writes the value just before the logic of the rule set is executed for the input record.

For more information, see "Debugging Rule Set Tests" on page 38.

12 (Optional) Select Preserve unmapped columns in the output table if you want columns that are not mapped to an output variable to be written to the output table.

13 Click Run to run the test. Alternatively, click Save to save the test definition without running it.

The status of the test is indicated by the icon in the Status column.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The test is not ready to run. The test definition is not complete, or it might contain errors.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The test is defined correctly and is ready to run.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The test is running.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The test completed successfully.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The test completed, but warnings were issued in the SAS log. The URI to the log file is shown on the Test Results page. See Step 14.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The test did not run successfully. Check the SAS log for information. The URI to the log file is shown on the Test Results page. See Step 14.</td>
</tr>
</tbody>
</table>

14 Click in the Results column to view the results of the test.

15 On the Test Results page, click Test Results in the navigation pane to display the URIs and other information for the test. Click Output, Code, or Log to display the output data set, the code that was generated by SAS Intelligent Decisioning, or the SAS log that was generated when the code was run.

**TIP** On the Output page, you can click on the values of character variables to display the entire value in a separate window. For data grid variables, you can choose to view the variable value in three different formats:

- Click the Data Grid tab to view the data grid value as a table.
- Click the Formatted tab to view the data grid as a formatted JSON character string.
Click the Plain tab to view the data grid as an unformatted character string.

**TIP** On the Log page, you can click ⌁ to download the log file.

---

## Run a Rule-Fired Analysis

If a rule’s conditions evaluate to True, then the rule is said to have fired. Rule-fired data includes summary information about how many times each rule fired and detailed information for each time that a rule evaluates to True. See “How Rules Are Evaluated and When Rule-Fired Records Are Generated” on page 18 for more information.

**Note:** Rule-fired data is not recorded for decisions that do not include a rule set, subdecisions, input records that are filtered out with a filtering rule set, or eligibility rules in a treatment. For rule sets that iterate over a data grid (in other words, the rule sets score the rows in the data grid), the rule-fired data indicates that the rules in the rule set fire once for the entire data grid instead of firing once for each row in the data grid.

**TIP** This rule-fired analysis uses the data that is in the ruleFiredFlags column in the test results output table. To analyze rule-fired data that is in the subject contact history, use the %DCM_GET_SUBJECTCONTACT_HISTORY and %DCM_RULEFIRE_DETAIL macros. For more information, see SAS Intelligent Decisioning: Macro Guide.

1. On the Test Results page, click **Rule-Fired Analysis** in the navigation pane.

2. Click **Run Rule-Fired Analysis**. SAS Intelligent Decisioning analyzes the test results to determine which rules fired for each row in the input table, and displays the Analysis page.

   The Analysis page displays the number of rules that fired for each output record that was generated by the decision. The number in the **Rules Fired Count** column is a link to more information. You can click on this link to display the rules that fired for that output row.

   For example, the following displays shows the rule-fired analysis for the low_ratio rule set.
3 Click on a number in the **Rule Fired Count** column. SAS Intelligent Decisioning displays the Rule Fired Count window. This window shows which rules produced the selected output record.

4 Click **Close** to close the Rule Fired Count window.

5 Click **Plot** in the navigation pane. SAS Intelligent Decisioning displays a bar chart that shows how many times each rule fired. Position your cursor over a bar to display the name of the rule and the number of times that the rule fired.
6 Click **Rule-Fired Analysis** in the navigation pane to display the URIs and other information for the rule-fired test.

7 Click **Close** to close the rule set.

---

### Working with Test Output Data

After you run a test, you can work with the output table in other SAS applications to analyze the data, create and compare models, discover relationships hidden in the data, and generate reports based on the data.

**Note:** The actions available to you depend on the applications that are available at your site.

On the Test Results page, select the **Output** table in the navigation pane, click **Actions**, and select one of the following options:

- **Explore Lineage**
  opens SAS Lineage Viewer. SAS Lineage Viewer enables you to better understand the relationships between objects in your SAS Viya applications. These objects include data, transformation processes, reports, and visualizations. For more information, see *SAS Lineage Viewer: User’s Guide*.

- **Explore and Visualize Data**
  opens the output table in SAS Visual Analytics. SAS Visual Analytics enables you to create, test, and compare models based on the patterns discovered during exploration of the data. You can export the model before or after performing model comparison for use with other SAS products or to put the
model into production. SAS Visual Analytics supports a range of visualization, discovery, and reporting features. For more information, see SAS Visual Analytics: Overview.

Prepare Data
opens the output table in SAS Data Studio. SAS Data Studio enables you to perform data transforms such as joining tables, appending data to a table, transposing columns, creating calculated columns, and so on. For more information, see SAS Data Studio: User’s Guide.

Manage Data
opens SAS Data Explorer. SAS Data Explorer enables you to import data, connect to databases, and load tables into memory. For more information, see SAS Data Explorer: User’s Guide.

Debugging Rule Set Tests
When you create a test, you can specify a variable to use as a debugging variable. You can specify an input-only variable or an input-output variable. The rule set writes the name and value of this variable to the log for each input record that is processed. It writes the value just before the logic of the rule set is executed for the input record. For more information, see Step 11 on page 34 of “Create and Run a New Test” on page 32.

To capture variable values for input-only or temporary variables after the rule set logic has executed for a specific record, you can specify that the variable is an output variable, and then re-run the test. Before publishing the rule set to a production environment, return the input and output settings for the variable to their previous settings. For more information, see “Input Variables, Output Variables, and Temporary Variables” on page 10 and “Edit Variable Properties” on page 13.

Publishing and Validating a Rule Set

Publishing a Rule Set

Introduction to Publishing
Publishing content makes it available to other applications. Publishing a rule set creates an entity that can be managed and run in another environment. For example, if you publish content to a SAS Micro Analytic Service destination, SAS Intelligent Decisioning generates code for the content and writes that code as a module in SAS Micro Analytic Service. The module becomes a callable REST API endpoint, independent of SAS Intelligent Decisioning.
Publish Rule Sets

**TIP** To view the publishing history for a rule set, open the rule set, and click the **History** tab.

1. Select the rule sets that you want to publish.
   - To publish the latest version of one or more rule sets, select rule sets on the **Rule Sets** page, click **Publish**, and select **Publish**.
   - To publish a selected version of one rule set:
     1. Open the rule set, and click the **Versions** tab.
     2. Select the version that you want to publish, click **Set Version**, and then click **Publish**.

   The Publish Rule Sets window appears.

2. Select the destination to which you want to publish. The publishing destinations that are available to you depend on what is configured at your site. See **SAS Viya Administration: Publishing Destinations** for more information.

   **Note:** You cannot publish filtering rule sets or rule sets that use SAS Data Quality functions to SAS Micro Analytic Service destinations. Also, rule sets that use these functions will fail if they are run inside the database.

3. (Optional) In the **Items to Publish** section, edit the **Published Name** if you do not want to use the default name for the published module. The maximum length and character restrictions differ depending on your destination. See **Table 2.4**.

**Table 2.4 Requirements and Restrictions for Published Names**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Maximum Length</th>
<th>Requirements And Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Micro Analytic Service</td>
<td>100</td>
<td>The published name cannot contain the following characters: @ # $ % ^ &amp; * ( )</td>
</tr>
<tr>
<td>Teradata</td>
<td>128</td>
<td>The published name must start with a letter or an underscore. It cannot contain spaces, multi-byte characters, or special characters other than the underscore.</td>
</tr>
<tr>
<td>Destination</td>
<td>Maximum Length</td>
<td>Requirements And Restrictions</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>SAS Cloud Analytic Services (CAS)</td>
<td>128</td>
<td>The published name cannot contain single or double quotation marks.</td>
</tr>
<tr>
<td>Apache Hadoop</td>
<td>128</td>
<td>The published name cannot contain colons (:) or double quotation marks.</td>
</tr>
</tbody>
</table>

4. (Optional) If you have previously published the rule set, select the check box in the **Replace** column in order to replace the previously published rule set of the same name in the same destination.

5. (Optional) Select the **Rule Fired Tracking** check box if you want the published rule set to generate rule-fired data.

**Note:** This option is disabled if you are publishing content to SAS Micro Analytic Service.

This rule-fired data is recorded in the ruleFiredFlags column in the output table. The rule-fired data that is recorded when you select **Record rule-fired data** for a record contacts node is recorded in the subject contact history.

6. Publish the rule sets.

- To publish content to a SAS Cloud Analytic Services (CAS) destination, you must reload the CAS destination table in order to make the newly published item available to other applications. Select one of the following options to publish the decision:
  - **Publish** publishes the rule sets and automatically reloads the CAS destination table. If another user is executing the code for an item that was previously published to CAS while the destination table is being reloaded, reloading the table might cause temporary problems with accessing the table content. After the table is reloaded, all authorized users can access all items in the table.
  - **Publish without reloading** publishes the rule sets but does not reload the CAS destination table. You must manually reload the table in order for the newly published items to be accessible.

- To publish the rule sets to Teradata, Apache Hadoop, or SAS Micro Analytic Service destinations, click **Publish**. You do not need to reload the destination table when you publish to these destinations.

The Publishing Results window appears. It displays the names of the published items, their status, and information about any issues that were encountered during the publishing process.

7. After the status changes to **Published successfully**, click **Close** to close the Publishing Results window.

8. (Optional) Click **Close** to close the rule set.
Validate a Published Rule Set

You can test the published rule set in the target publishing destination. When you publish the rule set, a validation test is automatically defined for that rule set in that destination. To run the publishing validation test:

1. On the **Scoring** tab, click the **Publishing Validation** tab. The icon in the **Status** column indicates that the test is not ready to run. The icon indicates that the test is ready to run.

2. Click on the test name. The **Edit Publishing Validation Test** window appears.

   Note: To generate the name of the publishing validation test, SAS Intelligent Decisioning appends a timestamp to the rule set name. The timestamp indicates when the rule set was published.

3. (Optional) Click in the **Location** field, and select the folder where you want to save the test definition and results.

   **TIP**  
   Selecting a location is optional, but it is highly recommended. Storing test definitions and test results in a folder simplifies the tasks of setting permissions and transferring the test files.

4. Click in the **Input table** field, and select the input table for the test.

   Note: For content that will be published to a SAS Micro Analytic Service destination, the values of date and datetime input variables must be numeric. They cannot be formatted values such as 18jul2019.

   Note: If the input table contains a character column, and that column contains control characters in any row, do not use the table as input for publishing validation tests.

   If you are validating content that was published to SAS Micro Analytic Service, the time required to run the test depends on the number of worker threads on your system, the number of threads in the middle tier, and the network latency between CAS and the middle tier server. It is recommended that you select an input table with as few input records as needed to accurately test the published content. See *SAS Micro Analytic Service: Programming and Administration Guide* for more information.

5. (Optional) Expand the **Advanced** section, click in the **Output data library** field, and select a library to store the validation test output data.

6. Click **Run** to run the test. Alternatively, click **Save** to save the test definition without running it.

   The status of the test is indicated by the icon in the **Status** column.
<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>The test is not ready to run. You might need to edit the test and select an input data table or map variables.</td>
</tr>
<tr>
<td>🔄</td>
<td>The test is defined correctly and is ready to run.</td>
</tr>
<tr>
<td>🔄</td>
<td>The test is running.</td>
</tr>
<tr>
<td>✅</td>
<td>The test completed successfully.</td>
</tr>
<tr>
<td>🔄</td>
<td>The test completed, but warnings were issued in the SAS log. The URI to the log file is shown on the Test Results page. See Step 14.</td>
</tr>
<tr>
<td>🔄</td>
<td>The test did not run successfully. Check the SAS log for information. The URI to the log file is shown on the Test Results page. See Step 14.</td>
</tr>
</tbody>
</table>

7 Click 🔄 in the Results column to view the test results.

8 On the Test Results page, click Test Results in the navigation pane to display the URIs and other information for the test. Click Output, Code, or Log to display the output data set, the code that was generated by SAS Intelligent Decisioning, or the SAS log that was generated when the code was run.

**TIP** On the Output page, you can click on the values of character variables to display the entire value in a separate window. For data grid variables, you can choose to view the variable value in three different formats:

- Click the Data Grid tab to view the data grid value as a table.
- Click the Formatted tab to view the data grid as a formatted JSON character string.
- Click the Plain tab to view the data grid as an unformatted character string.

**TIP** On the Log page, you can click ⬇️ to download the log file.

9 Click Close to close the rule set.

## Executing Published Content

How you execute published content depends on the destination to which the content is published.
Executing Content Published to SAS Micro Analytic Service Destinations

The user who is executing the published content must be authenticated. In SAS Viya, authentication options vary, based on which interface and operating system are used in your environment. External mechanisms include direct LDAP authentication, host authentication, Kerberos, Security Assertion Markup Language (SAML), and OAuth 2.0 with OpenID Connect. For additional information, see SAS Viya Administration: Authentication.

When a rule set or decision is published from SAS Intelligent Decisioning to a SAS Micro Analytic Service destination, an EXECUTE step is created in the published module. For information about the request and response data formats used in this step, see Execute a step in the REST API documentation for the Micro Analytic Score API. For an example that uses Python to execute a published decision in the maslocal destination, see “Execute a Published Decision” in SAS Intelligent Decisioning: Decision Management REST API Examples.

Executing Content That Has Been Published to SAS Cloud Analytic Services Destinations

To execute content that has been published to SAS Cloud Analytic Services (CAS), use the CAS Model Publishing and Scoring action set. For example, the following code runs a model named Evaluate_Loans on the local CAS server.

```sas
/* Start a CAS session named _mmcas_. */
cas _mmcas_;

/* Create librefs for all existing caslibs so that they are visible in the SAS Studio Libraries tree. */
caslib _all_ assign;

proc cas;
/* Specify the session to use for the runModelLocal action. */
session _mmcas_;

/* Define the parameter list for the runModelLocal action. */

destination_model_table = "targetCode";
destination_model_lib = "public";

destination_model = "Evaluate_Loans";

dp_inputTable="hmeq_test";
dp_inputCASLib="public";

dp_outputTable="hmeq_test_dm";
dp_outputCASLib="public";
```
parmlist = {
    modelTable={
        name=destination_model_table,
        caslib=destination_model_lib
    },
    modelName=destination_model,
    inTable={
        name=dp_inputTable,
        caslib=dp_inputCASLib
    },
    outTable={
        name=dp_outputTable,
        caslib=dp_outputCASLib
    }
};

/* Load the modelPublishing action set. */
loadactionset "modelPublishing";

/* Run the model locally on the CAS server. */
action runModelLocal submit result=r status=rc / parmlist;
run;
quit;

You can submit this code in SAS Data Studio. To open SAS Data Studio, click ⋯ and select Prepare Data. For more information, see SAS Data Studio: User’s Guide.

You can view additional examples of using this CAS action set to execute published content by viewing the test results that are generated by publishing validation tests. On the Test Results page for a decision or rule set, click Code to display the code that was generated by SAS Intelligent Decisioning. For information about running publishing validation tests and viewing the results, see “Validate a Published Decision” on page 133 and “Validate a Published Rule Set” on page 41.

For more information about CAS and the Model Publishing and Scoring action set, see the following documentation:

- Getting Started with CASL Programming
- SAS Cloud Analytic Services: CASL Reference
- SAS Cloud Analytic Services: User’s Guide

---

Importing and Exporting Rule Sets

You can import rule sets from and export rule sets to comma-delimited (CSV) files. The format of the CSV file is the same format that is used by the %DCM_IMPORT_RULESET macro. For more information, see “Format of Rule Set CSV Input File” in SAS Intelligent Decisioning: Macro Guide.
Note: If you do not need to edit content as a CSV file, you should use either the SAS transfer service or SAS Environment Manager to import and export content. For information about the transfer service, see "Promotion: How to Import (Command-Line Interface)" in SAS Viya Administration: Promotion (Import and Export). For information about using SAS Environment Manager, see "Export Content" in SAS Viya Administration: Folders and “Import a Package to a Folder” in SAS Viya Administration: Folders.

Import a Rule Set

**IMPORTANT** If you import rules from a CSV file into a rule set that already contains rules, the existing rules are replaced with the rules that are defined in the CSV file. To avoid replacing existing rules, create a new empty rule set, and then import the contents of the CSV file into the empty rule set. To append rules to a rule set, you can export the rule set, add content to the CSV file, and then re-import the CSV file.

1. Open the rule set into which you want to import new rules.
2. Click Import. If the rule set already contains rules, asks you if you want to replace the existing rules.
3. Click , and select the CSV file that you want to import.
4. Select the encoding of the CSV file, and click Import.

Export a Rule Set

**IMPORTANT** Do not modify the file structure or the header row in the exported CSV file. You can modify the data values.

To export a rule set, open the rule set and click Export. The table is exported into a CSV file, and a notification appears in the download bar at the bottom of the browser window.
Working with Treatments and Treatment Groups

About Treatments and Decisions ......................................................... 48
Example: A Decision That Includes a Treatment Group .................. 49
About Attributes, Eligibility Rules, and Effective Dates ................... 51
  Attributes and Attribute Aliases ................................................. 51
  Eligibility Rules ........................................................................... 51
  Effective Dates ............................................................................. 52
Arbitrating Treatments .................................................................... 52
About Channels .............................................................................. 52
Define a Treatment ......................................................................... 53
Define a Treatment Group ............................................................... 54
Activate a Treatment Group ............................................................. 55
Managing Attributes and Aliases ...................................................... 56
  Define Custom Attributes .......................................................... 56
  Add Attributes from a Different Treatment ................................. 57
  Duplicate Attributes .................................................................... 57
  Delete Attributes from a Treatment ............................................. 58
  View Attribute Aliases for a Treatment Group ............................ 58
  Remove an Attribute Alias from a Treatment Group .................. 58
  Determine Which Treatments Use an Attribute ......................... 58
Generate PDF Documentation for a Treatment Group ....................... 58
Delete Treatments from a Treatment Group ..................................... 59
Managing Eligibility Rules for a Treatment ..................................... 59
  Remove Eligibility Rule Set from a Treatment ......................... 59
  Change the Eligibility Rule Set for a Treatment ......................... 59
Managing Treatments and Treatment Groups .................................. 60
  Duplicate Treatments or Treatment Groups .............................. 60
  Delete Treatments or Treatment Groups ..................................... 61
  Rename a Treatment or Treatment Group ................................. 61
About Treatments and Decisions

In SAS Intelligent Decisioning, you use treatments to define offers that you might want to present to subjects that contact your company as a result of an inbound marketing campaign.

In an inbound marketing campaign, a subject initiates contact with a company through a specific channel such as email, a website, or a phone call. A subject can be an individual customer, a company account, a household, or other entity. Depending on the channel, an application or website captures the subject’s behavior or a human representative enters the subject’s information into a form or a customer service application.

The subject usually has an identification number (a subject ID). The customer service application or other calling application uses the subject ID to send a request to SAS Intelligent Decisioning that invokes a decision. The decision typically performs the following tasks:

1. Uses the subject ID to issue a web service call or to query the company database in order to retrieve additional customer data, such as name, address, and household income, if such information exists.

2. Determines the set of offers for which the subject qualifies.

3. Arbitrates the set of offers to determine which ones the subject is most likely to respond to.

4. Generates a response tracking code and subject contact history data. The subject contact history data typically includes the set of offers from step 2.

5. Returns to the calling application the response tracking code and the set of offers for which the subject qualifies.

At this point, the calling application or a customer service representative can present the offers to the customer. The calling application can use the response tracking code to update the subject contact history data. For example, it can record which treatments are presented to the customer and the subject’s response to the treatments.

“Example: A Decision That Includes a Treatment Group” on page 49 shows how to accomplish the tasks listed above by using a decision in SAS Intelligent Decisioning.
Example: A Decision That Includes a Treatment Group

Figure 3.1 shows a basic decision that defines offers, arbitrates these offers, and updates the subject contact history. This example contains only one treatment group and one record contacts node, but decisions can include multiple treatment groups and record contacts nodes.

This decision flow has the following elements:

1. The data query node named Get_customer_info queries the company database and retrieves additional information about the subject based on the subject ID. For example, if the treatment is an offer to upgrade the subject’s cell phone, the SQL query could retrieve the model number for the subject’s current phone, if
that information is available. For more information, see “Data Query Files” on page 86.

Nodes that retrieve additional information about the subject are typical but not required.

2 The treatment group name Combined_Treatments_Final defines a set of treatments (offers) and includes rules that define who is eligible to receive the treatments. The decision uses the information about the subject and the eligibility rules to determine the treatments for which the subject qualifies.

When you add a treatment group to a decision, SAS Intelligent Decisioning creates an output variable of type data grid. After the decision executes, this data grid contains a row for each treatment for which the subject qualifies.

You can include as many treatment groups in a decision as necessary. You can merge multiple output data grids into a single data grid if needed.

For more information, see the following topics:
- “About Attributes, Eligibility Rules, and Effective Dates” on page 51
- “About Channels” on page 52
- SAS Intelligent Decisioning: Using Data Grids

3 The model AggregatePromoPrediction and the rule set ArbitrationRules arbitrate the treatments. For example, the subject might qualify for several treatments, but you might want to offer them only the two treatments that they are most likely to accept. You might also want to return only the treatments that are appropriate for the channel.

- The model calculates a probability for each treatment (each row in the data grid) that indicates how likely the subject is to respond to the treatment.
- The rule set uses the DATAGRID_TOPN function to sort the treatment data grid based on the values of the probability column. The function returns the top two treatments to which the subject is most likely to respond. See “DATAGRID_TOPN” in SAS Intelligent Decisioning: Using Data Grids for more information.

You can use rule sets, scoring models, attribute values, or other methods to arbitrate treatments. For more information, see “Arbitrating Treatments” on page 52.

4 The record contacts node records information that you want to track, such as the treatments that are returned to the calling application and whether the record is used in generating aggregate reports for the channel. You can also specify variable values to track.

In real-time destinations (SAS Micro Analytic Service destinations), this node writes a record to the subject contact history. In all other destinations (SAS Cloud Analytic Services [CAS], Teradata, and Apache Hadoop), this node creates an output variable that contains the information that you specify that you want to track.

Record contacts nodes also create a response tracking code that the calling application can use to add additional information to the subject contact history.

TIP You can use record contacts nodes to track variable values even if you are not using treatments.
About Attributes, Eligibility Rules, and Effective Dates

A treatment defines the details of an offer that can be sent to a subject, who is eligible for the offer, and the dates during which the offer is available to be sent to a subject. A treatment is a set of attributes, eligibility rules, and effective dates.

Attributes and Attribute Aliases

Attributes are name-value pairs that define the specific details of the offer that you present to a subject. For example, for a treatment that gives subjects a 20% discount, you could define an attribute named DISCOUNT that has the value 20. To give subjects a special deal on a specific cell phone model, you can define an attribute named MODEL whose value is the specific model name. You could use treatment attributes to set values such as profitability, risk, cost, priority, or order. You can use the values of these attributes to arbitrate the treatments. For more information, see “Arbitrating Treatments” on page 52.

An attribute can be dynamic or fixed. You can customize dynamic attributes for each treatment group in which a particular treatment is included. See “Define Custom Attributes” on page 56 for more information.

You can assign aliases to attributes. An alias is an alternative name that you can assign to a treatment attribute after you add the treatment to a treatment group. Aliases are useful when attributes in different treatments within the group represent the same data but are defined with different names. For example, treatment A could define an attribute named DISCOUNT, and treatment B could define an attribute named DISC. Assuming that these two attributes represent the same value, you can assign an alias to one or both of the attributes. An attribute can have only one alias.

Eligibility Rules

Eligibility rules define who is eligible to receive the offer defined by the treatment. For example, your campaign can target people who already have a specific credit card and who are at least 30 years old, or people who own a home but who do not have a home equity line of credit. You define eligibility rules in the eligibility rule set. An eligibility rule set is a filtering rule set. For more information, see “About Rules, Assignment Rule Sets, and Filtering Rule Sets” on page 6.
Effective Dates

The effective dates for a treatment are the start and end dates when the treatment is active, that is, when it can be returned to a subject. Responses that are defined by a treatment are not returned to subjects outside of the effective dates. Effective dates are not required. A treatment that does not have effective dates is considered to always be active.

Note: The effective dates are always based on the time zone of the server where the decision is executed.

Arbitrating Treatments

You can determine which treatments a subject is most likely to respond to by using one or more of the following methods:

- use filtering rule sets to select only certain records for processing. Only the records whose conditions evaluate to True are processed by the remaining objects in the decision. For more information, see “About Rules, Assignment Rule Sets, and Filtering Rule Sets” on page 6.

Note: Filtering rule sets can be included directly in a decision, but they are also used as the eligibility rule set in treatments.

- use models to score treatments. For example, you might have a model that calculates propensity scores for individual treatments.

- use data grid functions to sort or subset the treatments according to the values of their attributes. For example, you could have attributes for value, profitability, or risk. For more information, see “Data Grid Functions” in SAS Intelligent Decisioning: Using Data Grids.

- use DS2 code files to create custom code for arbitrating the treatments. For more information, see Chapter 5, “Using Custom Code Files,” on page 81.

About Channels

Channels are routes by which your company and a subject are in contact. Typical channels are email, phone call, and website. A default set of channels is defined in the Treatment Channels lookup table. Your administrator can customize this lookup table for your enterprise. You can use this lookup table in rules to narrow the list of
offers to ones that are appropriate for the channel. See “Predefined Lookup Tables” on page 67 for more information.

You can create assignment rules to determine the channel through which your company and a subject are in contact if the channel was not specified in the original request.

---

**Define a Treatment**

**Create a New Treatment and Define Attributes**

1. Click 🔄, and then click **New Treatment**. The New Treatment window appears.
2. Enter a name for the treatment.
3. (Optional) Enter a description for the treatment.

   **TIP** You can edit the description at any time on the **Properties** tab.

4. Click 📄, and select the folder where you want to save the treatment.
5. Click **Save** to save the treatment. SAS Intelligent Decisioning opens the new treatment and displays the **Attributes** tab.
6. (Optional) Define treatment attributes. A treatment can have zero or more attributes. For more information about attributes, see “About Attributes, Eligibility Rules, and Effective Dates” on page 51.

   You can add new custom attributes, add attributes from other treatments, or duplicate attributes within the same treatment. For more information, see the following topics:

   - “Define Custom Attributes” on page 56
   - “Add Attributes from a Different Treatment” on page 57
   - “Duplicate Attributes” on page 57

**Specify Eligibility Rules**


7. On the **Eligibility Rule Set** tab, click **Add Rule Set**. The Select Rule Set window appears.
8. Select the filtering rule set that you want to use as the eligibility rule set for the treatment, and click **OK**.
9. At the top of the **Eligibility Rule Set** tab, select the version of the eligibility rule set that you want to use.

   **TIP** You can view or edit an eligibility rule set from the **Eligibility Rule Set** tab by clicking **Open**.
(Optional) Specify Effective Dates

10 On the Properties tab, select the Start date and End date for the period in which the treatment can be sent to a subject. If you do not specify effective dates, the treatment is considered as always active.

Note: The effective dates are always based on the time zone of the server where the decision is executed.

Define a Treatment Group

Create a New Treatment Group

1 Click , and then click New Treatment Group. The New Treatment Group window appears.

2 Enter a name for the treatment group.

3 (Optional) Enter a description for the treatment group.

TIP You can edit the description at any time on the Properties tab.

4 Click , and select the folder where you want to save the treatment group.

5 Click Save to save the treatment group. SAS Intelligent Decisioning opens the new treatment group and displays the Treatments tab.

Add Treatments

6 On the Treatments tab, click Add Treatments. The Add Treatments window appears.

7 Select the treatments that you want to add to the treatment group, and click OK.

TIP After you add a treatment to a treatment group, the variables in the eligibility rule set are added to the list of eligibility variables for the treatment group. You can view the eligibility variables for a treatment group on the Eligibility Variables tab.

8 (Optional) Select the version of each treatment that you want to use in the treatment group.

TIP You can change the version of a treatment at any time on the Treatments tab.
**Customize Dynamic Treatment Attributes**

For information about dynamic and fixed attributes, see “Define Custom Attributes” on page 56.

9 On the **Treatments** tab, click **Set Attributes**. The Set Attributes window appears. By default, dynamic attributes are automatically selected, and their values are set at run-time by the decision flow.

10 For each dynamic variable, verify that its setting is correct for the current treatment group.
   - If you want the attribute’s value to be set by the decision flow, leave the attribute selected.
   - If you want to enter a static value for the attribute, clear the check box, and enter the value that you want to be used in the current treatment group.

**Add Attribute Aliases**

For information about aliases, see “Attributes and Attribute Aliases” on page 51.

11 Click the **Attributes** tab.

12 Click 💾 for the attribute to which you want to assign an alias. The Choose an Alias window appears and displays the list of existing attribute names that you can associate with the selected attribute.
   - To assign an existing attribute name as an alias, select the attribute name in the list and click **OK**.
   - To enter a new alias, click **New Alias**, enter the new alias name, and click **Save**. When you create a new alias, SAS Intelligent Decisioning creates a new output variable for the treatment group.

**TIP** In the Set Attributes window, you can view the alias that is assigned to an attribute by clicking 🌀.

---

**Activate a Treatment Group**

Activating a treatment group creates a public module for each treatment in the group and makes the treatment group available to be referenced by decisions.

**IMPORTANT** If a decision that uses a treatment group is published to a SAS Micro Analytic Service destination, that decision always uses the latest active version of the treatment group.

When you publish a decision that uses a treatment group to SAS Cloud Analytic Services (CAS), Teradata, or Apache Hadoop, SAS Intelligent Decisioning creates a static copy of the active version of the treatment group. This static version is published with the decision. When the published decision is run, including when it is run in publishing validation tests, the static copy of the treatment group is used. For more information, see
To activate a treatment group, complete these steps:

1. Open the treatment group that you want to activate.
2. On the Versions tab, select the version that you want to activate, and click Activate.

**Managing Attributes and Aliases**

For additional information about attributes and aliases, see the following topics:

- "Attributes and Attribute Aliases" on page 51
- "Add Attribute Aliases" on page 55
- "Customize Dynamic Treatment Attributes" on page 55

**Define Custom Attributes**

1. On the Attributes tab of a treatment, click Add Attribute and select Custom attribute. The Add Attribute window appears.

2. Complete these steps for each attribute:
   
   a. Enter a name for the attribute.
   b. (Optional) Enter a description for the attribute.
   c. Select the data type for the attribute.
   d. Select Fixed or Dynamic, depending on how the attribute's value will be set when it is included in a treatment group.

   **Dynamic**
   
   The value of a dynamic attribute can be set by the decision at run-time. When you define a dynamic attribute, you can specify a default value for the attribute. Within each treatment group in which the attribute appears, you can specify that the attribute's value is set by the decision at run time, or you can set a static value that is used only within that specific treatment group.
Fixed
You define the value or set of values for fixed attributes when you define the attribute. The values cannot be customized in treatment groups.

Specify the value or values for the attribute.

- For dynamic attributes, you can specify only a single value. This value is used as the default value if the attribute’s value is not customized in treatment groups.
- For fixed attributes that are character, integer, or decimal attributes, you can specify a single value or a list of values. Check the List of values check box to specify a list of possible values for the attribute. In the Value field, press Enter after each specific value.
  
  For fixed attributes of other data types, you can specify only a single value.

Click Add to add the attribute to the table.

3 Click OK to save the attributes.

Add Attributes from a Different Treatment

1 On the Attributes tab of a treatment, click Add Attribute, and select Treatment. The Choose an Item window appears.

2 Select the treatment from which you want to copy an attribute, and click OK. The Add Attributes window appears.

3 Click to select all of the attributes, or select one or more individual attributes and click .

4 Click Add to add the attributes, or click Add and replace to replace existing attributes that have the same name.

Duplicate Attributes

1 On the Attributes tab of a treatment, select the attribute that you want to duplicate.

2 Click , and select Duplicate. The Duplicate Attributes window appears.

3 Enter a new name for the duplicate attribute, and click Duplicate.
Delete Attributes from a Treatment

On the Attributes tab of a treatment, select the attribute that you want to delete, click  
, and select Delete.

View Attribute Aliases for a Treatment Group

You can view attribute aliases in two ways:

- Click the Attribute Aliases tab in a treatment group.
- On the Treatments tab, click Set Attributes, and then click ✎ for a specific attribute.

Remove an Attribute Alias from a Treatment Group

On the Attribute Aliases tab of a treatment group, select the attributes whose aliases you want to remove, click  
, and select Remove alias.

Determine Which Treatments Use an Attribute

On the Attribute Aliases tab of a treatment group, select the attribute for which you want treatment information, click  
, and select Used by. SAS Intelligent Decisioning opens the Used By Treatments window, which lists the treatments that use the attribute.

Generate PDF Documentation for a Treatment Group

You can generate detailed documentation for a treatment group as a PDF document. The PDF includes the treatment group properties, and details about each treatment that is included in the group. Treatment details include treatment properties, the name of the eligibility rule set, and details about each of the treatment’s attributes.

1 Open the treatment group.
2 Click ☐, and select **Create document**. The Create Document window appears.

**TIP** If this option is disabled, there might be unsaved changes. Click [file].

3 (Optional) Enter a name for the document if you do not want to use the default name.

4 Click **Create**. SAS Intelligent Decisioning creates the PDF. The result depends on your browser's default download actions. The browser might automatically download the file and display a notification in the bar at the bottom of the browser window. Alternatively, the browser might ask you to choose a download location.

---

**Delete Treatments from a Treatment Group**

On the **Treatments** tab of the treatment group, select the treatments that you want to remove from the group, click ☐, and select **Delete**.

---

**Managing Eligibility Rules for a Treatment**

For additional information about eligibility rules, see “About Attributes, Eligibility Rules, and Effective Dates” on page 51 and “Specify Eligibility Rules” on page 53.

---

**Remove Eligibility Rule Set from a Treatment**

On the **Eligibility Rule Set** tab of a treatment, click **Remove**.

---

**Change the Eligibility Rule Set for a Treatment**

1 On the **Eligibility Rule Set** tab of a treatment, click **Replace**. The Select Rule Set window appears.

2 Select the filtering rule set that you want to use as the eligibility rule set for the treatment, and click **OK**.
At the top of the **Eligibility Rule Set** tab, select the version of the eligibility rule set that you want to use.

**TIP** You can view or edit an eligibility rule set on the **Eligibility Rule Set** tab by clicking **Open**.

---

## Managing Treatments and Treatment Groups

### Duplicate Treatments or Treatment Groups

**Note:** You cannot duplicate a treatment or treatment group if it is open.

To duplicate a single treatment or treatment group:

1. In the Treatments or Treatment Groups category view, select the treatment or treatment group that you want to duplicate.

2. Click \( \text{CTRL} \) and select **Duplicate**. The Duplicate Treatments or Duplicate Treatment Groups window appears.

3. Enter a new name for the duplicate treatment or treatment group.

4. (Optional) Enter a description for the treatment or treatment group.

5. Select the version of the treatment or treatment group that you want to duplicate.

6. Click \( \text{CTRL} \) and select the location where you want to save the duplicate treatment or treatment group.

7. Click **Duplicate**.

To duplicate multiple treatments or treatment groups:

1. In the Treatments or Treatment Groups category view, select the treatments or treatment groups that you want to duplicate.

2. Click \( \text{CTRL} \) and select **Duplicate**. SAS Intelligent Decisioning duplicates the latest version of the treatments or treatment groups and appends \\_Copy to the names of the duplicate copies. If needed, a number is also appended to the names of the duplicate copies.
Delete Treatments or Treatment Groups

Note: You cannot delete a treatment or treatment group if it is open.

1 In the Treatments or Treatment Groups category view, select the treatments or treatment groups that you want to delete.
2 Click ‹, and select **Delete**.

Rename a Treatment or Treatment Group

Note: You cannot rename a treatment or treatment group if it is open.

1 In the Treatments or Treatment Groups category view, select the treatment or treatment group that you want to rename.
2 Click ‹, and select **Rename**.
3 Enter a new name for the treatment or treatment group, and click **Rename**.

Move Treatments or Treatment Groups to a Different Folder

1 In the Treatments or Treatment Groups category view, select the treatments or treatment groups that you want to move.
2 Click ‹ and select **Move**. The Choose a Location window appears.
3 Select the location to which you want to move the treatments or treatment groups, and click **OK**.
Managing Versions of Treatments and Treatment Groups

Set the Displayed Version

The displayed version is the version whose information is displayed on the other tabs, such as the Properties and Attributes tabs. On the Versions tab, a ✓ indicates the displayed version. To change the displayed version, select the version that you want to view, and click Set Version. The displayed version is shown in the title bar.

Create a New Version

Note: The current version of an object is the version with the highest version number. When you create a new version, SAS Intelligent Decisioning locks the current version before it creates the new version.

Note: You cannot save changes to a version that is locked. If you modify a version that is locked and click ✓, SAS Intelligent Decisioning asks you if you want to replace the current unlocked version with your edited version.

IMPORTANT You cannot unlock a locked version.

To create a new version:


2 Select the version type: Minor or Major. Version numbers follow the format Major.Minor. If you select Major, the number to the left of the period is incremented. If you select Minor, the number to the right of the period is incremented.

3 (Optional) Enter information about the new version in the Notes field.

TIP You can edit these notes at any time on the Versions tab.

4 Click Save.
Copy the Content of a Version

You can copy the content of an object's version in the category view or on the Version tab for the object.

1 In the category view, complete these steps:
   a Select the treatment or treatment group whose contents you want to copy.
   b Click ⌧, and select Copy version. The Copy Version window appears.
   c Select the version whose contents you want to copy.

Alternatively, on the Versions tab of an object:
   a Select the version whose contents you want to copy.
   b Click ⌧, and select Copy version. The Copy Version window appears.

2 Click ⌧, and select the target treatment or treatment group into which you want to paste the contents of the version. You can paste version contents only into an object of the same type.

When you paste the contents, SAS Intelligent Decisioning creates a new version of the target object.

3 Select whether you want to create a new major or minor version of the target object.

4 (Optional) Modify the notes that will be associated with new version of the target object.

5 Click Paste Version, and then click Yes.

Upgrade an Object Version That Is Used in a Decision

If you create a new version of an object that is already used in a decision, you can upgrade the decision to use the new version.

1 On the Versions tab for the object, click ⌧, and select Upgrade decisions. The Upgrade Decisions window appears. This window lists all of the decisions that include the object.

2 In the Version to upgrade to field, select the version of the object to which you want to upgrade the decisions.

3 Select the check boxes for the decisions that you want to upgrade, and click Upgrade Decisions.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Lookup Tables and Functions</td>
<td>66</td>
</tr>
<tr>
<td>Predefined Lookup Tables</td>
<td>67</td>
</tr>
<tr>
<td>Create a New Lookup Table</td>
<td>67</td>
</tr>
<tr>
<td>Importing and Exporting Lookup Tables</td>
<td>68</td>
</tr>
<tr>
<td>Import or Refresh Lookup Tables</td>
<td>68</td>
</tr>
<tr>
<td>Export a Lookup Table</td>
<td>68</td>
</tr>
<tr>
<td>Add Lookup Table Entries</td>
<td>69</td>
</tr>
<tr>
<td>Edit Lookup Table Entries</td>
<td>69</td>
</tr>
<tr>
<td>Delete Lookup Table Entries</td>
<td>70</td>
</tr>
<tr>
<td>Copy a Lookup Table URL</td>
<td>70</td>
</tr>
<tr>
<td>Generate PDF Documentation for a Lookup Table</td>
<td>70</td>
</tr>
<tr>
<td>Managing Lookup Tables</td>
<td>71</td>
</tr>
<tr>
<td>Duplicate Lookup Tables</td>
<td>71</td>
</tr>
<tr>
<td>Delete Lookup Tables</td>
<td>71</td>
</tr>
<tr>
<td>Rename a Lookup Table</td>
<td>71</td>
</tr>
<tr>
<td>Move Lookup Tables</td>
<td>72</td>
</tr>
<tr>
<td>Managing Versions of Lookup Tables</td>
<td>72</td>
</tr>
<tr>
<td>Set the Displayed Version</td>
<td>72</td>
</tr>
<tr>
<td>Create a New Version</td>
<td>72</td>
</tr>
<tr>
<td>Copy the Content of a Version</td>
<td>73</td>
</tr>
<tr>
<td>Delete a Version</td>
<td>73</td>
</tr>
<tr>
<td>Upgrade an Object Version That Is Used in a Decision</td>
<td>74</td>
</tr>
<tr>
<td>Activating Lookup Tables</td>
<td>74</td>
</tr>
<tr>
<td>What Does Activating a Lookup Table Do?</td>
<td>74</td>
</tr>
<tr>
<td>Controlling Where Lookup Tables Are Activated And How They Are Used</td>
<td>75</td>
</tr>
<tr>
<td>Activate a Lookup Table</td>
<td>76</td>
</tr>
<tr>
<td>LOOKUP Function</td>
<td>77</td>
</tr>
<tr>
<td>LOOKUPVALUE Function</td>
<td>78</td>
</tr>
</tbody>
</table>
About Lookup Tables and Functions

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

You can import lookup data from comma-separated-values (CSV) files, such as those created by spreadsheet applications, into lookup tables in SAS Intelligent Decisioning. You can re-import updated CSV files as needed to refresh the lookup tables.

**Note:** SAS Intelligent Decisioning does not support CSV files that contain signature lines.

**Figure 4.1  CSV File Imported Into SAS Intelligent Decisioning**

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

SAS Intelligent Decisioning provides two functions, `LOOKUP` and `LOOKUPVALUE`, that enable you to determine whether a lookup key exists in a lookup table and to retrieve a lookup value from a lookup table.
Predefined Lookup Tables

SAS Intelligent Decisioning defines two lookup tables for you:

Treatment Channels
defines several channels for use with treatments. Channels are the route by which your company and a subject are in contact. The predefined channels include ATM, Agent, Call Center, Mail, Phone, and Web.

Subject Level
defines subject levels for use with treatments. A subject level is the type of account that is associated with a subject. For example, a subject might be an individual customer, a company account, or a household. The predefined levels are Account, Customer, and Household.

Your administrator can modify these tables as described in “Edit Lookup Table Entries” on page 69.

Create a New Lookup Table

1. Click to navigate to the Lookup Tables category view.
2. Click New Lookup Table. The New Lookup Table window appears.
3. Enter a name for the new lookup table. Lookup table names are limited to 250 characters. Lookup table names are case insensitive and must be unique within the database.
4. (Optional) Enter a description for the new lookup table. Descriptions are limited to 1000 characters.
   TIP To modify this description at any time, click next to the Description field on the Properties tab.
5. Click , select the folder where you want to create the new lookup table, and click Save. The application opens the new lookup table and displays the Lookup Table tab.
6. Add entries to the new table either by importing a CSV file or by adding entries manually. See “Import or Refresh Lookup Tables” on page 68 and “Add Lookup Table Entries” on page 69 for more information.
Importing and Exporting Lookup Tables

You can import lookup tables from and export lookup tables to comma-delimited (CSV) files. The format of the CSV file is the same format that is used by the %DCM_IMPORT_LOOKUP macro. For more information, see “Format of the Lookup CSV Input File” in SAS Intelligent Decisioning: Macro Guide.

Note: If you do not need to edit content as a CSV file, you should use either the SAS transfer service or SAS Environment Manager to import and export content. For information about the transfer service, see “Promotion: How to Import (Command-Line Interface)” in SAS Viya Administration: Promotion (Import and Export). For information about using SAS Environment Manager, see “Export Content” in SAS Viya Administration: Folders and “Import a Package to a Folder” in SAS Viya Administration: Folders.

Import or Refresh Lookup Tables

Note: It is recommended that a single lookup table contains no more than 10,000 entries. If your lookup table is very large, it might be necessary to increase the amount of memory specified by the jvm configuration option. See “SAS Intelligent Decisioning Configuration Properties” in SAS Intelligent Decisioning: Administrator’s Guide.

You can import entries into an empty table, and you can refresh an existing lookup table by re-importing the same table.

1. Open the lookup table to which you want to import entries.
2. Click Import. The Import Lookup Table window appears.
3. Click , and select the CSV file that contains the lookup table entries.
4. Select the encoding for the lookup table, and click Import.

Export a Lookup Table

IMPORTANT Do not modify the file structure or the header row in the exported CSV file. You can modify the data values.
To export a lookup table, open the lookup table and click **Export**. The table is exported into a CSV file, and a notification appears in the download bar at the bottom of the browser window.

### Add Lookup Table Entries

**Note:** It is recommended that a single lookup table contains no more than 10,000 entries. If your lookup table is very large, it might be necessary to increase the amount of memory specified by the jvm configuration option. See “SAS Intelligent Decisioning Configuration Properties” in *SAS Intelligent Decisioning: Administrator’s Guide*.

**TIP** You cannot add new entries to a lookup table version that has been activated. In order to edit the table, you must create a new version.

1. Open the lookup table to which you want to add entries.
2. Click **New Entries** if the lookup table is empty, or click **+** if the lookup table already contains entries. The **Add Table Entries** window appears.
3. Enter the lookup key name and value for the new entry. Key names and lookup values are each limited to 100 characters. Key names must be unique within the same lookup table.
   To add additional entries, click **Add an entry**, and enter the new key name and value.
4. Click **Save** to save the new entries and close the **Add Table Entries** window.

### Edit Lookup Table Entries

**TIP** You cannot edit entries in a lookup table version that has been activated. In order to edit the table, you must create a new version.

1. Open the lookup table.
2. Select the entries that you want to edit, and click **편집**. The **Edit Table Entries** window appears.
3. Edit the exiting entries, and click **Save**.
Delete Lookup Table Entries

Open the lookup table, select the entries that you want to delete, and click 🗑️.

Copy a Lookup Table URL

To create a link for external documentation that automatically opens a lookup table in SAS Intelligent Decisioning, complete these steps:

1. Open the lookup table.
2. Click ☑️, and select Copy object URL. The Copy Lookup URL window appears.
3. Click Copy, and then click Close.
   Paste the link into your documentation.

Generate PDF Documentation for a Lookup Table

You can generate detailed documentation for a lookup table as a PDF document. The PDF includes the lookup table properties and a table of the current key-value pairs that are contained in the lookup table.

1. Open the lookup table.
2. Click ☑️, and select Create document. The Create Document window appears.

**TIP** If this option is disabled, there might be unsaved changes. Click 🗑️.

3. (Optional) Enter a name for the document if you do not want to use the default name.
4. Click Create. SAS Intelligent Decisioning creates the PDF. The result depends on your browser’s default download actions. The browser might automatically download the file and display a notification in the bar at the bottom of the browser window. Alternatively, the browser might ask you to choose a download location.
Managing Lookup Tables

Duplicate Lookup Tables

**Note:** If you duplicate a lookup table that has an active version, the duplicate table is automatically activated.

**Note:** You cannot duplicate the predefined Treatment Channels or Subject Level lookup tables.

To duplicate a single lookup table:

1. Select the table that you want to duplicate, click ✂, and select **Duplicate**.
2. Enter a new name for the duplicate lookup table. Names are limited to 250 characters. Lookup table names are case insensitive and must be unique within the database.
3. (Optional) Enter a description for the duplicate table. Descriptions are limited to 1000 characters.
4. Select the version of the lookup table that you want to duplicate.
5. Click 🔎 and select the location where you want to save the duplicate lookup table.
6. Click **Duplicate**.

To duplicate multiple lookup tables, select the tables that you want to duplicate, click ✂, and select **Duplicate**. SAS Intelligent Decisioning duplicates the latest version of the lookup tables and appends an -Copy to the names of the duplicate copies. If needed, a number is also appended to the duplicate table names.

Delete Lookup Tables

Select the tables that you want to delete, click ✂, and select **Delete**.

Rename a Lookup Table

1. Select the table that you want to rename, click ✂, and select **Rename**. The Rename window appears.
2 Enter a new name for the table, and click Rename.

Move Lookup Tables

Note: You cannot move the predefined Treatment Channels or Subject Level lookup tables.

1 Select the tables that you want to move, click , and select Move. The Choose a Location window appears.

2 Select the folder where you want to move the tables, and click OK.

Managing Versions of Lookup Tables

Set the Displayed Version

The displayed version is the version whose information is displayed on the Properties and Lookup Table tabs. On the Versions tab, a ✓ indicates the displayed version. To change the displayed version, select the version that you want to view, and click Set Version. The displayed version is shown in the title bar.

Create a New Version

Note: When you create a new version of a lookup table, the previous version is not locked. The only way to lock a lookup table is to activate it.

Note: You cannot save changes to a version that is locked. If you modify a version that is locked and click ✓, SAS Intelligent Decisioning asks you if you want to replace the current unlocked version with your edited version.

IMPORTANT You cannot unlock a locked version.

To create a new version:

2 Select the version type: Minor or Major. Version numbers follow the format Major.Minor. If you select Major, the number to the left of the period is incremented. If you select Minor, the number to the right of the period is incremented.

3 (Optional) Enter information about the new version in the Notes field.

**TIP** You can edit these notes at any time on the Versions tab.

4 Click Save.

---

### Copy the Content of a Version

You can copy the content of an object’s version in the category view or on the Version tab for the object.

1 In the category view, complete these steps:
   a Select the lookup table whose contents you want to copy.
   b Click , and select Copy version. The Copy Version window appears.
   c Select the version whose contents you want to copy.
   
   Alternatively, on the Versions tab of an object:
   a Select the version whose contents you want to copy.
   b Click , and select Copy version. The Copy Version window appears.

2 Click , and select the target lookup table into which you want to paste the contents of the version.

When you paste the contents, SAS Intelligent Decisioning creates a new version of the target object.

3 Select whether you want to create a new major or minor version of the target object.

4 (Optional) Modify the notes that will be associated with new version of the target object.

5 Click Paste Version, and then click Yes.

---

### Delete a Version

*Note:* In order to be able to delete a specific version of a lookup table, you must have permission to delete the lookup table itself.
On the Versions tab, select the version that you want to delete, click \( \text{Delete} \), and select Delete.

You cannot delete the current version.

**Upgrade an Object Version That Is Used in a Decision**

If you create a new version of an object that is already used in a decision, you can upgrade the decision to use the new version.

1. On the Versions tab for the object, click \( \text{Upgrade decisions} \), and select Upgrade decisions. The Upgrade Decisions window appears. This window lists all of the decisions that include the object.

2. In the Version to upgrade to field, select the version of the object to which you want to upgrade the decisions.

3. Select the check boxes for the decisions that you want to upgrade, and click Upgrade Decisions.

**Activating Lookup Tables**

**What Does Activating a Lookup Table Do?**

When you activate a lookup table, the lookup table is converted into executable code in the form of a SAS Cloud Analytic Services (CAS) format. If your site has a SAS Micro Analytic Service publishing destination, then a SAS Micro Analytic Service module is also created for the table. The CAS format is used when you create and run tests and scenario tests (pre-publishing tests) for the rule sets and decisions that use the lookup tables. The CAS format and SAS Micro Analytic Service module also enable objects that are published to those destination types to use the most recently activated version of a lookup table instead of a static copy.

Teradata and Hadoop destinations do not support the format or module that is created when you activate a lookup table. Objects that are published to these destinations must include a static version of each lookup table.

You must activate a version of any lookup table that is used in a rule set or decision. When you activate a version of a lookup table, that version is locked and cannot be edited. Each lookup table can have only one active version.

The active version of a lookup table is used in the following ways:

- when you run a test or scenario test prior to publishing an object
to generate a static copy of the table when you publish an object if a static copy is needed, such as when you publish the object to Teradata or Hadoop destinations.

- by published objects in CAS and SAS Micro Analytic Service destinations when those objects do not include static copies

Controlling Where Lookup Tables Are Activated And How They Are Used

Your administrator can use the following configuration options to control where lookup tables are activated and how they are used:

`sas.referencedata.publish.lookupDisableMasPublish = ON | OFF`

controls whether lookup tables are activated in SAS Micro Analytic Service destinations. When this option is set to OFF and you activate a lookup table, it is activated in all of the SAS Micro Analytic Service destinations that are defined at your site. If this option is set to ON, then lookup tables are not activated in these destinations, and your administrator must set the `sas.businessrules.lookupStaticBinding` option to ON in order to include static copies of lookup tables in generated code and ensure that published content executes correctly.

`sas.businessrules.lookupStaticBinding = ON | OFF`

controls whether a static copy of the active version of a lookup table is included in the generated code for an object when that object is published. When this option is set to ON, a static copy of the active version of any lookup table that is used in an object is included in the generated code when those objects are published. When this option is set to OFF, a published object’s use of lookup tables depends on the destination to which they were published.

- Objects that are published to SAS Cloud Analytic Services (CAS) and SAS Micro Analytic Service destinations use the most recently activated version of lookup tables.

- Objects that are published to Teradata and Hadoop destinations cannot use lookup tables.

The following table describes how these two options work together for each destination type.
### Table 4.1 Configuration Settings for Lookup Tables

<table>
<thead>
<tr>
<th>Configuration Settings</th>
<th>Destinations Where Lookup Tables Are Activated And Which Version is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>sas.referencedata.</td>
<td>Lookup tables are activated in all destinations. Published objects in CAS and SAS Micro Analytic Service destinations use the most recently activated versions of lookup tables. Published objects in Hadoop and Teradata destinations cannot use lookup tables. This configuration is the default configuration.</td>
</tr>
<tr>
<td>publish.</td>
<td></td>
</tr>
<tr>
<td>lookupDisableMas</td>
<td></td>
</tr>
<tr>
<td>Publish</td>
<td></td>
</tr>
<tr>
<td>sas.businessrules</td>
<td></td>
</tr>
<tr>
<td>lookupStaticBinding</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>Lookup tables are activated in all destinations. However, published objects in all destinations use static copies of the lookup table versions that were active when the object was published.</td>
</tr>
<tr>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>Lookup tables are activated in all destinations except SAS Micro Analytic Service destinations. Published objects in SAS Micro Analytic Service destinations cannot use lookup tables. Published objects in CAS destinations use the most recently activated versions of lookup tables. Published objects in Hadoop and Teradata destinations cannot use lookup tables.</td>
</tr>
<tr>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td>Lookup tables are activated in all destinations except SAS Micro Analytic Service destinations. Published objects in all destinations, including SAS Micro Analytic Service destinations, use static copies of the lookup table versions that were active when the object was published.</td>
</tr>
</tbody>
</table>

**Note:** In multi-tenant environments, each tenant has its own lookup tables that are unique to that tenant.

For more information, see "Configuration Properties" in *SAS Intelligent Decisioning: Administrator’s Guide*.

### Activate a Lookup Table

1. Open the lookup table that you want to activate.
2. On the **Versions** tab, select the version that you want to activate, and click **Set Version** to set it as the displayed version. See "Set the Displayed Version" on page 72 for more information.
TIP  You can also activate a lookup table on the Properties tab. The displayed version is activated.

3  Click Activate.

LOOKUP Function

Determines whether a lookup key exists in a lookup table.

Restrictions:
- You can specify the LOOKUP function in condition expressions only.
- If an expression contains the LOOKUP function, then the expression cannot contain anything else.

Returned data type:
- Boolean

Syntax

LOOKUP ('lookup_table_name', variable_or_value)

lookup_table_name
- specifies the name of the lookup table that you want to search.

variable_or_value
- specifies either the literal key value or a variable that contains a lookup key value.

Example

Suppose you have a Country_Codes lookup table that uses two-letter abbreviations for countries as the lookup key and country names as the lookup values.

<table>
<thead>
<tr>
<th></th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AU</td>
<td>Australia</td>
</tr>
<tr>
<td>2</td>
<td>BR</td>
<td>Brazil</td>
</tr>
<tr>
<td>3</td>
<td>CA</td>
<td>Canada</td>
</tr>
<tr>
<td>4</td>
<td>CR</td>
<td>Costa Rica</td>
</tr>
</tbody>
</table>

To verify that the value of the variable Cntry_Key exists as a lookup key in the table Country_Codes, you can use the following expression:

LOOKUP('Country_Codes',Cntry_Key)

If the value of Cntry_Key exists as a lookup key, the LOOKUP function returns the value True.
In the following rule, if the key specified by the variable Cntry_Key exists in the lookup table Country_Codes, then the value that is associated with that key is assigned to the variable Country_Name.

**LOOKUPVALUE Function**

Retrieves a lookup value from a lookup table.

**Restrictions:**
- You can specify the LOOKUPVALUE function only in action expressions.
- If an expression contains the LOOKUPVALUE function, then the expression cannot contain anything else.

**Returned data type:**
Lookup tables are stored as character data. However, you can assign the results of the LOOKUPVALUE function to the following types of variables: Character, Integer, Decimal, Date, Datetime, or Boolean. The LOOKUPVALUE function converts the results to match the type of the variable.

**Syntax**

```
LOOKUPVALUE (lookup_table_name, variable_or_value)
```

- **lookup_table_name**
  - specifies the name of the lookup table that you want to search.

- **variable_or_value**
  - specifies either the literal key value or a variable that contains the lookup key value.

**Example**

Suppose you have a Country_Codes lookup table that uses two-letter abbreviations for countries as the lookup key and country names as the lookup values. The Country_Codes lookup table contains the lookup key **CA**, and the lookup value that corresponds to that key is **Canada**.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AU</td>
<td>Australia</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
</tr>
<tr>
<td>CR</td>
<td>Costa Rica</td>
</tr>
</tbody>
</table>
If the Cntry_Key variable in the current input record contains the value CA, you can use the following expression to retrieve the lookup value that is associated with that key from the table Country_Codes:

```plaintext```
LOOKUPVALUE('Country_Codes', Cntry_Key)
```

In the following rule, if the key specified by the variable Cntry_Key exists in the lookup table Country_Codes, then the value that is associated with that key is assigned to the variable Country_Name.

```
IF Cntry_Key
THEN LOOKUPVALUE Cntry_Name
```
Using Custom Code Files

Introduction to Custom Code Files

You can define custom code files to do things that are not possible in rules, models, or treatments. For example, you can define a code file that makes HTTP calls to REST APIs, interacts with a database, manipulates files in the file system, or performs custom data transformations.
IMPORTANT  You are responsible for ensuring that only users who are authorized to do so can install Python libraries, develop, and test code nodes that use SQL or Python, and execute decisions that use code nodes.

IMPORTANT  All changes that you make to a code file affect all unpublished decisions that use that code file. For more information, see "Content That Is Used by Decision Tests" on page 125 and "Content Executed by Published Decisions" on page 137.

Note: Not all of the code that you can write in a custom code file will work in all publishing destinations or in decision tests.

For information about adding a custom code file to a decision, see "Adding Objects to a Decision" on page 101.

DS2 Code Files

Rules for Creating DS2 Code Files

When you are developing your DS2 package, follow these rules:

- Do not change the package name in the PACKAGE statement:
  ```
  package "$\{PACKAGE\_NAME\}" /inline;
  ```
  This line must appear exactly as shown and must begin in column one of the code file. The token is replaced with a package name that SAS Intelligent Decisioning uses to maintain the relationship between the code file and the decisions that use it.

- Do not specify any DS2 options (DS2\_OPTIONS statement) in your package code.

- Custom DS2 code files in SAS Intelligent Decisioning support only three data types: double, varchar, and package datagrid variables.

  These data type names are case sensitive. List input-only parameters first. List input-output and output-only parameters next, using the in\_out modifier. Do not specify a length for input-output and output-only parameters. The length for these parameters is derived from the variables that are passed into the method. For example, the following lines define the variables LOAN, REASON, and ASSETS.

  ```
  double "loan"
  in\_out varchar "reason"
  package datagrid "assets"
  ```
IMPORTANT Before you validate a DS2 code file that uses a data grid package, verify that the declaration statements for the package specify `datagrid` instead of `dcm_datagrid`. For example:

```sas
in_out package datagrid variable;
dcl package datagrid variable();
```

Note: Packages are input-output data types.

- Your package must define an EXECUTE method.
- The variables that you use as parameters for the EXECUTE method in your code must be defined as variables in your decision.
- If you use a DS2 SQLSTMT package in your code, do not run a publishing validation test in SAS Cloud Analytic Services (CAS). Instead, validate your logic in a SAS Micro Analytic Service destination. Use a published name that is not the same as the final name that you will use to publish your decision.

For information about developing DS2 packages, processing data grids, and the SAS APIs, see the following:

- SAS DS2 Programmer’s Guide
- SAS DS2 Language Reference
- SAS Intelligent Decisioning: Using Data Grids
- http://developer.sas.com

---

Example: Custom DS2 Code File

The following DS2 package sends a request to the external API `http://helloacm.com/api/fortune`. This API returns a character string that contains escaped characters. The GETFORTUNE method uses the DS2 function TRANSTRN to modify these characters.

```sas
package "${PACKAGE_NAME}" /inline;
dcl package http fortune_pkg();
dcl varchar(1048576) character set utf8 http_response;
dcl int rc;
dcl int getMethodDefined;

method execute(in_out varchar "fortune");
    if missing(getMethodDefined) then do; /* establish GET method */
        fortune_pkg.createGetMethod('http://helloacm.com/api/fortune');
        "getMethodDefined" = 1;
    end;
```
"fortune" = 'Gloom and Doom'; /* default fortune */
fortune_pkg.executeMethod();
fortune_pkg.getResponseBodyAsString(http_response, rc);
if (rc = 0) then do; /* clean up escaped characters in the response */
    "fortune" = transtrn(http_response, '\n', ' ');
    "fortune" = transtrn("fortune", '\t', ' ');
    "fortune" = transtrn("fortune", '\"', '"');
end;
end;
endpackage;

Testing DS2 Code Files in SAS Studio

Note: If your custom code uses data grid variables, see “Working with Data Grids in SAS Studio” in SAS Intelligent Decisioning: Using Data Grids.

Replace the placeholder name ${PACKAGE_NAME} with a valid DS2 package name.

To test your package in a separate DS2 invocation from where the calling program is running, replace the inline modifier with overwrite=yes.

proc ds2;
    package "testCustomCode" /overwrite=yes;
    method execute(double l, double w, double h, double d, 
in_out double vol, in_out double wgt);

        vol = l * w * h;
        wgt = vol * d;
    end;
endpackage;
run;
quit;

In the following example, the package is compiled in the same PROC DS2 invocation as the DATA step that instantiates the package, so the code specifies the /inline package modifier.

/* Create test data. */
data work.testdata;
    length material $13;
    long=40; wide=20; high=10; density=0.098; material = 'aluminum'; output;
    long=20; wide=10; high=4;  density=0.284; material = 'iron';     output;
run;
proc ds2;

    /* Replace the placeholder name with a package name. */
    package "testCustomCode" /inline;
    method execute(double l, 
    double w, 
    double h, 
    double d, 
    in_out double vol, 
    in_out double wgt);
\[
\text{vol} = l \times w \times h;
\]
\[
\text{wgt} = \text{vol} \times d;
\]
end;
endpackage;

\[
\text{\texttt{/* Use a DATA step to execute the custom code. */}}
\]
data _null_;  
dcl package testCustomCode myCustomCode(); /* Instantiate the package. */  
dcl double volume;  
dcl double weight;  
method run();  
  /* Read in the variables long, wide, high, and density. */  
  set work.testdata(drop=(material));  
  volume = .;  
  weight = .;  
  myCustomCode.execute(l, w, h, d, v, wgt);  
  put _all_;  
end;  
enddata;  
run;  
quit;

---

**Python Code Files**

**IMPORTANT**  You can publish decisions that contain Python code files only to SAS Micro Analytic Service and SAS Cloud Analytic Services (CAS) destinations.

When you publish a decision that contains a Python code file, SAS Intelligent Decisioning generates a private DS2 PyMAS package. The package is assigned a random name. Your Python program is encapsulated inside a DS2 EXECUTE method. When the decision is executed, the DS2 process sends the Python program to a Python process to be executed.

When you are developing your Python code, follow these rules:

- Your Python code must define an EXECUTE function. This function is the only public function allowed.

- A doctoring is required and must begin with `Output:`. This string must immediately follow the Python EXECUTE function declaration, and it must be indented within the EXECUTE function definition. In the docstring, list all of the output variables produced by the program. For example, if your program has two output variables named `prediction` and `probability`, your docstring would appear as follows:

  ```
  "Output: prediction, probability"
  ```

- Optional input arguments to the Python EXECUTE function are not supported.
You must specify the data types for the input and output variables for the EXECUTE function in your Python code on the Variables tab of the decision. The DS2 package code uses this information to resolve the data types for the variables. These variables are specified in the signature of the DS2 EXECUTE method. If you do not specify the variables on the Variables tab, SAS Intelligent Decisioning creates decision variables with a data type of Unknown. Before you can save the decision, you must edit the decision variables and correct the data types.

The Python EXECUTE function must return standard Python data types. For more information, see “Return Values” in SAS Micro Analytic Service: Programming and Administration Guide.

You can import libraries and define other functions and classes that are used by the Python EXECUTE function.

Test your Python code by using a Python interpreter that is outside of SAS Intelligent Decisioning before you incorporate your code into a decision.

After you test your code outside of SAS Intelligent Decisioning, incorporate it into a decision and test the decision. In order to run a test for a decision that contains a Python code file, you or an administrator must configure support for Python. See “Configuring Support for Python Code Files” in SAS Intelligent Decisioning: Administrator’s Guide for more information.

For more information about using Python with DS2, see the following:


Data Query Files

Using Data Query Files

Data query nodes always return a data grid as output. For information about data grids, see SAS Intelligent Decisioning: Using Data Grids.

**IMPORTANT** You can publish decisions that contain data query files only to SAS Micro Analytic Service destinations.

In order to test decisions that contain data query files, an administrator must configure support for SQL query files. See “Configuring Support for Data Query Files” in SAS Intelligent Decisioning: Administrator’s Guide.

**IMPORTANT** Tests that are running in SAS Intelligent Decisioning might encounter a significant performance impact. Pre-publish testing requires SAS
Intelligent Decisioning to convert the SQL code to use HTTP protocol and instantiate a MAS module for each call to the SQL package. These actions incur a significant performance impact and create a limitation on the size of the results table. When you test a decision that uses a data query node, use an input data set that is as small as possible.

Developing SQL Code

When you are developing your SQL code, follow these rules:

- Data query nodes support SELECT, INSERT, UPDATE, and DELETE statements. They do not support any data definition language (DDL) statements such as ALTER or DROP that alter the structure of the table.

- Data query nodes support only the following data types: decimal, string, date, datetime, and integer. You must specify a length for string variables. You cannot specify a length for other variable types. Specify input variables with a question mark (?). Use the AS keyword to specify the input and output variables in the decision as aliases for the database column names. Enclose the decision variable specifications in braces ({}). For example, in the following SELECT statement, DEBTINC, REASON, and BAD are decision variables. The variable BAD is an input variable in the decision. This statement retrieves the values of the DEBTRATIO, CAUSE, TIMESTAMP, and BADLOAN columns from the database and assigns their values to the decision variables DEBTINC, REASON, TS, and BAD.

```sql
SELECT debtRatio AS {:debtinc:decimal},
       cause AS {:reason:string:8},
       timestamp AS {:ts:datetime}
FROM hmeq_test WHERE badloan = {?:bad:decimal}
```

The following INSERT statement adds a row that contains the columns NAME and AMT to the table MYTABLE:

```sql
INSERT INTO mytable {?:name:string:1000},{?:amt:integer}
```

**Note:** You can read data from native SQL date and datetime variables. However, do not rely on the accuracy of data that contains fractional seconds.

- Do not use an asterisk (*) to select database columns. SAS Intelligent Decisioning does not have any metadata about the table, so it cannot determine what columns are in the table.

- Database column names that contain multi-byte characters, spaces, or special characters such as colons and dashes are not supported in data query nodes.

- When you create a new data query file, SAS Intelligent Decisioning adds the comment /* include sqlReturnInfo */ to the file. When this comment is present and the query selects data, SAS Intelligent Decisioning generates variables for the return code, row count, and query output. For more information, see “Decision Variables for Data Query Files” on page 88.

For syntax information about SQL statements, see “FedSQL Statements” in SAS FedSQL Language Reference.
For information about the functions that are supported in SQL code, see “FedSQL Functions” in SAS FedSQL Language Reference.


## Decision Variables for Data Query Files

When you create a data query file, SAS Intelligent Decisioning automatically adds the comment /* include sqlReturnInfo */. When this comment is present and the query selects data, SAS Intelligent Decisioning creates three query file variables. When you add the query file to a decision, SAS Intelligent Decisioning creates decision variables and maps the query file variables to the decision variables, as shown in the following table.

<table>
<thead>
<tr>
<th>Query File Variable</th>
<th>Decision Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file_name_returnCode_out</td>
<td>returnCode</td>
<td>The return code that is generated by the query file.</td>
</tr>
<tr>
<td>file_name_rowCount_out</td>
<td>rowCount</td>
<td>The number of rows returned by the query.</td>
</tr>
<tr>
<td>file_name_out</td>
<td>dgo</td>
<td>The data grid variable that contains the output of the data query.</td>
</tr>
</tbody>
</table>

If the comment is present but the query does not select data, SAS Intelligent Decisioning creates only the return code variables.

If you remove the comment and the query selects data, only the output variables are created.

If you remove the comment and the query does not select data, no variables are created.

## Using Macro Variables in SAS Studio

When you author a data query file in SAS Studio, you can pass input data to the decision by using macro variables. When you add the data query file to a decision, SAS Intelligent Decisioning creates decision variables that have the same name as the macro variables. When the decision executes, the values of the decision variables are set to the same values as the macro variables.

To define the macro variable in SAS Studio, open a SAS Program tab and submit the %LET statement. For example, the following statement defines the macro variable &myName:

```sas
%let myname = 'Arinya';
```
To use the macro variable in the **Query** tab, reference it with an ampersand followed by the macro variable name:

```
SELECT name FROM table WHERE name = &myname
```

When you create a filter expression, set the appropriate options so that SAS Studio recognizes the macro variable. The settings depend on whether the variable is character or numeric.

- If the macro variable is numeric, check the **Allow macros** check box.
- If the macro variable is character, check the **Match Case** check box, and clear the **Quote Strings** check box.

You can use custom SQL code for your data query instead of using a quick filter. However, SAS Intelligent Decisioning cannot determine the variable’s data type when it is creating the decision variable. SAS Intelligent Decisioning assumes that the variable is a numeric unless its name ends with `char` or `string`. The `char` and `string` suffixes are not case sensitive.

For more information, see “Filtering Data” in *SAS Studio: User’s Guide* and “Macro Variables Defined by Users” in *SAS Macro Language: Reference*.

---

### Create a New Code File

1. Click to navigate to the Code files category view.
3. Enter a name for the new code file.

   ![Note](image)

   **Note:** When a data query file generates output, the file name can be up to 28 characters long. When the data query file is added to a decision, SAS Intelligent Decisioning creates an output-only decision variable. The name of the decision variable is the data query file name plus `_out`.

4. Select the code file type.
5. (Optional) For data query file types, select the editor with which you want to edit the file.

   ![TIP]

   **TIP** For data query files that you choose to edit in SAS Studio, SAS Intelligent Decisioning adds `.cgy.df` to the file name, and the file type is displayed as **Data Query**. For data query files that you choose to edit in the SQL editor, the file type is displayed as **SQL**.

6. (Optional) Enter a description for the file. Descriptions are limited to 1000 characters.

   ![TIP]

   **TIP** You can edit the description at any time on the **Properties** tab.
7 Click  and select the folder where you want to save the file.

8 Click **Save**. SAS Intelligent Decisioning opens the new code file.

For data query files that you choose to edit with SAS Studio, SAS Intelligent Decisioning displays the **Properties** tab, and you can click **Open Data Query** to open the file in SAS Studio.

For all other file types, SAS Intelligent Decisioning opens the file in the appropriate code editor.

---

### Validate SQL or DS2 Code Files

To validate a code file of type SQL or DS2, open the file, and then click **Validate**. For DS2 code files, SAS Intelligent Decisioning compiles the code.

**IMPORTANT** Before you validate a DS2 code file that uses a data grid package, verify that the declaration statements for the package specify `datagrid` instead of `dcm_datagrid`. For example:

```sql
in_out package datagrid variable;
dcl package datagrid variable();
```

For SQL query files, SAS Intelligent Decisioning runs the query in order to verify the SAS Micro Analytic Service connection string, scans the SQL SELECT statement for syntax errors, and looks for references to non-existent tables and columns.

---

### Managing Code Files

#### Duplicate Code Files

**Note:** You cannot duplicate a code file if it is open.

To duplicate a single code file:

1 In the Code Files view, select the code file that you want to duplicate.

2 Click  and select **Duplicate**. The Duplicate Code Files window appears.

3 Enter a new name for the duplicate code file.

4 (Optional) Enter a description for the code file.
5 Click ☐, and select the folder where you want to save the duplicate file.

6 Click Duplicate.

To duplicate multiple code files:

1 In the Code files view, select the code files that you want to duplicate.

2 Click and select Duplicate. SAS Intelligent Decisioning duplicates the code files and appends _Copy to the names of the duplicate copies. If needed, a number is also appended to the names of the duplicate copies.

Delete Code Files

In the Code Files category view, select the code files that you want to delete, click ☑, and select Delete.

Rename a Code File

Note: You cannot rename a code file if it is open.

1 In the Code Files category view, select the code file that you want to rename.

2 Click ☐ and select Rename. The Rename window appears.

3 Enter a new name for the code file, and click Rename.

Move Code Files to a Different Folder

1 In the Code Files category view, select the code files that you want to move.

2 Click ☐ and select Move. The Choose a Location window appears.

3 Select the location to which you want to move the code files, and click OK.
# Working with Decisions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Decisions</td>
<td>94</td>
</tr>
<tr>
<td>Create a Decision</td>
<td>94</td>
</tr>
<tr>
<td>Views for Editing a Decision</td>
<td>95</td>
</tr>
<tr>
<td>The Decision Flow Tab versus the Decision Tab</td>
<td>95</td>
</tr>
<tr>
<td>Yes/No Branches and Condition Nodes</td>
<td>95</td>
</tr>
<tr>
<td>Controlling the Tab Display</td>
<td>96</td>
</tr>
<tr>
<td>Managing the Variables in a Decision</td>
<td>97</td>
</tr>
<tr>
<td>About Variables</td>
<td>97</td>
</tr>
<tr>
<td>Add Variables from a Data Table</td>
<td>99</td>
</tr>
<tr>
<td>Add Variables from a Rule Set or Decision</td>
<td>99</td>
</tr>
<tr>
<td>Create Custom Variables on the Variables Tab</td>
<td>100</td>
</tr>
<tr>
<td>Duplicate a Variable</td>
<td>100</td>
</tr>
<tr>
<td>Delete Variables</td>
<td>101</td>
</tr>
<tr>
<td>Edit Variable Properties</td>
<td>101</td>
</tr>
<tr>
<td>Edit Metadata for Data Grid Variables</td>
<td>101</td>
</tr>
<tr>
<td>Determine Which Objects Use a Particular Variable</td>
<td>101</td>
</tr>
<tr>
<td>Adding Objects to a Decision</td>
<td>101</td>
</tr>
<tr>
<td>Add an Existing Object</td>
<td>101</td>
</tr>
<tr>
<td>Add a New Code File or Data Query File</td>
<td>102</td>
</tr>
<tr>
<td>Adding Record Contacts Nodes</td>
<td>103</td>
</tr>
<tr>
<td>Adding Branches to a Decision</td>
<td>105</td>
</tr>
<tr>
<td>Mapping Variables within a Decision</td>
<td>112</td>
</tr>
<tr>
<td>About Decision Variables and Mapping</td>
<td>112</td>
</tr>
<tr>
<td>Scoring Rows in a Data Grid</td>
<td>114</td>
</tr>
<tr>
<td>Mapping Data Grid Variables</td>
<td>114</td>
</tr>
<tr>
<td>Map Object Variables to Decision Variables</td>
<td>114</td>
</tr>
<tr>
<td>Editing the Objects in a Decision</td>
<td>115</td>
</tr>
<tr>
<td>Open an Object from within a Decision</td>
<td>115</td>
</tr>
<tr>
<td>Edit the Properties of a Decision Node</td>
<td>116</td>
</tr>
<tr>
<td>Reorder Objects</td>
<td>116</td>
</tr>
<tr>
<td>Delete an Object</td>
<td>116</td>
</tr>
<tr>
<td>Copy a Decision URL</td>
<td>117</td>
</tr>
<tr>
<td>Generate PDF Documentation for a Decision</td>
<td>117</td>
</tr>
</tbody>
</table>
About Decisions

A decision enables you to combine rule sets, analytical models, treatment groups, code files, record contacts nodes, and conditional logic into a single process. You can also add a decision to another decision.

Create a Decision

1. Click  to navigate to the Decisions category view.
2. Click New Decision. The New Decision window appears.
3. Enter a name for the decision if you do not want to use the default name. Decision names are limited to 100 characters and must be unique within a folder.
4. (Optional) Enter a description for the new decision. Descriptions are limited to 1000 characters.
5 Click  and select the folder where you want to save the decision.

6 Click Save. SAS Intelligent Decisioning opens the new decision and displays the Decision Flow tab.

7 Add objects, and if needed, add variables to the decision. For more information, see “Adding Objects to a Decision” on page 101 and “Managing the Variables in a Decision” on page 97.

Views for Editing a Decision

The Decision Flow Tab versus the Decision Tab

There are two tabs on which you can view and edit decisions.

- On the Decision Flow tab, you can edit a decision by using a graphical editor.
  To add objects to a decision, right-click on a node to display a menu or click  or . To choose objects that have already been defined by navigating to their location, click . To add branches, record contacts nodes, or custom code files by selecting the object type, click .

- On the Decision tab, you can edit the decision by using a tabular view similar to the rule set editor.
  Condition nodes on the Decision tab are equivalent to Yes/No branches on the Decision Flow tab. You can add conditions to a decision, but the Decision tab does not support Equals, Range, or LIKE branches. If you add one of these branch types to your decision, the Decision tab is disabled.

Save your work before switching tabs. Click  to refresh the view in a tab.

Yes/No Branches and Condition Nodes

A Yes/No branch on the Decision Flow tab corresponds to a condition (an IF statement) on the Decision tab. Objects in a THEN clause on the Decision tab correspond to nodes in a Yes branch path on the Decision Flow tab. Objects in an ELSE clause correspond to nodes in a No branch path.

For example, suppose you have the following nodes on the Decision Flow tab:
The same nodes appear on the **Decision** tab as an IF-THEN-ELSE statement:

On the **Decision Flow** tab:
- Click ![properties](image) to open the properties pane for the selected object. Click ![hide](image) to hide the properties pane.
- Click ![overview](image) to open the diagram overview. The diagram overview is a scaled-down version of the entire diagram. You can pan across the diagram and position the view over different sections of the diagram. The section that is currently visible on the screen is outlined. The overview is useful when a decision diagram is too large to display all of the nodes on one screen.

On the **Decision** tab:
- Click ![collapse](image) or ![expand](image) to collapse or expand a single node in the decision.
- Click ![bulk collapse](image) or ![bulk expand](image) to collapse or expand all of the nodes in the decision.
Managing the Variables in a Decision

About Variables

The Properties of a Variable

Note: For information about data grid variables, see “Using Data Grids in SAS Intelligent Decisioning” in SAS Intelligent Decisioning: Using Data Grids.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Variable names must start with a letter or an underscore (_), and they can contain only alphanumeric characters and the underscore. They can be up to 32 characters long and must be unique within a rule set.</td>
</tr>
<tr>
<td></td>
<td>Note: SAS Intelligent Decisioning does not support double-byte character set (DBCS) characters in variable names.</td>
</tr>
<tr>
<td></td>
<td>Note: Do not use any of these operators or keywords as variable names: AND, OR, IN, NOT, LIKE, TRUE, or FALSE. Do not use <em>N</em> or any DS2 reserved word as a variable name. See “Reserved Words in the DS2 Language” in SAS DS2 Programmer’s Guide for information about reserved words in the DS2 language.</td>
</tr>
<tr>
<td>Data type</td>
<td>SAS Intelligent Decisioning supports the following data types: Boolean, character, data grid, date, datetime, decimal, and integer. For Boolean values, enter True or 1 and False or 0. If you specify True or False, SAS Intelligent Decisioning represents Boolean values using the numbers 1 and 0 in the code that it generates.</td>
</tr>
<tr>
<td></td>
<td>Note: For content that will be published to a SAS Micro Analytic Service destination, the values of date and datetime input variables must be numeric. They cannot be formatted values such as 18jul2019.</td>
</tr>
<tr>
<td>Input and Output</td>
<td>A variable can be an input variable, an output variable, both, or neither (a temporary variable). See “Input Variables, Output Variables, and Temporary Variables” on page 98 for more information.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Length     | For character variables that are input-only variables or that are both input and output variables, the length is set to the larger of the following two values:  
  - the length of the column to which the variable is mapped when the decision is run  
  - the default length of 100 characters  
  The maximum length for character variables is 32767.  
  The length for Boolean and numeric variable types is set automatically. |
| Initial value | You can specify an initial value for all data types except data grids. Initial values are used only at run time and only for output-only variables.                                                              |
| Description | Descriptions are limited to 256 characters.                                                                                                                                                                 |

**Input Variables, Output Variables, and Temporary Variables**

For each variable used in a decision, you must specify whether the variable is an input variable, an output variable, both an input and an output variable, or a temporary variable.

- Input variables are variables that are present in the input table for a decision. When a decision is deployed in a production system, all input variables must be mapped to table columns in input data. When you test a decision in SAS Intelligent Decisioning, for each input variable, you must either map it to a table column or specify a constant as its input value. When you create or edit a variable, clear the **Input** check box for any variable that you do not want to be mapped to a column in an input table or for which you do not want to specify a value.

- Output variables are variables that are written to the output table that is created when a decision is run. When you create or edit a variable, clear the **Output** check box for any variable that you want to exclude from the output data.

- Temporary variables are variables that are not present in the input data and they are not written to the output table. To create a temporary variable for use only while a decision is executing, clear both the **Input** and **Output** check boxes.

When you create a new variable, it is created as both an input and output variable by default.
Add Variables from a Data Table

1. On the Variables tab, click Add variable and select Data table. The Choose Data window appears, and the list of SAS Cloud Analytic Services (CAS) tables that are loaded into memory is displayed on the Available tab.

   If the table that you need does not appear in the list of available tables, try the following solutions:
   - If the table appears on the Data Sources tab, right-click on the table, and select Load to load the table into memory. If the table does not appear on the Available tab, click .
   - If the table does not appear on the Data Sources tab, import the data. The process of importing the data loads it into memory. For information about importing data from different sources, see “Making Data Available to CAS” in SAS Data Explorer: User’s Guide.

2. Select the table from which you want to import variables, and click OK. The Add Variables window appears.

3. Select the variables that you want to import and click . To import all of the variables in the table, click .

4. Click Add to add the select variables, or click Add and replace to replace existing variables that have the same name.

5. On the Variables tab, select or clear the Input and Output check boxes as necessary. See “Input Variables, Output Variables, and Temporary Variables” on page 98 for more information.

Add Variables from a Rule Set or Decision

1. On the Variables tab, click Add variable, and select Rule set or Decision. The Choose an Item window appears.

2. Select the rule set or decision from which you want to import variables, and click OK. The Add Variables window appears.

3. Select the variables that you want to import and click . To import all of the variables in the table, click .

4. Click Add to add the select variables, or click Add and replace to replace existing variables that have the same name.

5. On the Variables tab, select or clear the Input and Output check boxes as necessary. See “Input Variables, Output Variables, and Temporary Variables” on page 98 for more information.
Create Custom Variables on the Variables Tab

Note: For information about data grid variables, see “Defining Data Grid Variables” in SAS Intelligent Decisioning: Using Data Grids.

To create custom variables on the Variables tab:

1. Click Add variable and select Custom variable. The Add Variables window appears.

2. Complete these steps for each variable that you want to add:
   a. Enter the name of the new variable, and select the data type of the variable. See “The Properties of a Variable” on page 97 for additional information.
   b. (Optional) Click Optional to display the Description, Initial value, and Length fields.
   c. (Optional) Enter a length, initial value, and description for the new variable. See “The Properties of a Variable” on page 97 for additional information.
   d. Click Add. SAS Intelligent Decisioning adds the new variable to the table of variables. By default, variables are added to the table as both input and output variables.
   e. (Optional) Clear the check boxes in the Input or Output columns.
      - Clear the Input check box for any variable that you do not want to be mapped to a column in an input table or for which you do not want to specify a value.
      - Clear the Output check box for any variable that you want to exclude from the output data.
      - Clear both the Input and Output check boxes to create a temporary variable.

3. Click OK to add the variables and close the Add Variables window.

Duplicate a Variable

1. On the Variables tab, select the variable that you want to duplicate, click , and select Duplicate. The Duplicate Variable window appears.

2. Enter a new name for the duplicate variable.

3. (Optional) Enter a description for the variable.

4. Click Duplicate.
Delete Variables

On the **Variables** tab, select the check box for the variables that you want to delete, click , and select **Delete**.

**Note:** You cannot delete a variable if it is used in a rule set or decision.

Edit Variable Properties

On the **Variables** tab, click on the variable name of the variable that you want to edit. The **Edit Variable** window appears. Edit the properties as needed, and then click **OK**. See “The Properties of a Variable” on page 97 for additional information.

Edit Metadata for Data Grid Variables

For information, see “Editing Data Grid Variable Metadata” in SAS Intelligent Decisioning: Using Data Grids.

Determine Which Objects Use a Particular Variable

On the **Variables** tab, select the check box for the variable, click , and select **Used by**. The **Variable Used By** window appears. This window lists the objects that use the selected variable.

Adding Objects to a Decision

Add an Existing Object

1. On the **Decision Flow** tab, click to choose objects that have already been defined by navigating to their location.

   Alternatively, on the **Decision** tab, click **Add** and select the object type that you want to add to the decision. If an object in the decision is already selected, select
Edit ▷ Add below, and select the object type. The Content Selection window appears.

2 On the Decision Flow tab, drag the object from the list of objects onto the diagram where you want to add it. Alternatively, on the Decision tab, select the object that you want to add.

3 (Optional) Reorder the objects in the decision. For more information, see “Editing the Objects in a Decision” on page 115.

4 (Optional) For rule sets and subdecisions, in the Properties pane, select the version of the rule set or subdecision that you want to include in the decision.

IMPORTANT You cannot add a decision as a subdecision within itself.

TIP You can change the version at any time by changing the version on the Properties pane. See “Edit the Properties of a Decision Node” on page 116 for more information.

Note: You cannot specify the version of a model, treatment group, or code file that is used in a decision. For more information, see “Content That Is Used by Decision Tests” on page 125 and “Content Executed by Published Decisions” on page 137.

5 (Optional) On the Properties tab, select the Subject ID variable and the Subject level variable that you want to associate with subject contact records that are recorded by the decision.

TIP If you do not specify the subject ID or subject level, the subject ID or subject level is recorded as NONE in the subject contact history.

For more information about subjects and subject contact histories, see the following topics:

- “About Treatments and Decisions” on page 48
- "Example: A Decision That Includes a Treatment Group” on page 49
- "Adding Record Contacts Nodes” on page 103
- “Predefined Lookup Tables” on page 67

6 Map the object’s variables to decision variables. For more information, see “Mapping Variables within a Decision” on page 112.

Add a New Code File or Data Query File

See Chapter 5, “Using Custom Code Files,” on page 81 for information about the content of custom code files. For information about the content that is used when a
decision is executed, see “Content That Is Used by Decision Tests” on page 125 and “Content Executed by Published Decisions” on page 137.

1. On the Decision Flow tab, select ▼, and drag the code file or query object onto the diagram where you want to add it. The new file window appears.

   Alternatively, on the Decision tab, complete these steps:

   a. Click Add and select the object type that you want to add to the decision. If an object in the decision is selected, select Edit ➔ Add below, and select the object type. The content selection window appears.


2. Enter a name for the new file if you do not want to use the default name. File names are limited to 255 characters and must be unique within a folder.

3. (Optional) Enter a description for the new file. Descriptions are limited to 1000 characters.

   TIP You can edit the description at any time in the Properties pane.

4. Click ▼, and select the folder where you want to save the file.

5. (Optional) If you are creating a new data query file, select the editor you want to use to create the file.

6. Click Save. SAS Intelligent Decisioning creates a new code file and adds it to the decision.

7. In the Properties pane on the Decision Flow tab, click Open to open the new file in the editor.

   Alternatively, on the Decision tab, click ▽, click the Properties tab, and then click Open to open the new file in the editor.

8. Add your custom code to the file. For information, see Chapter 5, “Using Custom Code Files,” on page 81.

9. Click ▽ and Close to save and close the code file.

10. Click ☐ to refresh the decision and update variable mappings.

---

Adding Record Contacts Nodes

About Recording Contacts

The primary purpose of record contacts nodes is to record the outcome of a decision. You can use record contacts nodes to record the values of specific variables at specific points in a decision. For treatment groups, a record contacts node records metadata for the treatments.
Record contacts nodes record information when the decision they are in is executed in the publishing destination. The behavior of record contacts nodes differs based on whether the decision is published to a SAS Micro Analytic destination or to a destination of another type.

- For decisions that are published to a SAS Micro Analytic Services destination, the record contacts node writes a record to the subject contact history. The record contains the information that you specified that you want to track, such as the data grid of treatments returned by the decision and the values of other variables. In addition, the record contacts node creates an output variable, the record contacts variable, that contains a response tracking code. An application, such as a customer service application, can use this response tracking code to add additional data to the subject contacts history.

- For destinations of other types, the record contacts node does not write a record to the subject contact history. Instead, the information that you specified that you want to track is written to the record contacts variable. This variable is a character string, and it also contains the response tracking code. You can issue a POST request to post the data to the subject contact history. The POST request type is `application/vnd.sas.decision.subject.contact+json`. For more information, see the Subject Contacts API documentation.

A decision can contain multiple record contacts nodes. For example, your decision might have different paths for different channels, and you might want a record contacts node on each path.

## Add a Record Contacts Node

1. On the **Decision Flow** tab, select ![Record Contacts](record-contacts-icon.png), and drag the **Record Contacts** object onto the diagram where you want to add it. SAS Intelligent Decisioning opens the **Properties** pane for the node.

   Alternatively, on the **Decision** tab, select **Add ➤ Record contacts**. If an object in the decision is selected, select **Edit ➤ Add below ➤ Record contacts**.

2. (Optional) Rename the record contacts node. Record contacts node names must be unique within the decision. To rename the node:
   a. On the **Decision Flow** tab, click ![Rename](rename-icon.png) for the node, and select **Rename**. Alternatively, on the **Decision** tab, select the node, and select **Edit ➤ Rename**. The Rename window appears.
   b. Enter a new name and click **Rename**.

3. Click ![Properties](properties-icon.png) to edit the properties of the record contacts node.

4. (Optional) Select the variable that contains the channel information to which you want to attribute the contact data that is recorded by the node.

5. (Optional) Select the variables whose values you want to record.

---

**Note:** You cannot track data grids in record contacts node because of the amount of space required to store the data. For character variables, SAS Intelligent Decisioning records only the first 4000 characters of the value.
6 (Optional) Select **Track treatments** if you want to record which treatments are sent to the calling application. Then, select the data grid variable for the specific set of treatments that you want to track.

**TIP** If you specify a subject ID or subject level in the decision properties, this information is included in the records that are generated for the record contacts node. If you do not specify the subject ID or subject level, the subject ID or subject level is recorded as **NONE**.

7 (Optional) Clear the **Record rule-fired data** check box if you do not want to record the rule-fired information for all rules in the decision up to the point at which the record contacts node is included.

**Note:** Rule-fired data is not recorded for subdecisions.

**Note:** This rule-fired data is recorded in the subject contact history. The rule-fired data that is recorded when you select **Record rule-fired data** when you publish a rule set or decision is recorded in the ruleFiredFlags column in the output table.

8 (Optional) Clear the **Record path tracking** check box if you do not want to record the path-tracking information for all nodes in the decision up to the point at which the record contacts node is included.

**Note:** Path-tracking data is not recorded for decisions that are included in the current decision.

**Note:** This path-tracking data is recorded in the subject contact history. The path-tracking data that is recorded when you select **Record path tracking** when you publish a decision is recorded in the pathID column in the output table.

9 (Optional) Clear the **Include in contact policy** check box if you do not want this contact record included in aggregate reports for the channel.

10 Click **OK** to save your changes.

---

### Adding Branches to a Decision

### Overview of Branches

Branches enable you to add conditional logic to a decision. Depending on the branch type, a branch can have multiple outgoing paths. You can add four types of branches to a decision:
The conditions for the branch paths are evaluated in the order in which you specify them. The first path whose condition evaluates to True is taken by the executing decision. The conditions for the remaining branches are not evaluated.

### About Yes/No Branches

A Yes/No branch compares the value of a numeric or character variable to another variable or to a constant. A Yes/No branch has a single condition and two outgoing paths: Yes and No. For example, the following branch tests whether the value of the DEBTINC variable is less than the constant 35.5.

The following figure shows the property pane for this branch:

- **Variable:** DEBTINC
- **Operator:** <
- **Comparison mode:** Value
- **Variable or value:** 35.5

**TIP** On the Decision Flow tab, this branch type is called a Yes/No branch. On the Decision tab, this branch is called a condition. A condition is the only branch type that you can add to a decision when you are working on the Decision tab.

For more information, see "Add a Yes/No Branch" on page 109 or "Add a Condition (Decision Tab Only)" on page 112.
About Equals Branches

An Equals branch compares the value of a numeric or character variable to other variables or to literal values. By default, this branch has one outgoing path for each comparison variable or literal value, plus a branch labeled Other for any values that are not included in the branches that you create. However, you can combine an outgoing path with the path that immediately follows it by selecting the OR check box.

For example, the following branch compares the character variable MODEL to the string ‘Aventador’, then to ‘Huracan’, and so on, in the order listed in the property pane. The first two comparison strings and the last two strings are combined into one path using the OR operator.

The following figure shows the property pane for this branch:

For more information, see “Add Equals, Range, or LIKE Branches” on page 110.

About Range Branches

A Range branch compares the value of a numeric variable against one or more ranges of values. You can specify range values by using variables or constants.
This branch has an outgoing path for each range, plus a branch labeled **Other** for any values that are not included in the branches that you create.

For example, the following branch compares the value of the BALANCE variable to three ranges. The first range has no minimum value, so it is treated as if the minimum is negative infinity. The third branch has no maximum value, so it is treated as if the maximum is positive infinity.

The following figure shows the property pane for this branch:

![Branch diagram](image)

**Branch variable:**

```
balance
```

<table>
<thead>
<tr>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4999.99</td>
</tr>
<tr>
<td>2</td>
<td>9999.99</td>
</tr>
<tr>
<td>3</td>
<td>No maxmum</td>
</tr>
</tbody>
</table>

For more information, see “Add Equals, Range, or LIKE Branches” on page 110.

### About LIKE Branches

A LIKE branch compares the value of a character variable against one or more strings by using the LIKE operator. In LIKE expressions, you can use the underscore (\_) and percent (%) characters as wildcards. For more information, see “Using the LIKE Operator” on page 24.

By default, this branch has one outgoing path for each comparison string, plus a branch labeled **Other** for any values that are not included in the branches that you create. You can combine paths by selecting the **OR** check box.

For example, the following branch compares the last four characters of the variable EXPIRATIONDATE to determine whether it ends with the characters “2019” or “2020”.

---

108 Chapter 6 / Working with Decisions
The following figure shows the property pane for this branch:

Branch variable:

ExpirationDate

Paths:

<table>
<thead>
<tr>
<th>Value</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>%2019</td>
<td></td>
</tr>
<tr>
<td>%2020</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see “Add Equals, Range, or LIKE Branches” on page 110.

Add a Yes/No Branch

For an example of a Yes/No branch, see “About Yes/No Branches” on page 106.

Note: When you add a Yes/No branch, any objects that follow the currently selected object become part of the No path.

1. Select , and drag the Branch object from the list of objects onto the diagram where you want to add it. The Create New Branch window appears.

2. Select Yes/No, and click OK. SAS Intelligent Decisioning adds the branch and opens the Properties pane.

3. In the Variable field on the Properties pane, select the first variable in the branch condition.

4. Select the expression operator.

5. Select Value or Variable for the comparison mode.

Value

compares the first variable to a literal value. The branch condition becomes first-variable operator value.

Variable

compares the first variable to the value of a second variable. The branch condition becomes first-variable operator second-variable.
6 Select a variable or enter a literal value in the Variable or value field, depending on what you selected in Step 5.

7 (Optional) Add objects to the Yes and No branch paths. Right-click on the branch node, and select one of the following options:

- **Add to yes path**: opens the selection window for the selected object type. After you select the object that you want to add to the decision, SAS Intelligent Decisioning adds it to the Yes path in the decision. This path is executed if the branch condition evaluates to True.

- **Add to no path**: opens the selection window for the selected object type. After you select the object that you want to add to the decision, SAS Intelligent Decisioning adds it to the No path in the decision. This path is executed if the branch condition evaluates to False.

For more information, see the following topics:

- "Add an Existing Object" on page 101
- "Add a New Code File or Data Query File" on page 102
- "Adding Record Contacts Nodes" on page 103

---

## Add Equals, Range, or LIKE Branches

1 Select ☐, and drag the **Branch** object from the list of objects onto the diagram where you want to add it. The Create New Branch window appears.

2 Select **Equals**, **Range**, or **LIKE**, and click **OK**. SAS Intelligent Decisioning adds the branch and opens the **Properties** pane.

    **Note:** When you add an **Equals** branch, a **Range** branch, or a **LIKE** branch to a decision, the **Decision** tab is disabled. For more information, see “The Decision Flow Tab versus the Decision Tab” on page 95.

3 In the **Properties** pane, select the branch variable. The values of this variable are used to control which outgoing path is executed.

    SAS Intelligent Decisioning automatically adds a path labeled **Other**.

4 For each outgoing path that you want to add to the branch node, complete the following steps:

    a Click ➕ to add the path.

    b Specify the values or expressions for the path.

<pre><code>| Branch Type | Values or Expressions |
|-------------|-----------------------|
| Equals      | Specify a comparison variable or constant of the branch variable for the path. See “About Equals Branches” on page 107 for more information. |
</code></pre>
<table>
<thead>
<tr>
<th>Branch Type</th>
<th>Values or Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Specify a minimum value, a maximum value, or both for the branch variable. Minimum and maximum values are optional. If you do not specify a minimum value, the minimum is treated as negative infinity. If you do not specify a maximum value, the maximum is treated as positive infinity. If the minimum and maximum are the same value, the path is followed only if the branch variable is equal to that value. See “About Range Branches” on page 107 for more information.</td>
</tr>
<tr>
<td>LIKE</td>
<td>Specify the LIKE expression for the path in single quotation marks. See “Using the LIKE Operator” on page 24 and “About LIKE Branches” on page 108 for more information.</td>
</tr>
</tbody>
</table>

5 (Optional) Click ↑ or ↓ to reorder the paths. Branch paths are evaluated in order. The decision follows the first path that evaluates to True.

6 (Optional) Combine branch paths. For Equals and LIKE branches, you can select the OR check box to combine a branch path with the path that immediately follows it. The application combines the conditions for the branch paths by using the OR operator.

7 Add objects to the outgoing branch paths. Right-click on the branch node, and select Add to branch path ⇒ Branch label ⇒ Object type. SAS Intelligent Decisioning opens the appropriate window, depending on the object type. For more information, see the following topics:
   - “Add an Existing Object” on page 101
   - “Add a New Code File or Data Query File” on page 102
   - “Adding Record Contacts Nodes” on page 103

Rename Equals, Range, or LIKE Branches

The default name for Equals, Range, and LIKE branches is “Branch (variable)”. Multiple branches in the same decision can have the same name.

1 Right-click on the branch node, and select Rename. The Rename window appears.

2 Enter the new name, and click Rename. The maximum length is 100 characters.
Add a Condition (Decision Tab Only)

Note: When you add a condition, any objects that follow the currently selected object become part of the ELSE clause.

1. Click **Add** and select **Add condition**. If an object in the decision is selected, select **Edit ➔ Add below ➔ Condition**. SAS Intelligent Decisioning adds an IF-THEN-ELSE statement to the decision.

2. In the first field of the IF condition, select the first variable in the condition expression.

3. Select the expression operator.

4. Select **Value** or **Variable** for the comparison mode.
   - **Value** compares the first variable to a literal value. The condition becomes `first-variable operator value`.
   - **Variable** compares the first variable to the value of a second variable. The condition becomes `first-variable operator second-variable`.

5. Select a variable or enter a literal value in the **Variable or value** field, depending on what you selected in Step 5.

   For example, the condition expression `DEBTINC > 35.5` appears as follows:

   ![Condition Expression Example](image)

6. (Optional) Add objects to the THEN and ELSE clauses. “Yes/No Branches and Condition Nodes” on page 95 For more information, see the following topics:
   - “Add an Existing Object” on page 101
   - “Add a New Code File or Data Query File” on page 102
   - “Adding Record Contacts Nodes” on page 103

Mapping Variables within a Decision

About Decision Variables and Mapping

The objects that you add to a decision, such as models and rule sets (including eligibility rule sets in treatments), each define their own variables. When you add a rule set, model, or subdecision to a decision, SAS Intelligent Decisioning
automatically defines decision variables that have the same name and data type as
the object’s variables. SAS Intelligent Decisioning automatically maps the object’s
variables to the decision variables that have the same name and data type.

When you add a treatment group to a decision, SAS Intelligent Decisioning creates
a decision variable of type data grid with the name \texttt{group\_name\_out}. If the treatment
group name is longer than 32 characters, then this name is truncated.

When you add a data query file to a decision, the decision variables that are created
depend on whether the query selects data and on whether the file contains the
comment \texttt{/* include sqlReturnInfo */}. For more information, see “Decision
Variables for Data Query Files” on page 88.

You can create decision variables with different names, and then change the
decision variable mappings on the \textbf{Input Variables} and \textbf{Output Variables} property
panes for objects in the decision.

For example, suppose you have a decision named Credit Approval, and this
decision contains a model named Loan Default and a rule set named Evaluate
Credit. The model has an output variable named \texttt{em\_prob}. The value of this
variable must be passed as input to the Evaluate Credit rule set, but the rule set is
expecting a variable named \texttt{probability}. In order for the value to be passed to the
rule set, you must map the \texttt{em\_prob} output variable of the model to the \texttt{em\_prob}
decision variable, and you must map the \texttt{probability} input variable of the rule to the
\texttt{em\_prob} decision variable.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Variable} & \textbf{Maps To} \\
\hline
\texttt{probability} & \texttt{em\_prob} \\
\texttt{income} & \texttt{income} \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Variable} & \textbf{Maps To} \\
\hline
\texttt{em\_prob} & \texttt{em\_prob} \\
\hline
\end{tabular}
\end{table}

\begin{tikzpicture}
\begin{scope}[local bounding box=decision]
\node [decisionDecision] {Credit Approval Decision Variables};
\node [decisionVariable, below of=decision, xshift=2cm] {
\begin{itemize}
\item approval
\item \texttt{em\_prob}
\item income
\item probability
\item state
\end{itemize}};
\end{scope}
\begin{scope}[local bounding box=model]
\node [modelModel, below of=decisionVariable] {
\begin{itemize}
\item state
\item income
\end{itemize}};
\node [modelVariable, right of=modelModel, xshift=2cm] {
\begin{itemize}
\item probability
\end{itemize}};
\end{scope}
\begin{scope}[local bounding box=ruleSet]
\node [ruleSetRuleSet, below of=modelVariable] {
\begin{itemize}
\item approval
\end{itemize}};
\end{scope}
\begin{scope}[local bounding box=font]
\node [text=12pt, align=left, below of=ruleSet] {
\begin{tabular}{|c|c|}
\hline
\textbf{Input Variables} & \textbf{Maps To} \\
\hline
\texttt{probability} & \texttt{em\_prob} \\
\texttt{income} & \texttt{income} \\
\hline
\end{tabular}};
\node [text=12pt, align=left, below of=modelVariable] {
\begin{tabular}{|c|c|}
\hline
\textbf{Output Variables} & \\
\hline
\texttt{em\_prob} & \texttt{em\_prob} \\
\hline
\end{tabular}};
\end{scope}
\end{tikzpicture}

\textbf{IMPORTANT} \hspace{1em} When the decision is published and run in a production
environment, the decision expects the input data to contain variables that
have the same name and data type as the decision’s input variables.
Scoring Rows in a Data Grid

By default, each node in a decision is executed for the first record in the input data table, then for the second input record, and so on, before execution moves to the next node. However, if the node object uses a data grid variable, you can specify that the object is executed against each row of the data grid instead of against each record in the input data.

To execute an object against each row in a data grid, select the **Score rows in this data grid** option when you are mapping decision variables for the object. Then, select the data grid. This option is available on the **Input Variables** and **Output Variables** property panes for rule sets, models, and subdecisions that contain a data grid. When you select this option, the object processes only the data grid. All other output variables are passed through to the next node in the decision.

**IMPORTANT** When the node object is a filtering rule set, and you select **Score rows in this data grid**, rows that do not meet the criteria defined by the rules are removed from the data grid.

Mapping Data Grid Variables

When you add an object to a decision and the object contains a data grid variable, SAS Intelligent Decisioning creates a decision variable for the data grid in the same way that it creates decision variables for object variables of other data types. When you select **Score rows in this data grid** for an object that uses a data grid, you can choose to map the columns in the object’s data grid variable either to columns in the decision’s data grid variable or to other decision variables. In the lists of variables in the **Input Variables** property pane, the decision’s scalar variables are identified by the icon, and the decision’s data grid columns are identified by the icon.

For more information, see “Map Object Variables to Decision Variables” on page 114.

Map Object Variables to Decision Variables

On the **Decision Flow** tab, complete these steps:

1. Select the object whose variables you want to map.
2. (Optional) Click to open the **Input Variables** property pane.
3. (Optional) Select **Score rows in this data grid**, and select the name of the data grid.
Note: This option appears only for objects that process data grid variables. For more information, see “Scoring Rows in a Data Grid” on page 114.

4 (Optional) For each object variable in the Input Variable column, select the decision variable that you want to map it to in the Maps To column.

5 (Optional) Click $\mathbb{E}$.

6 If the object that you selected is a treatment group, select the data grid that contains the outcome of the treatment group.

7 For each object variable in the Output Variable column, select the decision variable that you want to map it to in the Maps To column. For information about mapping variables for data query files, see “Decision Variables for Data Query Files” on page 88.

On the Decision tab, complete these steps:

1 Select the object for which you want to map variables.

2 Click $\mathbb{E}$ to open the Properties window.

3 (Optional) Select Score rows in this data grid, and select the name of the data grid.

Note: This option appears only for rule sets, models, and decisions that process data grid variables. For more information, see “Scoring Rows in a Data Grid” on page 114.

4 For each object variable in the Input Variable column, select the decision variable that you want to map it to in the Maps To column.

5 If the selected node is a treatment group, select the data grid that contains the outcome of the treatment group.

6 For each object variable in the Output Variable column, select the decision variable that you want to map it to in the Maps To column. For information about mapping variables for data query files, see “Decision Variables for Data Query Files” on page 88.

7 Click OK to close the Properties window.

---

**Editing the Objects in a Decision**

**Open an Object from within a Decision**

On a node in the Decision Flow diagram, double-click on the object name, or click $\mathbb{E}$ for the object that you want to open, and select Open.
The object opens in the appropriate editor. Models open in SAS Model Manager if you have access to that application.

## Edit the Properties of a Decision Node

On the **Decision Flow** tab:

1. Select the decision node.
2. (Optional) Click to display the **Input Variables** pane, and modify the input variable mappings for the node.
3. (Optional) Click to display the **Output Variables** pane, and modify the output variable mappings for the node.
4. (Optional) Click to display the **Properties** pane.
   a. (Optional) Select a different object or object version for the selected node.
   b. (Optional) For code file nodes, modify the node description, or click **Open** to edit the code file in the appropriate editor.

On the **Decision** tab:

1. Click for the node. The properties window for the node appears.
2. (Optional) On the **Variables** tab, modify the variable mappings.
3. (Optional) On the **Properties** tab, select a different object or object version. For code files, modify the description, or click **Open** to edit the file in the appropriate editor.
4. Click **Close** to save your changes and close the properties window.

## Reorder Objects

On the **Decision Flow** tab, you can drag rule sets, branches, models, and code files from one position to another.

On the **Decision** tab, to move an object up or down, including into and out of conditions, select the object and click ↑ or ↓. You can also move the selected object by using Shift + ↑ and Shift + ↓.

## Delete an Object

On the **Decision Flow** tab, click ⏺️ on the object that you want to delete, and select **Delete**.

On the **Decision** tab, click ⏺️ on the object that you want to delete.
Copy a Decision URL

To create a link for external documentation that automatically opens a decision in SAS Intelligent Decisioning, complete these steps:

1. Open the decision.
2. Click ..., and select **Copy object URL**. The Copy Decision URL window appears.
3. Click **Copy**, and then click **Close**.
   Paste the link into your documentation.

Generate PDF Documentation for a Decision

You can generate detailed documentation for a decision as a PDF document. The PDF includes the decision properties, the properties and variable mappings for each node, and a table of the decision variables. You can download documents with additional details about the rule sets, treatment groups, and subdecisions that are used in the decision.

1. Open the decision, and click the **Decision Flow** tab.
2. Click ..., and select **Create document**. The Create Document window appears.
   
   **TIP** If this option is disabled, there might be unsaved changes. Click .

3. (Optional) Enter a name for the document if you do not want to use the default name.
4. (Optional) Select **Show a diagram of the decision** to include the decision diagram in the PDF.
5. (Optional) Select **Include code from code nodes** to include the code from custom code files that are used in the decision.
6. (Optional) Select **Choose additional documents to download** to display a window from which you can download additional documents. You can download documents for any rule sets, treatment groups, and subdecisions that are used in the decision.
7 Click **Create**. SAS Intelligent Decisioning creates the PDF. The result depends on your browser’s default download actions. The browser might automatically download the file and display a notification in the bar at the bottom of the browser window. Alternatively, the browser might ask you to choose a download location.

If you download additional documents in **Step 4**, the Download Additional Documents window appears.

8 (Optional) Click **Download** for each additional document that you want to download.

9 (Optional) Click **Close** to close the Download Additional Documents window.

---

**Managing Decisions**

**Duplicate Decisions**

---

**Note:** You cannot duplicate a decision if it is open.

---

To duplicate a single decision:

1 In the Decisions view, select the decision that you want to duplicate.

2 Click ↗ and select **Duplicate**. The Duplicate Decision window appears.

3 Enter a new name for the duplicate decision

4 (Optional) Enter a description for the decision.

5 Select the version of the decision that you want to duplicate.

6 Click ↘ and select the location where you want to save the duplicate decision.

7 Click **Duplicate**.

To duplicate multiple decisions:

1 In the Decisions view, select the decisions that you want to duplicate.

2 Click ↗ and select **Duplicate**. SAS Intelligent Decisioning duplicates the latest version of the decisions and appends _copy to the names of the duplicate copies. If needed, a number is also appended to the names of the duplicate copies.
Delete Decisions

**Note:** You cannot delete a decision if it is open.

In the Decisions category view, select the decisions that you want to delete, click \( 	ext{ } \), and select **Delete**.

Rename a Decision

**Note:** You cannot rename a decision if it is open.

1. In the Decisions category view, select the decision that you want to rename.
2. Click \( 	ext{ } \) and select **Rename**. The Rename window appears.
3. Enter a new name for the decision, and click **Rename**.

Move Decisions to a Different Folder

1. In the Decisions category view, select the decisions that you want to move.
2. Click \( 	ext{ } \) and select **Move**. The Choose a Location window appears.
3. Select the location to which you want to move the decisions, and click **OK**.

Managing Versions of Decisions

Set the Displayed Version

The displayed version is the version whose information is displayed on the other tabs, such as the **Properties** and **Decision** tabs. On the **Versions** tab, a \( \checkmark \) indicates the displayed version. To change the displayed version, select the version that you want to view, and click **Set Version**. The displayed version is shown in the title bar.
Create a New Version

Note: The current version of an object is the version with the highest version number. When you create a new version, SAS Intelligent Decisioning locks the current version before it creates the new version.

Note: You cannot save changes to a version that is locked. If you modify a version that is locked and click , SAS Intelligent Decisioning asks you if you want to replace the current unlocked version with your edited version.

IMPORTANT You cannot unlock a locked version.

To create a new version:


2 Select the version type: Minor or Major. Version numbers follow the format Major.Minor. If you select Major, the number to the left of the period is incremented. If you select Minor, the number to the right of the period is incremented.

3 (Optional) Enter information about the new version in the Notes field.

TIP You can edit these notes at any time on the Versions tab.

4 Click Save.

Copy the Content of a Version

You can copy the content of an object’s version in the category view or on the Version tab for the object.

1 In the category view, complete these steps:
   a Select the decision whose contents you want to copy.
   b Click , and select Copy version. The Copy Version window appears.
   c Select the version whose contents you want to copy.

Alternatively, on the Versions tab of an object:
   a Select the version whose contents you want to copy.
   b Click , and select Copy version. The Copy Version window appears.
Managing Versions of Decisions

Delete a Version

Note: In order to be able to delete a specific version of a decision, you must have permission to delete the decision itself.

On the Versions tab, select the version that you want to delete, click , and select Delete.
You cannot delete the current version.

Upgrade an Object Version That Is Used in a Decision

If you create a new version of an object that is already used in a decision, you can upgrade the decision to use the new version.

1. On the Versions tab for the object, click , and select Upgrade decisions. The Upgrade Decisions window appears. This window lists all of the decisions that include the object.

2. In the Version to upgrade to field, select the version of the object to which you want to upgrade the decisions.

3. Select the check boxes for the decisions that you want to upgrade, and click Upgrade Decisions.
Testing a Decision

Create and Run a New Test

Testing a decision is optional, but doing so is a best practice. Testing enables you to discover any problems before the decision is published and incorporated into a production system.

1. On the Scoring tab, click the Tests tab.


3. Enter a name for the test if you do not want to use the default name.

4. (Optional) Enter a description for the test. Descriptions are limited to 1000 characters.

5. (Optional) Click for the Location field, and select the folder where you want to save the test definition and results.

TIP Selecting a location is optional, but it is highly recommended. Storing test definitions and test results in a folder simplifies the tasks of setting permissions and transferring the test files.

6. Click for the Input table field, select the input table for the test, and click OK.

7. Verify or change the variable mappings. The input variables in the decision must be mapped to columns in the input table that you selected for the test.

SAS Intelligent Decisioning automatically maps the input variables in the decision to columns in the input table when the names and data types of the variables match those of the table columns. If any input variables cannot be mapped automatically, an error message is displayed.

You can change the automatic variable mappings in the Variable Mappings window.

To change variable mappings:

a. Click Variables. The Variable Mappings window appears.

b. For each input variable, select the table column to which the variable should be mapped. Alternatively, for Decimal, Integer, and Character variables, you
can select **Use value** for the table column, and specify a literal value in the **Value** column. When you are entering literal values, remember these rules:

- Do not enclose character strings in quotation marks.
- To specify a missing value for character variables, select **Use value** and leave the **Value** column empty. When SAS Intelligent Decisioning generates code, it generates an empty string (""). For numeric values, enter a period (.).

---

**Note:** For content that will be published to a SAS Micro Analytic Service destination, the values of date and datetime input variables must be numeric. They cannot be formatted values such as 18jul2019.

---

c Click **OK** to close the Variable Mappings window.

8 (Optional) Click **Advanced** to display the advanced options.

9 (Optional) Click ☰ and select the library where you want to write the output of the test.

10 (Optional) Select the version of the decision that you want test.

11 (Optional) Select the variable that you want to serve as an input debug variable. You can specify an input-only variable or an input-output variable. The decision writes the name and value of this variable to the log for each input record that is processed. It writes the value just before the logic of the decision is executed for the input record.  

For more information, see "Debugging Rule Set Tests" on page 38.

12 (Optional) Select **Preserve unmapped columns in the output table** if you want columns that are not mapped to an output variable to be written to the output table.

13 (Optional) To generate a data grid variable of all decision variables and their values, select **Enable value tracing**. For more information, see "Enabling Value Tracing" on page 124.

14 Click **Run** to run the test. Alternatively, click **Save** to save the test definition without running it.

The status of the test is indicated by the icon in the **Status** column.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon1.png" alt="Icon 1" /></td>
<td>The test is not ready to run. The test definition is not complete, or it might contain errors.</td>
</tr>
<tr>
<td><img src="icon2.png" alt="Icon 2" /></td>
<td>The test is defined correctly and is ready to run.</td>
</tr>
<tr>
<td><img src="icon3.png" alt="Icon 3" /></td>
<td>The test is running.</td>
</tr>
<tr>
<td><img src="icon4.png" alt="Icon 4" /></td>
<td>The test completed successfully.</td>
</tr>
<tr>
<td><img src="icon5.png" alt="Icon 5" /></td>
<td>The test completed, but warnings were issued in the SAS log. The URI to the log file is shown on the Test Results page. See Step 15.</td>
</tr>
</tbody>
</table>
Icon   Status

- The test did not run successfully. Check the SAS log for information. The URI to the log file is shown on the Test Results page. See Step 15.

15 Click in the Results column to view the results of the test.

16 On the Test Results page, click Test Results in the navigation pane to display the URIs and other information for the test. Click Output, Code, or Log to display the output data set, the code that was generated by SAS Intelligent Decisioning, or the SAS log that was generated when the code was run.

**TIP** On the Output page, you can click on the values of character variables to display the entire value in a separate window. For data grid variables, you can choose to view the variable value in three different formats:
- Click the Data Grid tab to view the data grid value as a table.
- Click the Formatted tab to view the data grid as a formatted JSON character string.
- Click the Plain tab to view the data grid as an unformatted character string.

**TIP** On the Log page, you can click to download the log file.

---

**Enabling Value Tracing**

When you test a decision, you can select the **Enable value tracing** check box to trace how variable values change for each node in the decision flow. When you select this option, SAS Intelligent Decisioning creates an output data grid variable named nodeTraceDataGrid. This data grid contains a row for each rule set, model, subdecision, and code file node in the decision. It contains one column for every variable in the decision. Condition nodes, branch nodes, and record contact nodes are not included.

The first column in nodeTraceDataGrid contains the node name, and the remaining columns contain the values of the input and output variables, including temporary variables.

Within the nodeTraceDataGrid variable, data grids are written as string variables, and their length is limited to 32767 characters.

When a node sets a variable’s value, that value appears unchanged in all subsequent rows until another node changes the variable to a different value. In the following example, the Code_1 node sets score variable to a value of 1. The Code_2 and Code_3 nodes do not change that value. The Rule_set_1 node changes the value to 1.25. The Code_1 node sets the value of the
scoreProbDataGrid variable to null. The Code_3 node sets the value of scoreProbDataGrid to a data grid with two columns: score and probability.

```javascript
{
  "Data Grid": [{
    "metadata": [
      {"Node Name": "string"},
      {"additionalDiscount": "double"},
      {"age": "double"},
      {"cost": "double"},
      {"currentPhone": "double"},
      {"score": "double"},
      {"treatmentName": "double"},
      {"ServicePlans_out": "string"},
      {"newScore": "double"},
      {"probability": "double"},
      {"scoreProbDataGrid": "string"}
    ],
    "data": [
      ["Code_1", null, 42, 699.99, null, 1, null, null, 1.1, 0.625547, null],
      ["Code_2", null, 42, 699.99, null, 1, null, null, 1.1, 0.625547, null],
      ["Code_3", null, 42, 699.99, null, 1, null, null, 1.1, 0.625547,
        {"Data Grid": [{"metadata": [{"score": "double"},
        {"probability": "double"}], "data": [[1.23, 4.56]]}]},
      ["Rule_set_1", 99.99, 42, 699.99, null, 1.25, null, null, 1.1, 0.625547,
        {"data": [{"metadata": [{"score": "double"},
        {"probability": "double"}], "data": [[1.23, 4.56]]}]},
      ["Combined_Treatments", 99.99, 42, 699.99, null, 1.25, null, null, 1.1, 0.625547,
        {"data": [{"metadata": [{"score": "double"},
        {"probability": "double"}], "data": [[1.23, 4.56]]}]}
    ]
  }, {
    "data": [
      ["Code_1", null, 42, 699.99, null, 1, null, null, 1.1, 0.625547, null],
      ["Code_2", null, 42, 699.99, null, 1, null, null, 1.1, 0.625547, null],
      ["Code_3", null, 42, 699.99, null, 1, null, null, 1.1, 0.625547,
        {"Data Grid": [{"metadata": [{"score": "double"},
        {"probability": "double"}], "data": [[1.23, 4.56]]}]},
      ["Rule_set_1", 99.99, 42, 699.99, null, 1.25, null, null, 1.1, 0.625547,
        {"data": [{"metadata": [{"score": "double"},
        {"probability": "double"}], "data": [[1.23, 4.56]]}]},
      ["Combined_Treatments", 99.99, 42, 699.99, null, 1.25, null, null, 1.1, 0.625547,
        {"data": [{"metadata": [{"score": "double"},
        {"probability": "double"}], "data": [[1.23, 4.56]]}]}
    ]
  ]
}
```

## Content That Is Used by Decision Tests

For each object in a decision, the content that is used when the decision is tested depends on the object type.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Object Content that Is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subdecision</td>
<td>The version that you select when you add the subdecision to the decision</td>
</tr>
<tr>
<td>Rule set</td>
<td>The version that you select when you add the rule set to the decision</td>
</tr>
<tr>
<td>Model</td>
<td>The latest version</td>
</tr>
<tr>
<td>Treatment group</td>
<td>The active version</td>
</tr>
</tbody>
</table>
Run a Rule-Fired Analysis

If a rule’s conditions evaluate to True, then the rule is said to have fired. Rule-fired data includes summary information about how many times each rule fired and detailed information for each time that a rule evaluates to True. See "How Rules Are Evaluated and When Rule-Fired Records Are Generated" on page 18 for more information.

Note: Rule-fired data is not recorded for decisions that do not include a rule set, subdecisions, input records that are filtered out with a filtering rule set, or eligibility rules in a treatment. For rule sets that iterate over a data grid (in other words, the rule sets score the rows in the data grid), the rule-fired data indicates that the rules in the rule set fire once for the entire data grid instead of firing once for each row in the data grid.

TIP  This rule-fired analysis uses the data that is in the ruleFiredFlags column in the test results output table. To analyze rule-fired data that is in the subject contact history, use the %DCM_GET_SUBJECTCONTACT_HISTORY and %DCM_RULEFIRE_DETAIL macros. For more information, see SAS Intelligent Decisioning: Macro Guide.

1 On the Test Results page, click Rule-Fired Analysis in the navigation pane.

2 Click Run Rule-Fired Analysis. SAS Intelligent Decisioning analyzes the test results to determine which rules fired for each row in the input table, and displays the Analysis page.

   The Analysis page displays the number of rules that fired for each output record that was generated by the decision. The number in the Rules Fired Count column is a link to more information. You can click on this link to display the rules that fired for that output row.

   For example, in the following display there is one output record for which two rules fired.
3 Click on a number in the Rule Fired Count column. SAS Intelligent Decisioning displays the Rule Fired Count window. This window shows which rules produced the selected output record.

4 Click Close to close the Rule Fired Count window.

5 Click Plot in the navigation pane. SAS Intelligent Decisioning displays a bar chart that shows how many times each rule fired. Position your cursor over a bar to display the name of the rule and the number of times that the rule fired.
6 Click **Rule-Fired Analysis** in the navigation pane to display the URIs and other information for the rule-fired test.

7 Click **Close** to close the decision.

---

### Run a Path Tracking Analysis

Decision path tracking shows you the route that input records take through the nodes in your decision.

**Note:** The path tracking results do not include data for input records that are filtered out with a filtering rule set.

**Note:** Path-tracking data is not recorded for decisions that are included in the current decision.

---

**TIP** This path-tracking analysis uses the data that is in the pathID column in the test results output table. To analyze path-tracking data that is in the subject contact history, use the `%DCM_DECISION_PATH_FREQUENCY`, `%DCM_DECISION_PATH_NODES`, and `%DCM_DECISION_NODES_COUNTS` macros. For more information, see *SAS Intelligent Decisioning: Macro Guide*.

---

1 Click **Decision Path Tracking** in the navigation pane.

2 Click **Run Path Tracking** to run a decision path analysis.

3 Click **Analysis and Plot** to display a Sankey diagram that shows the flow of the input records through the nodes in the decision. The numbers in the diagram are the number of rows in the input table that followed each path.
Note: Nodes that are not executed are shown to the right of the Sankey diagram.

4 Click **Node Count** in the navigation pane to display a table showing the number of input records evaluated at each rule set, code file, and model node in the decision.

5 Click **Close** to close the Test Results window.

---

**Working with Test Output Data**

After you run a test, you can work with the output table in other SAS applications to analyze the data, create and compare models, discover relationships hidden in the data, and generate reports based on the data.
The actions available to you depend on the applications that are available at your site.

On the Test Results page, select the Output table in the navigation pane, click Actions, and select one of the following options:

### Explore Lineage
opens SAS Lineage Viewer. SAS Lineage Viewer enables you to better understand the relationships between objects in your SAS Viya applications. These objects include data, transformation processes, reports, and visualizations. For more information, see [SAS Lineage Viewer: User’s Guide](#).

### Explore and Visualize Data
opens the output table in SAS Visual Analytics. SAS Visual Analytics enables you to create, test, and compare models based on the patterns discovered during exploration of the data. You can export the model before or after performing model comparison for use with other SAS products or to put the model into production. SAS Visual Analytics supports a range of visualization, discovery, and reporting features. For more information, see [SAS Visual Analytics: Overview](#).

### Prepare Data
opens the output table in SAS Data Studio. SAS Data Studio enables you to perform data transforms such as joining tables, appending data to a table, transposing columns, creating calculated columns, and so on. For more information, see [SAS Data Studio: User’s Guide](#).

### Manage Data
opens SAS Data Explorer. SAS Data Explorer enables you to import data, connect to databases, and load tables into memory. For more information, see [SAS Data Explorer: User’s Guide](#).

---

### Publishing and Validating a Decision

#### Publishing a Decision

### Introduction to Publishing

Publishing content makes it available to other applications. Publishing a decision creates an entity that can be managed and run in another environment. For example, if you publish content to a SAS Micro Analytic Service destination, SAS Intelligent Decisioning generates code for the content and writes that code as a module in SAS Micro Analytic Service. The module becomes a callable REST API endpoint, independent of SAS Intelligent Decisioning.
Note: The publishing destinations that are available are determined by your system administrator. See SAS Viya Administration: Publishing Destinations for more information.

Publishing Decisions That Include Analytic Store Models

If you are publishing a decision that includes an analytic store model, the model’s analytic store (ASTORE) file must be in the /opt/sas/viya/config/data/modesvr/astore directory on the CAS server for the destination to which you are publishing the decision. The ASTORE file is copied to that location when you do any of the following:

- run a decision test for the decision that uses the analytic store model
- set the model as a project champion in SAS Model Manager
- publish the model to SAS Micro Analytic Service from SAS Model Manager

If you are publishing a decision that includes an analytic store model and the model has not been set as a project champion or published from SAS Model Manager, you must test the decision before you publish it. For more information, see “Configuring Access to Analytic Store Models” in SAS Intelligent Decisioning: Administrator’s Guide and “Testing a Decision” on page 122.

Publish Decisions

TIP To view the publishing history for a decision, open the decision, and click the History tab.

1 Select the decisions that you want to publish.
   - To publish the latest version of one or more decisions, select the decisions on the Decisions page, click \( \text{ } \), and select Publish.
   - To publish a selected version of one decision:
     1 Open the decision, and click the Versions tab.
     2 Select the version that you want to publish, click Set Version, and then click Publish.

   The Publish Decisions window appears.

2 Select the destination to which you want to publish. The publishing destinations that are available to you depend on what is configured at your site. See SAS Viya Administration: Publishing Destinations for more information.

3 (Optional) In the Items to Publish section, edit the Published Name if you do not want to use the default name for the published module. The maximum length and character restrictions differ depending on your destination. See Table 6.1.
Table 6.1  Requirements and Restrictions for Published Names

<table>
<thead>
<tr>
<th>Destination</th>
<th>Maximum Length</th>
<th>Requirements And Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Micro Analytic Service</td>
<td>100</td>
<td>The published name cannot contain the following characters: ! @ $ % ^ &amp; ( )</td>
</tr>
<tr>
<td>Teradata</td>
<td>128</td>
<td>The published name must start with a letter or an underscore. It cannot contain spaces, multi-byte characters, or special characters other than the underscore.</td>
</tr>
<tr>
<td>SAS Cloud Analytic Services (CAS)</td>
<td>128</td>
<td>The published name cannot contain single or double quotation marks.</td>
</tr>
<tr>
<td>Apache Hadoop</td>
<td>128</td>
<td>The published name cannot contain colons (:) or double quotation marks.</td>
</tr>
</tbody>
</table>

4  (Optional) If you have previously published the decision, select the check box in the Replace column in order to replace the previously published decision of the same name in the same destination.

5  (Optional) Select the Rule Fired Tracking check box if you want the published decision to generate rule-fired data.

Note: This option is disabled if you are publishing content to SAS Micro Analytic Service. Rule-fired data is not recorded for subdecisions. This rule-fired data is recorded in the ruleFiredFlags column in the output table. The rule-fired data that is recorded when you select Record rule-fired data for a record contacts node is recorded in the subject contact history.

6  (Optional) Select the Path Tracking check box if you want the published decision to generate path tracking data.

Note: This option is disabled if you are publishing content to SAS Micro Analytic Service. Path-tracking data is not recorded for decisions that are included in the current decision. This path-tracking data is recorded in the pathID column in the output table. The path-tracking data that is recorded when you select Record path tracking for a record contacts node is recorded in the subject contact history.

7  Publish the decisions.

- To publish content to a SAS Cloud Analytic Services (CAS) destination, you must reload the CAS destination table in order to make the newly published
item available to other applications. Select one of the following options to publish the decision:

**Publish**
publishes the decisions and automatically reloads the CAS destination table. If another user is executing the code for an item that was previously published to CAS while the destination table is being reloaded, reloading the table might cause temporary problems with accessing the table content. After the table is reloaded, all authorized users can access all items in the table.

**Publish without reloading**
publishes the decisions but does not reload the CAS destination table. You must manually reload the table in order for the newly published items to be accessible.

To publish the decisions to Teradata, Apache Hadoop, or SAS Micro Analytic Service destinations, click **Publish**. You do not need to reload the destination table when you publish to these destinations.

The Publishing Results window appears. It displays the names of the published items, their status, and information about any issues that were encountered during the publishing process.

8 After the status changes to **Published successfully**, click **Close** to close the Publishing Results window.

9 (Optional) Click **Close** to close the decision.

---

**Validate a Published Decision**

You can test the published decision in the publishing destination. When you publish the decision, a validation test is automatically defined for that decision in that destination. To run the publishing validation test:

1 On the **Scoring** tab, click the **Publishing Validation** tab. The icon in the **Status** column indicates that the test is not ready to run. The icon indicates that the test is ready to run.

2 Click on the test name. The **Edit Publishing Validation Test** window appears.

   **Note:** To generate the name of the publishing validation test, SAS Intelligent Decisioning appends a timestamp to the decision name. The timestamp indicates when the decision was published.

3 (Optional) Click **folder** in the **Location** field, and select the folder where you want to save the test definition and results.

   **TIP** Selecting a location is optional, but it is highly recommended. Storing test definitions and test results in a folder simplifies the tasks of setting permissions and transferring the test files.

4 Click **folder** in the **Input table** field, and select the input table for the test.
Note: For content that will be published to a SAS Micro Analytic Service destination, the values of date and datetime input variables must be numeric. They cannot be formatted values such as 18jul2019.

Note: If the input table contains a character column, and that column contains control characters in any row, do not use the table as input for publishing validation tests.

If you are validating content that was published to SAS Micro Analytic Service, the time required to run the test depends on the number of worker threads on your system, the number of threads in the middle tier, and the network latency between CAS and the middle tier server. It is recommended that you select an input table with as few input records as needed to accurately test the published content. See *SAS Micro Analytic Service: Programming and Administration Guide* for more information.

5 (Optional) Expand the Advanced section, click in the Output data library field, and select a library to store the validation test output data.

6 Click Run to run the test. Alternatively, click Save to save the test definition without running it.

The status of the test is indicated by the icon in the Status column.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Circle" /></td>
<td>The test is not ready to run. You might need to edit the test and select an input data table or map variables.</td>
</tr>
<tr>
<td><img src="image" alt="Check" /></td>
<td>The test is defined correctly and is ready to run.</td>
</tr>
<tr>
<td><img src="image" alt="Paw" /></td>
<td>The test is running.</td>
</tr>
<tr>
<td><img src="image" alt="Check" /></td>
<td>The test completed successfully.</td>
</tr>
<tr>
<td><img src="image" alt="Check with exclamation" /></td>
<td>The test completed, but warnings were issued in the SAS log. The URI to the log file is shown on the Test Results page. See Step 15.</td>
</tr>
<tr>
<td><img src="image" alt="Circle with x" /></td>
<td>The test did not run successfully. Check the SAS log for information. The URI to the log file is shown on the Test Results page. See Step 15.</td>
</tr>
</tbody>
</table>

7 Click in the Results column to view the test results.

8 On the Test Results page, click Test Results in the navigation pane to display the URIs and other information for the test. Click Output, Code, or Log to display the output data set, the code that was generated by SAS Intelligent Decisioning, or the SAS log that was generated when the code was run.

**TIP** On the Output page, you can click on the values of character variables to display the entire value in a separate window. For data grid variables, you can choose to view the variable value in three different formats:
Click the **Data Grid** tab to view the data grid value as a table.

Click the **Formatted** tab to view the data grid as a formatted JSON character string.

Click the **Plain** tab to view the data grid as an unformatted character string.

**TIP** On the **Log** page, you can click ⌂ to download the log file.

9 Click **Close** to close the decision.

---

**Executing Published Content**

How you execute published content depends on the destination to which the content is published.

---

**Executing Content Published to SAS Micro Analytic Service Destinations**

The user who is executing the published content must be authenticated. In SAS Viya, authentication options vary, based on which interface and operating system are used in your environment. External mechanisms include direct LDAP authentication, host authentication, Kerberos, Security Assertion Markup Language (SAML), and OAuth 2.0 with OpenID Connect. For additional information, see [*SAS Viya Administration: Authentication*](#).

When a rule set or decision is published from SAS Intelligent Decisioning to a SAS Micro Analytic Service destination, an EXECUTE step is created in the published module. For information about the request and response data formats used in this step, see [Execute a step](#) in the REST API documentation for the Micro Analytic Score API. For an example that uses Python to execute a published decision in the maslocal destination, see “Execute a Published Decision” in [*SAS Intelligent Decisioning: Decision Management REST API Examples*](#).

---

**Executing Content That Has Been Published to SAS Cloud Analytic Services Destinations**

To execute content that has been published to SAS Cloud Analytic Services (CAS), use the CAS Model Publishing and Scoring action set. For example, the following code runs a model named Evaluate_Loans on the local CAS server.
/* Start a CAS session named _mmcas_. */
cas _mmcas_

/* Create librefs for all existing caslibs so that they */
/* are visible in the SAS Studio Libraries tree. */
caslib _all_ assign;

proc cas;
    /* Specify the session to use for the runModelLocal action. */
    session _mmcas_

    /* Define the parameter list for the runModelLocal action. */
    destination_model_table = "targetCode";
    destination_model_lib = "public";
    destination_model = "Evaluate_Loans";
    dp_inputTable="hmeq_test";
    dp_inputCASLib="public";
    dp_outputTable="hmeq_test_dm";
    dp_outputCASLib="public";

    parmlist = {
        modelTable={
            name=destination_model_table,
            caslib=destination_model_lib
        },
        modelName=destination_model,
        inTable={
            name=dp_inputTable,
            caslib=dp_inputCASLib
        },
        outTable={
            name=dp_outputTable,
            caslib=dp_outputCASLib
        }
    });

    /* Load the modelPublishing action set. */
    loadactionset "modelPublishing";

    /* Run the model locally on the CAS server. */
    action runModelLocal submit result=r status=rc / parmlist;
    run;
    quit;

You can submit this code in SAS Data Studio. To open SAS Data Studio, click ☐ and select Prepare Data. For more information, see SAS Data Studio: User’s Guide.

You can view additional examples of using this CAS action set to execute published content by viewing the test results that are generated by publishing validation tests. On the Test Results page for a decision or rule set, click Code to display the code that was generated by SAS Intelligent Decisioning. For information about running
publishing validation tests and viewing the results, see “Validate a Published Decision” on page 133 and “Validate a Published Decision” on page 133.

For more information about CAS and the Model Publishing and Scoring action set, see the following documentation:

- **Getting Started with CASL Programming**
- **SAS Cloud Analytic Services: CASL Reference**
- **SAS Cloud Analytic Services: User’s Guide**

## Content Executed by Published Decisions

When you execute a published decision, the version of the content that is executed depends on the publishing destination.

<table>
<thead>
<tr>
<th>Destination Type</th>
<th>Content That Is Executed by Published Decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Micro Analytic Service</td>
<td>Most of the published decision’s content (the SAS Micro Analytic Service module) is locked. Updates to rule sets, models, or custom code files that are used in the decision do not affect the published module. Treatment groups are not locked. If a new version of a treatment group is activated, the newly activated version is used by the published module. Lookup tables are locked if the lookupStaticBinding configuration option is turned on. If this option is off, lookup tables are not locked, and published modules use newly activated versions of lookup tables. For more information, see “Properties for All Environments” in <strong>SAS Intelligent Decisioning: Administrator’s Guide</strong>.</td>
</tr>
<tr>
<td>SAS Cloud Analytic Services</td>
<td>Most of the published decision’s content is locked. Updates to rule sets, models, treatment groups, or custom code files that are used in the decision do not affect the published decision. Lookup tables are locked if the lookupStaticBinding configuration option is turned on. If this option is off, lookup tables are not locked, and published modules use newly activated versions of lookup tables. For more information, see “Properties for All Environments” in <strong>SAS Intelligent Decisioning: Administrator’s Guide</strong>.</td>
</tr>
<tr>
<td>Teradata or Hadoop</td>
<td>All of the published decision’s content, including treatment groups and lookup tables, is locked. Updates to the objects used in the decision are not used by the published module.</td>
</tr>
</tbody>
</table>