Essential Functions of SAS® Intelligent Decisioning

Course Notes

Essential Functions of SAS® Intelligent Decisioning Course Notes was developed by Lise Cragen. Additional contributions were made by Christopher Meade. Instructional design, editing, and production support was provided by the Learning Design and Development team.

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

Essential Functions of SAS® Intelligent Decisioning Course Notes

Copyright © 2021 SAS Institute Inc. Cary, NC, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

Book code E78789, course code EBDID0D, prepared date 07Sep2021. EBDID0D_001

ISBN 978-1-955977-33-3

Table of Contents

To learn more…

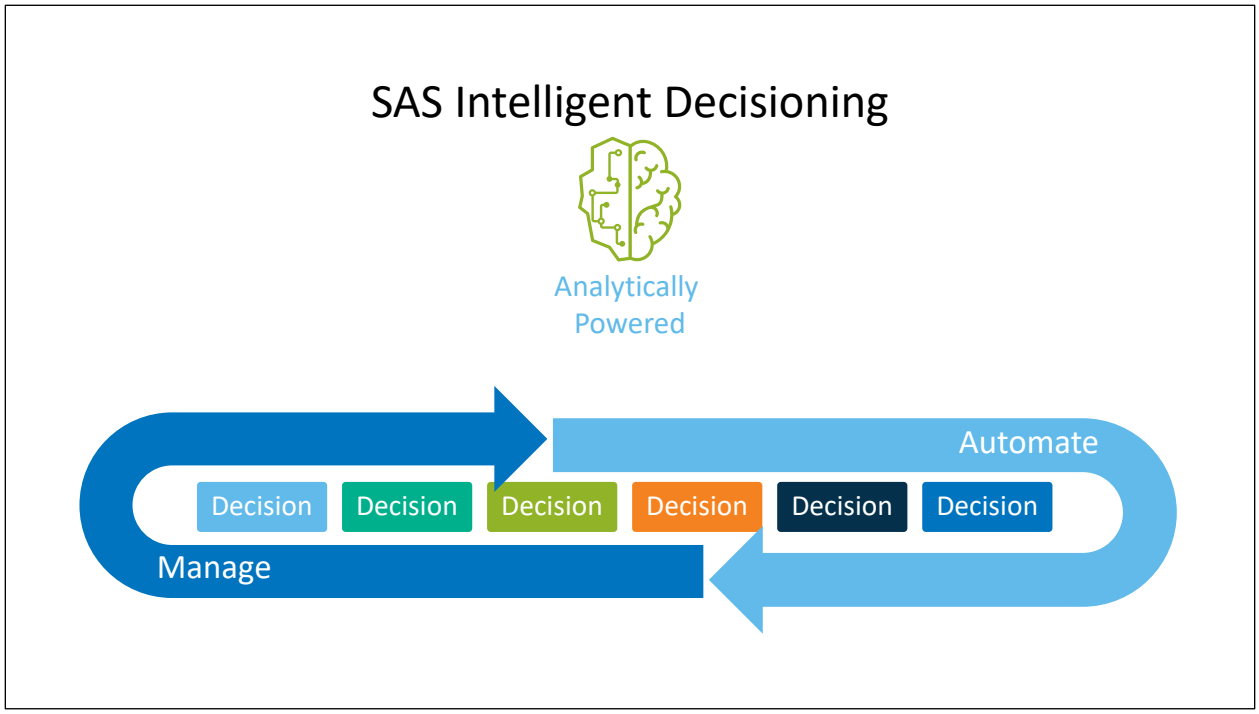
For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the web at http://support.sas.com/training/ as well as in the Training Course Catalog.

For a list of SAS books (including e-books) that relate to the topics covered in this course notes, visit https://www.sas.com/sas/books.html or call 1-800-727-0025. US customers receive free shipping to US addresses.

Lesson 1 SAS Intelligent Decisioning: The Big Picture

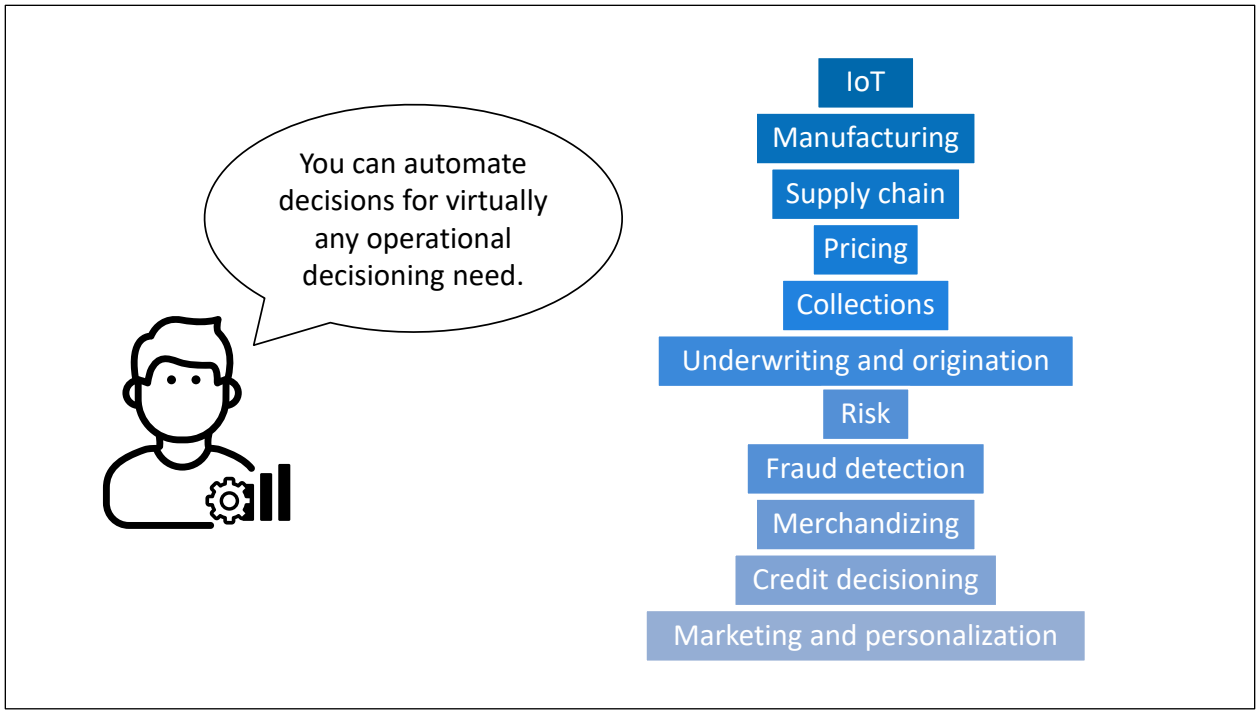
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

1.1 Introduction to SAS Intelligent Decisioning

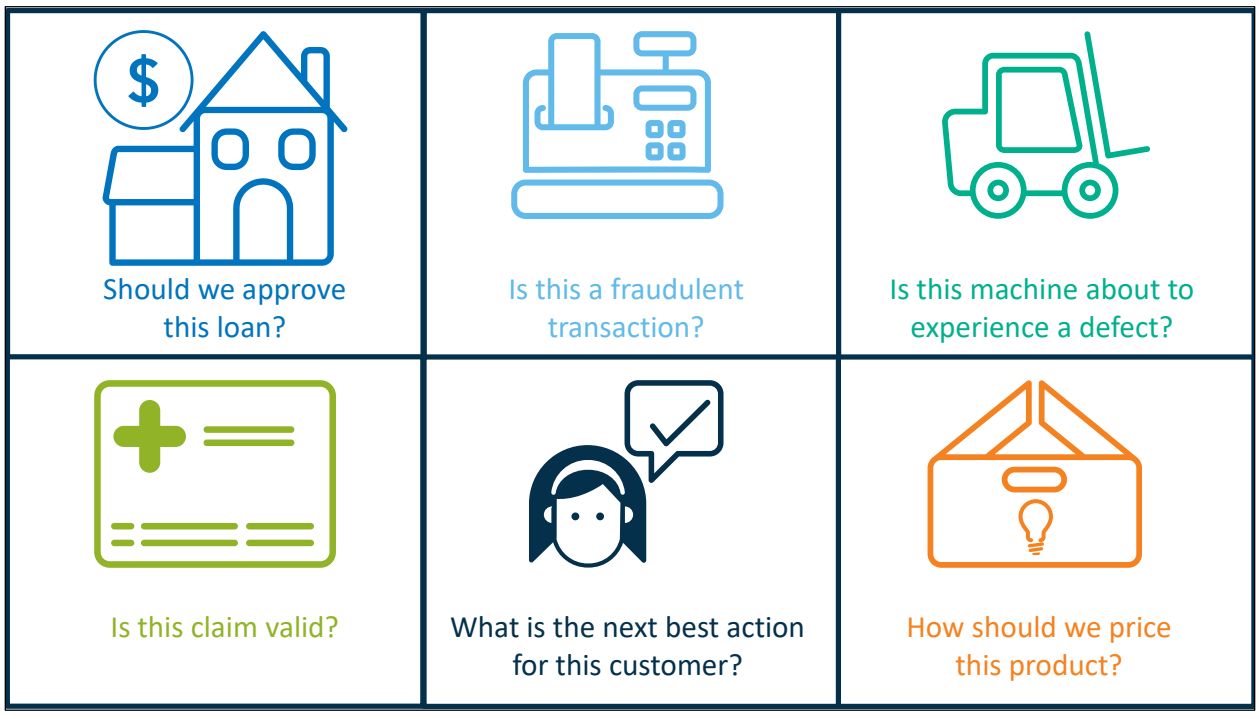


SAS Intelligent Decisioning is an analytically powered decisioning solution that enables you to automate and manage a range of operational decisions.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

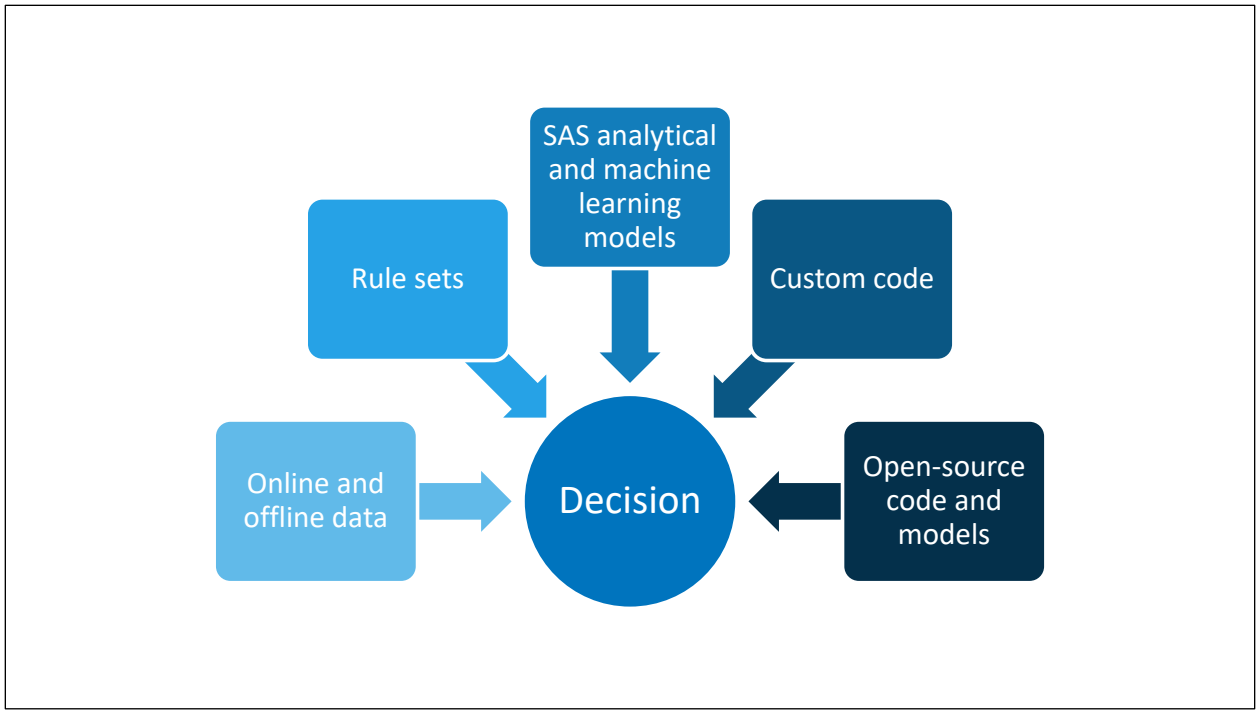


You can automate decisions extending to include virtually any operational decisioning need, including the areas listed here.

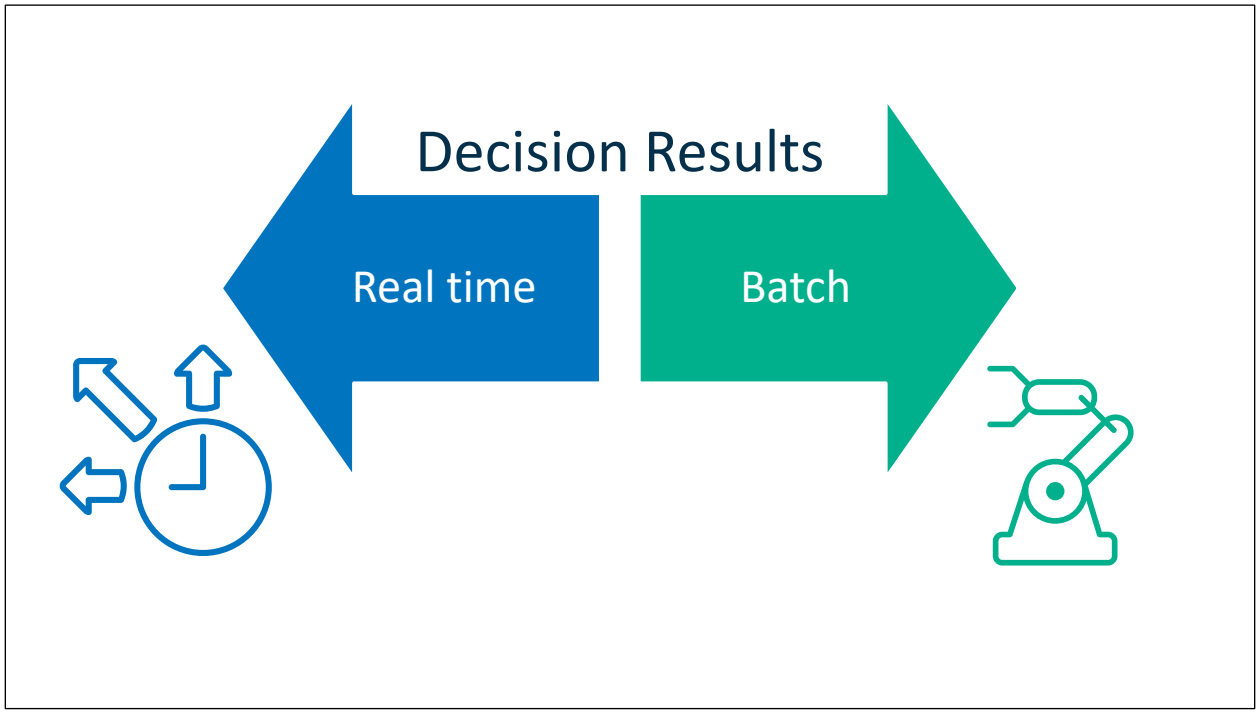


Let's look at some examples of types of decisions that could be automated using SAS Intelligent Decisioning. You could build a decision to determine whether to approve a loan, detect fraudulent transactions, or identify whether a machine is about to experience a defect. You could create a decision to identify whether a claim is valid, determine the next best action for a particular customer, or choose an appropriate price for a product.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

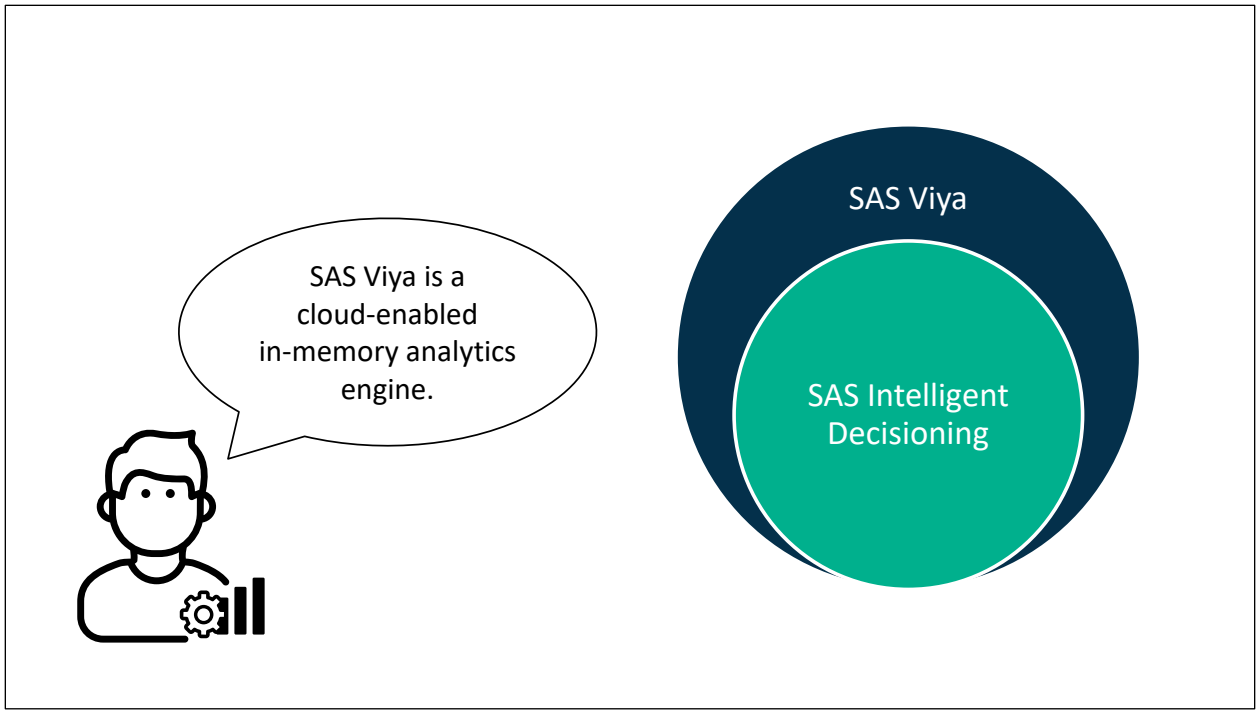


When you build a decision, you can use data captured in real time along with data stored in internal repositories. You can build and apply rule sets, which can be simple or complex. You can apply SAS analytical and machine learning models or execute custom code written to produce a result needed for the decision. You can execute open source code and models.

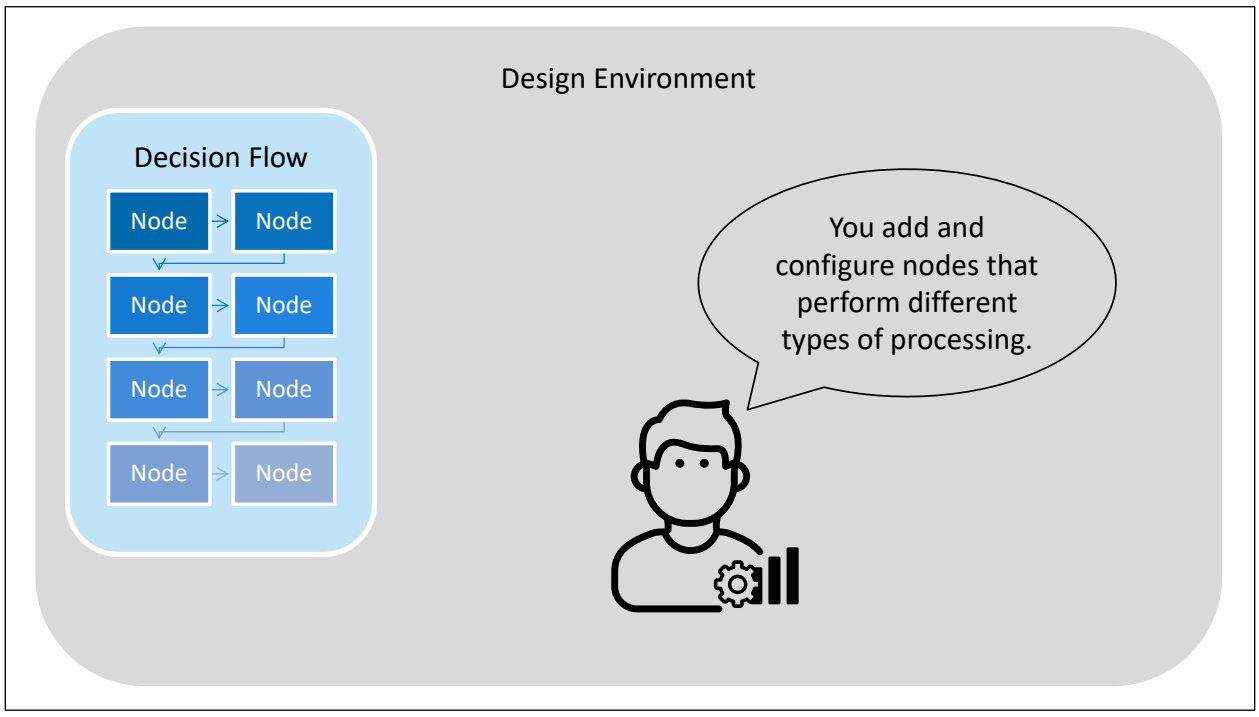


Decisions can be published for real-time or batch execution.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

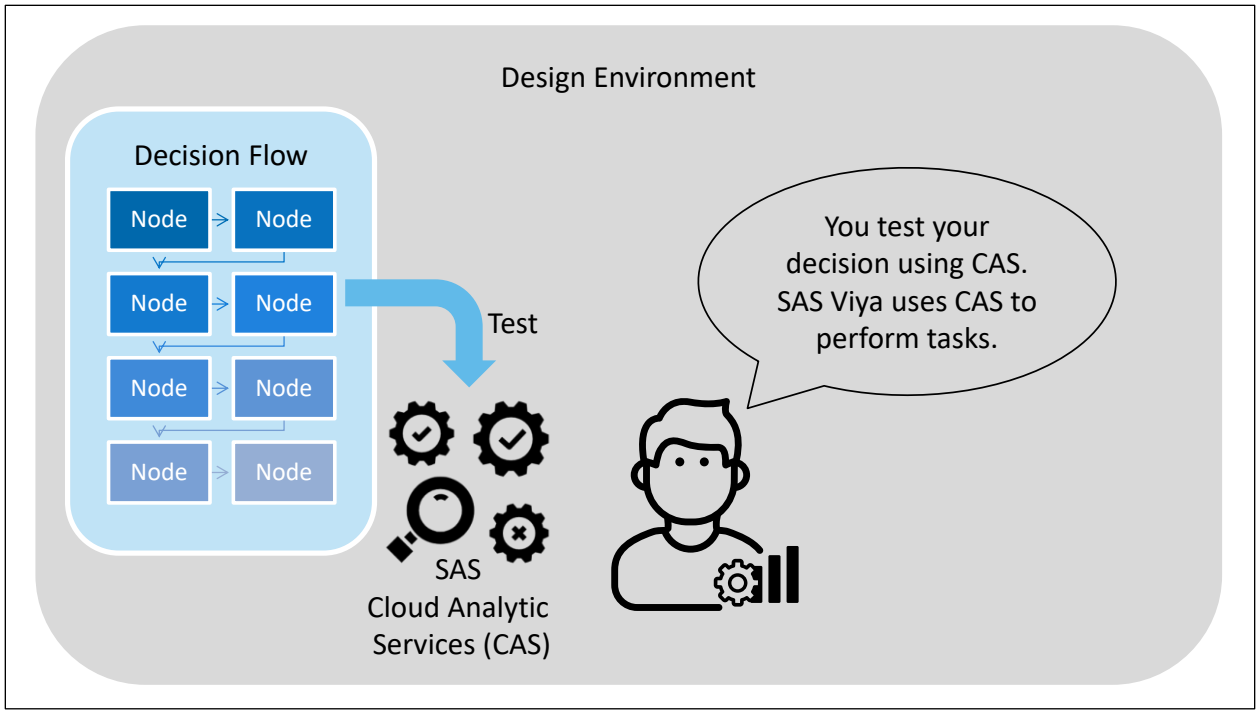


SAS Intelligent Decisioning runs in SAS Viya. SAS Viya is a cloud-enabled, in-memory analytics engine that provides quick, accurate, and reliable analytical insights.

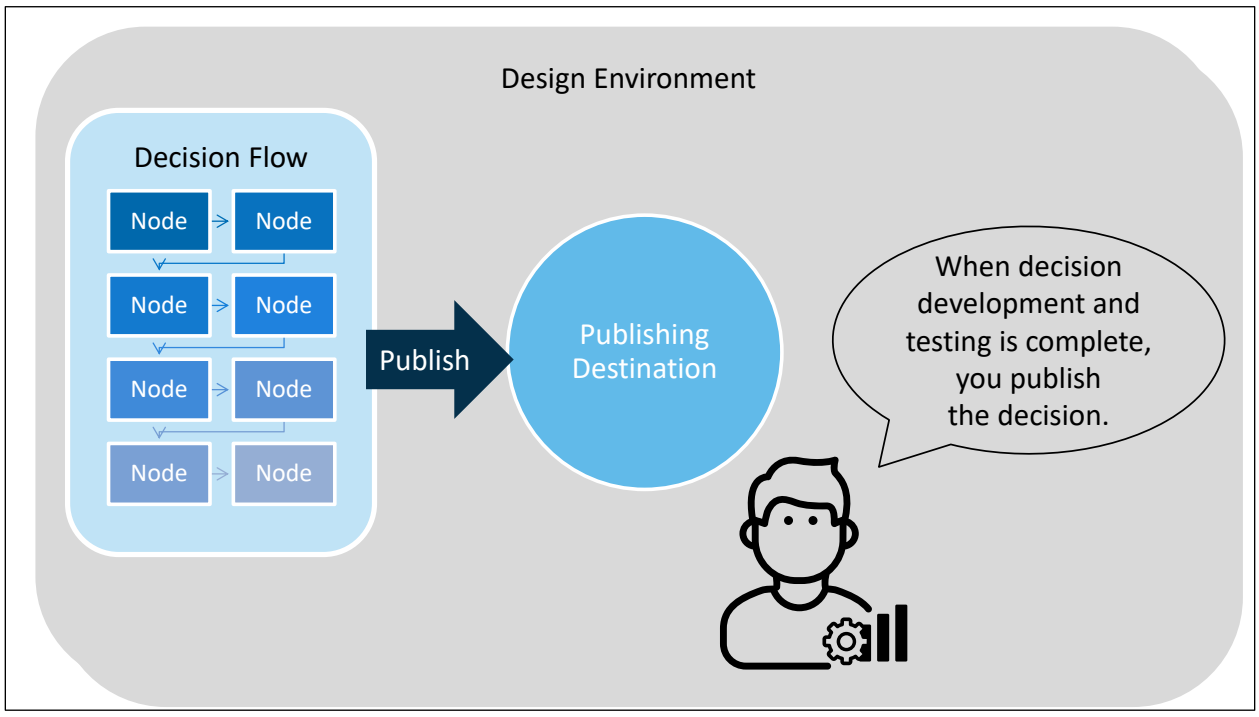


You build and test a decision in a design environment. When you build a decision, you create a decision flow. You define the steps that you want the decision to perform by adding and configuring nodes that perform different types of processing.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

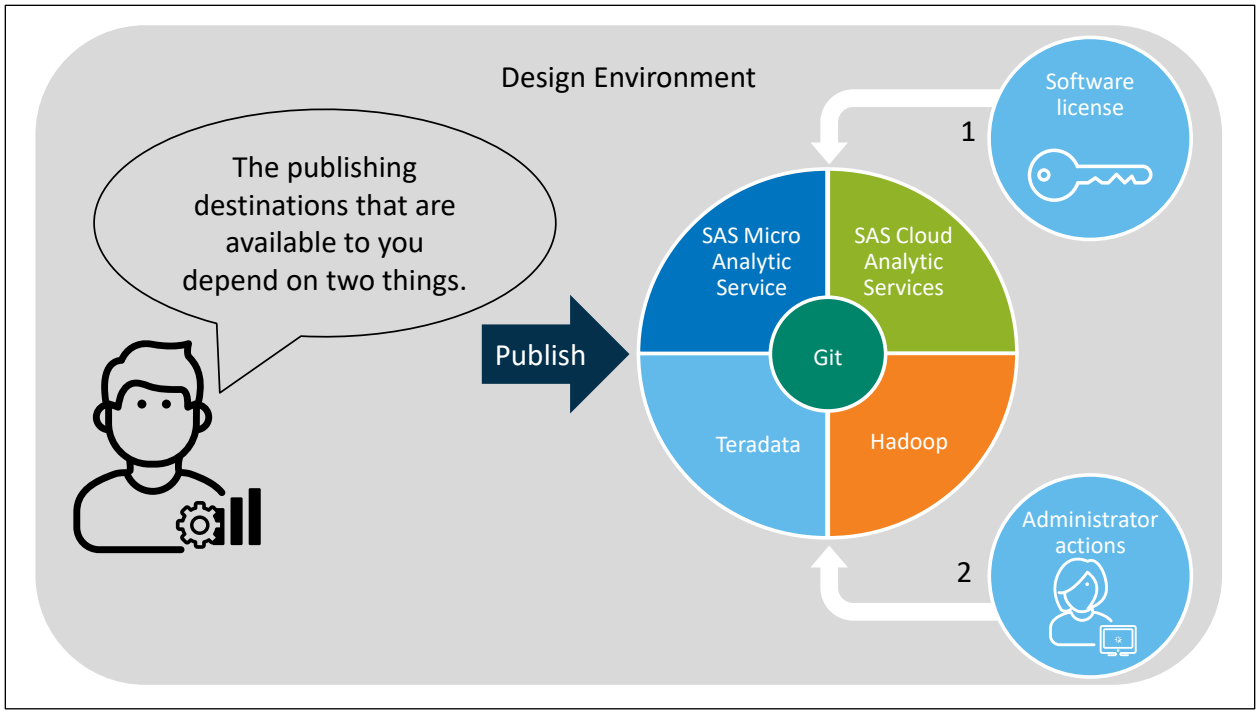


You test your decision using SAS Cloud Analytic Services, also known as CAS. SAS Viya uses CAS to perform tasks.

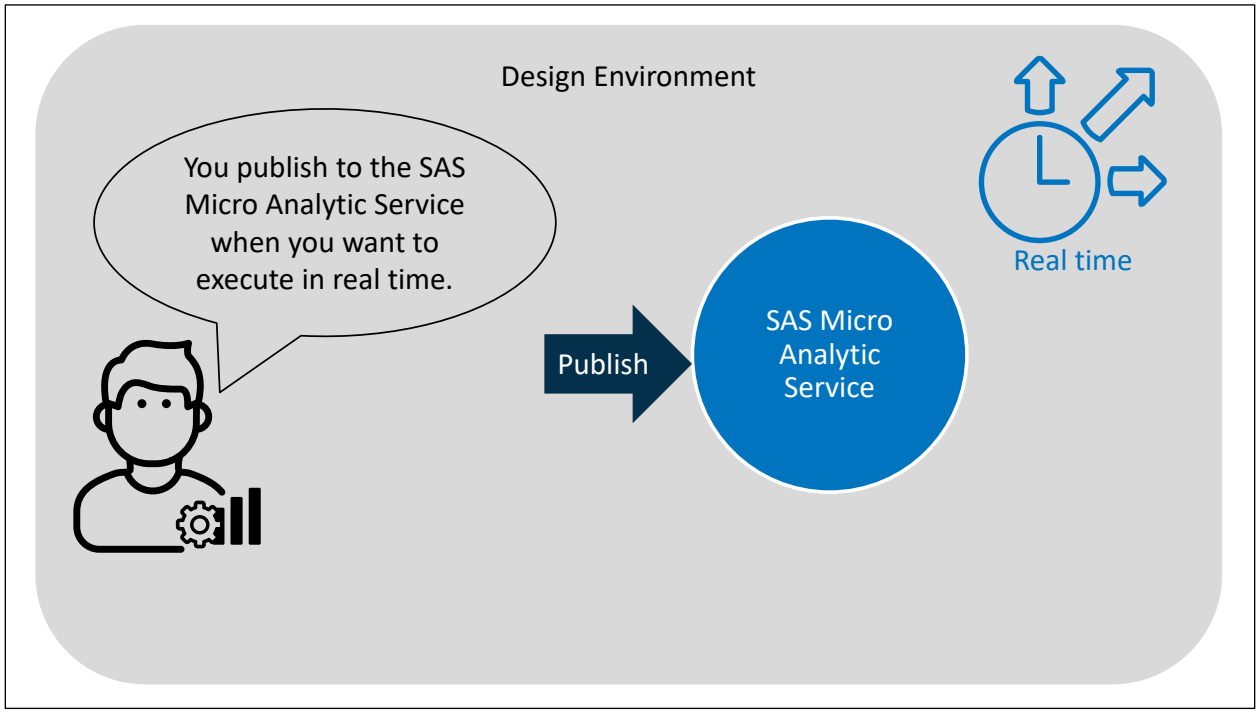


When decision development and testing is complete, you publish the decision. Publishing a decision creates an entity that can be managed and run in another environment. When you publish a decision, you choose a publishing destination.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

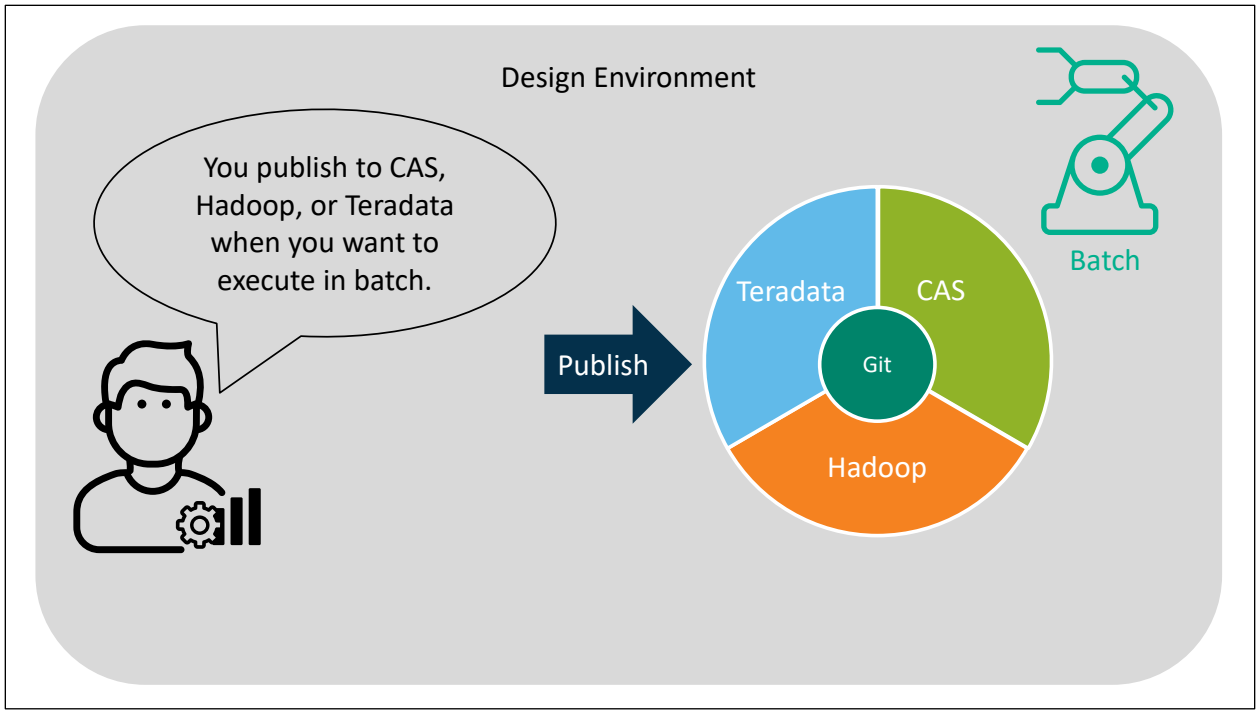


The publishing destinations available to you depend on your software license and might depend on actions taken by an administrator. You might be able to publish to one or more of these destinations: SAS Micro Analytic Service, SAS Cloud Analytic Services, Hadoop, Git or Teradata.

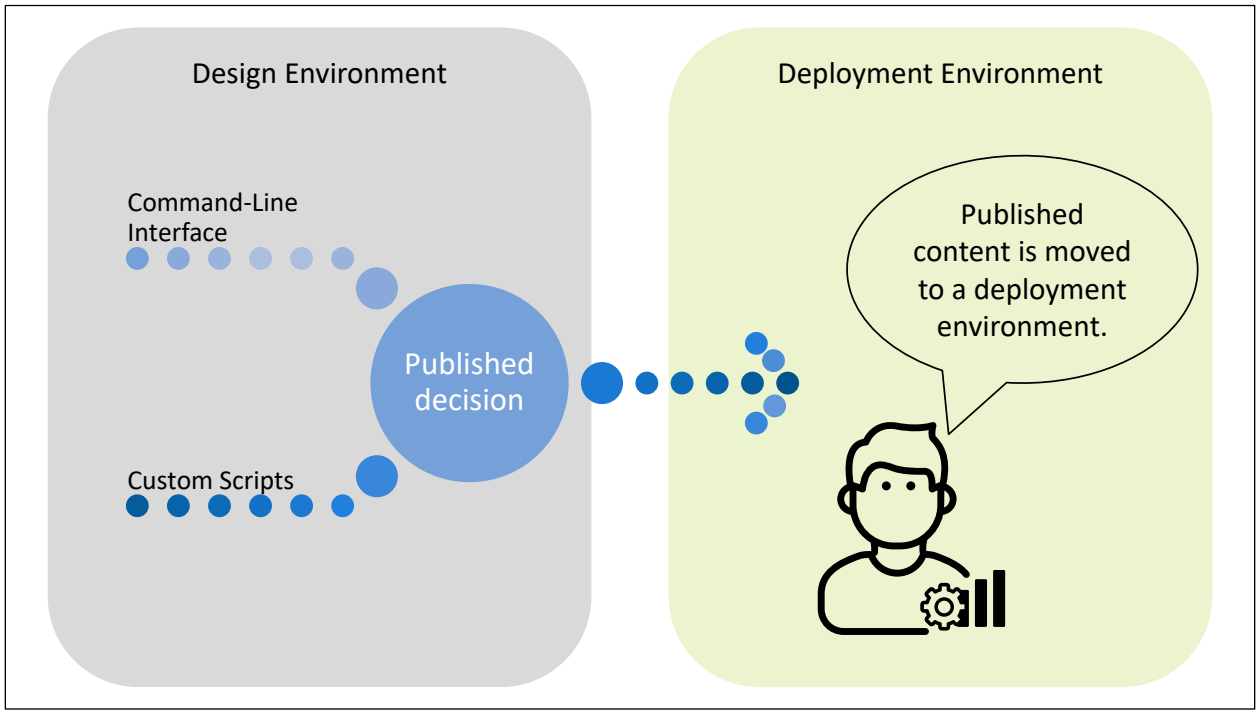


When you want to execute a decision in real time, you publish to the SAS Micro Analytic Service. The SAS Micro Analytic Service is a memory-resident, high-performance program execution service that is included with SAS Intelligent Decisioning.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

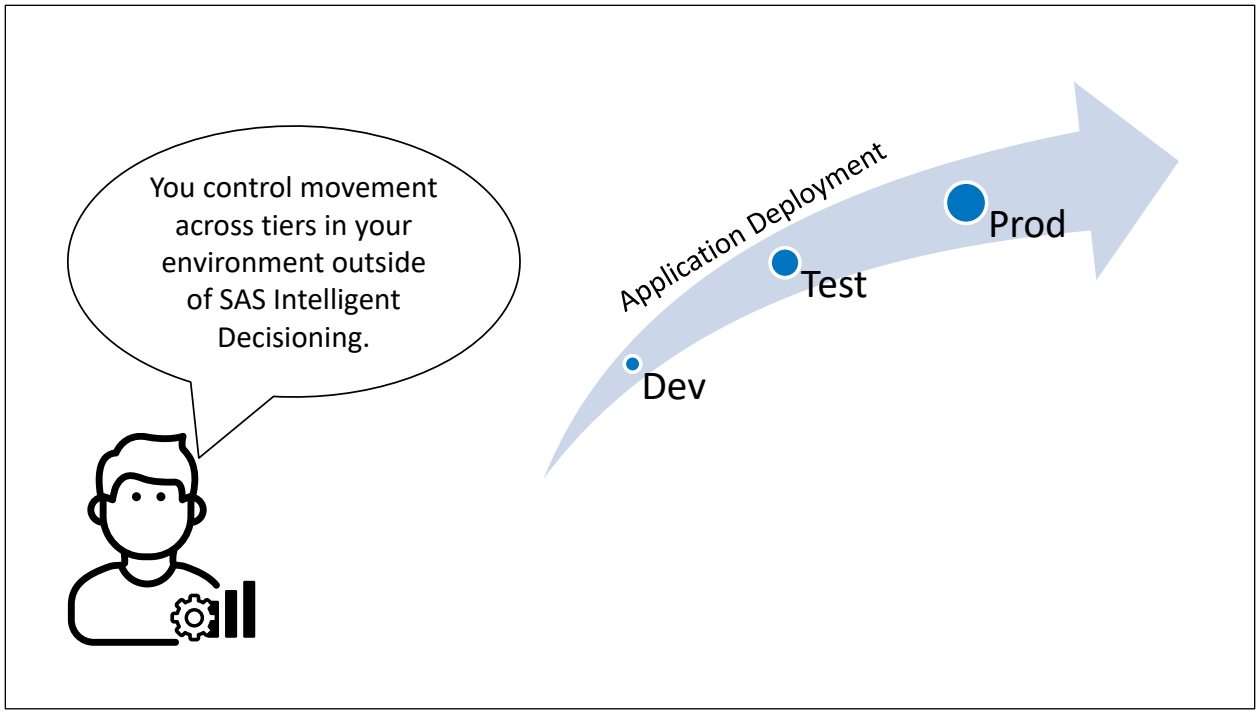


You publish to CAS, Hadoop, Git or Teradata when you want to execute in batch.

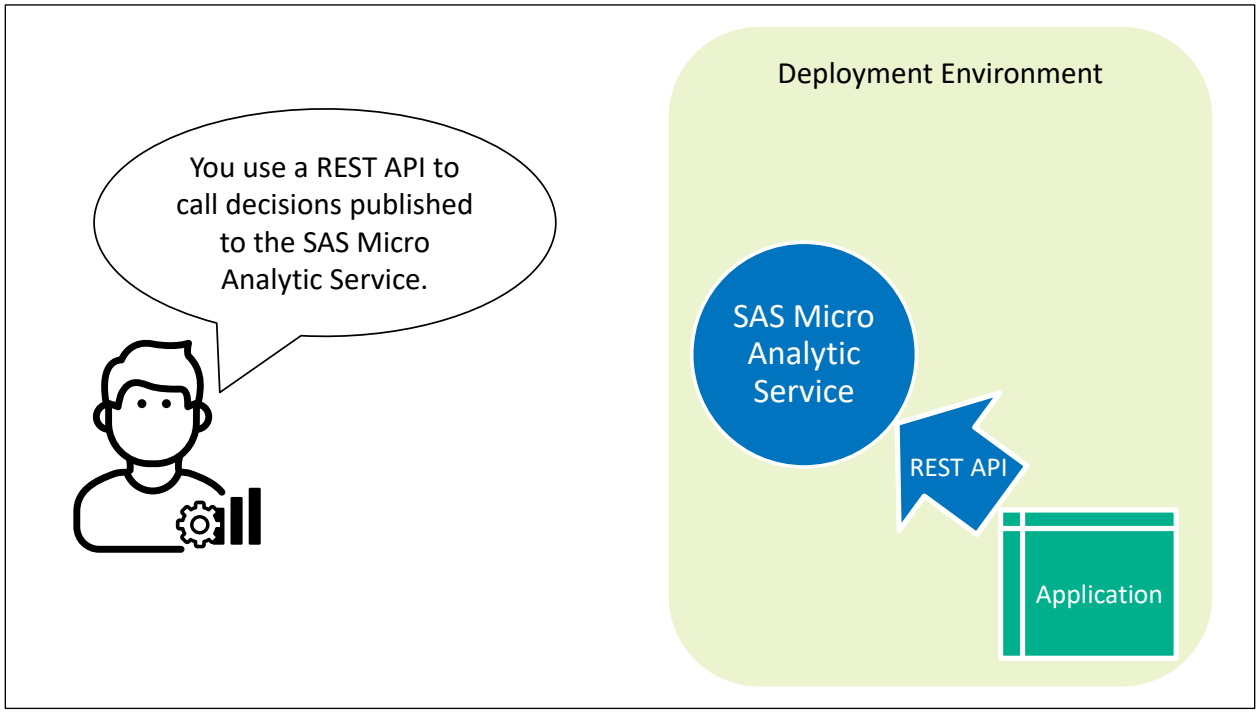


After a decision is published, the published content is moved from the design environment to a deployment environment. In the current release of SAS Intelligent Decisioning, these steps are performed using a command-line interface and custom scripts.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

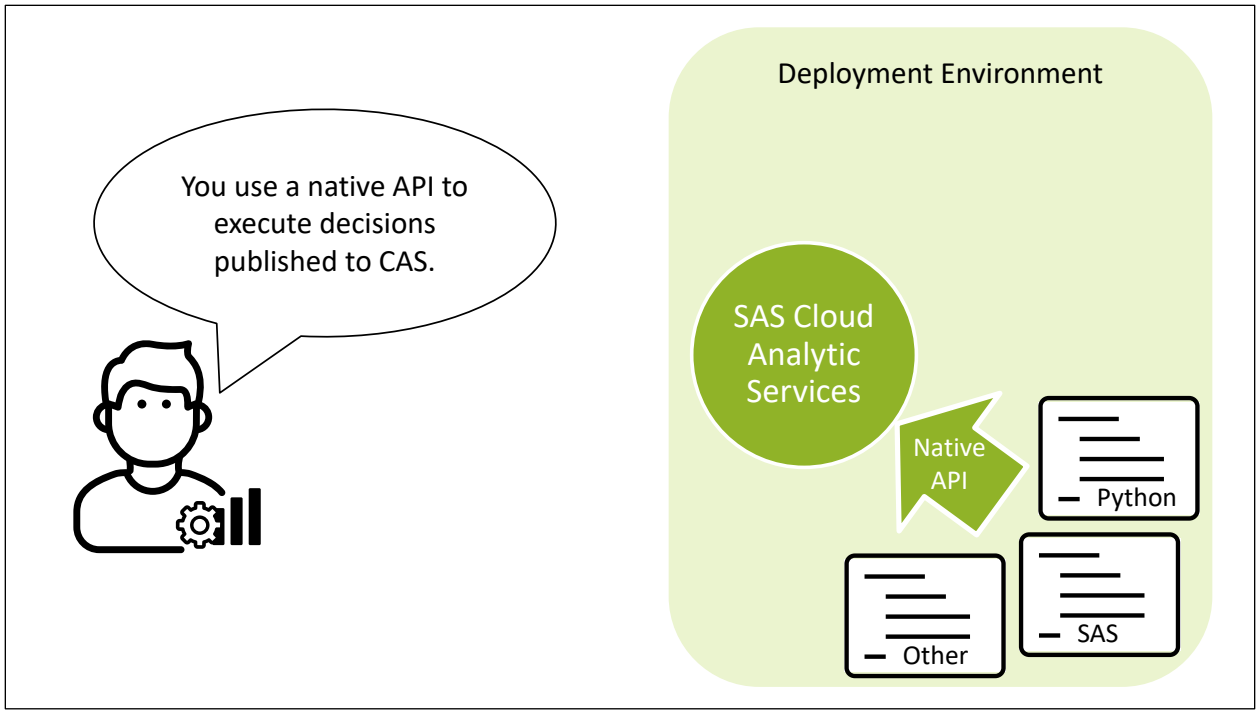


Your organization might have multiple tiers defined for application deployment (such as development, test, and production). After publishing in Intelligent Decisioning, you control movement of objects across those tiers outside of the solution.

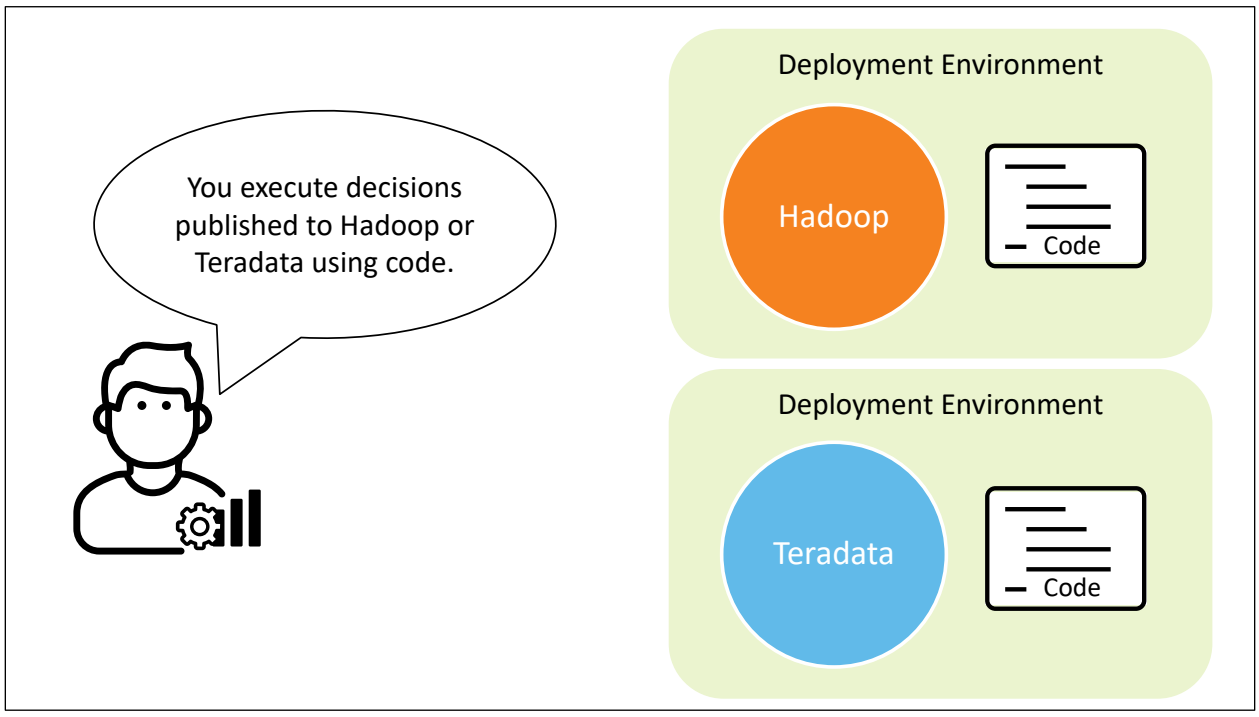


When you publish content to a SAS Micro Analytic Service destination, SAS Intelligent Decisioning creates a callable REST API endpoint, independent of Intelligent Decisioning. You use a REST API provided with Intelligent Decisioning to access the decision from other applications.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

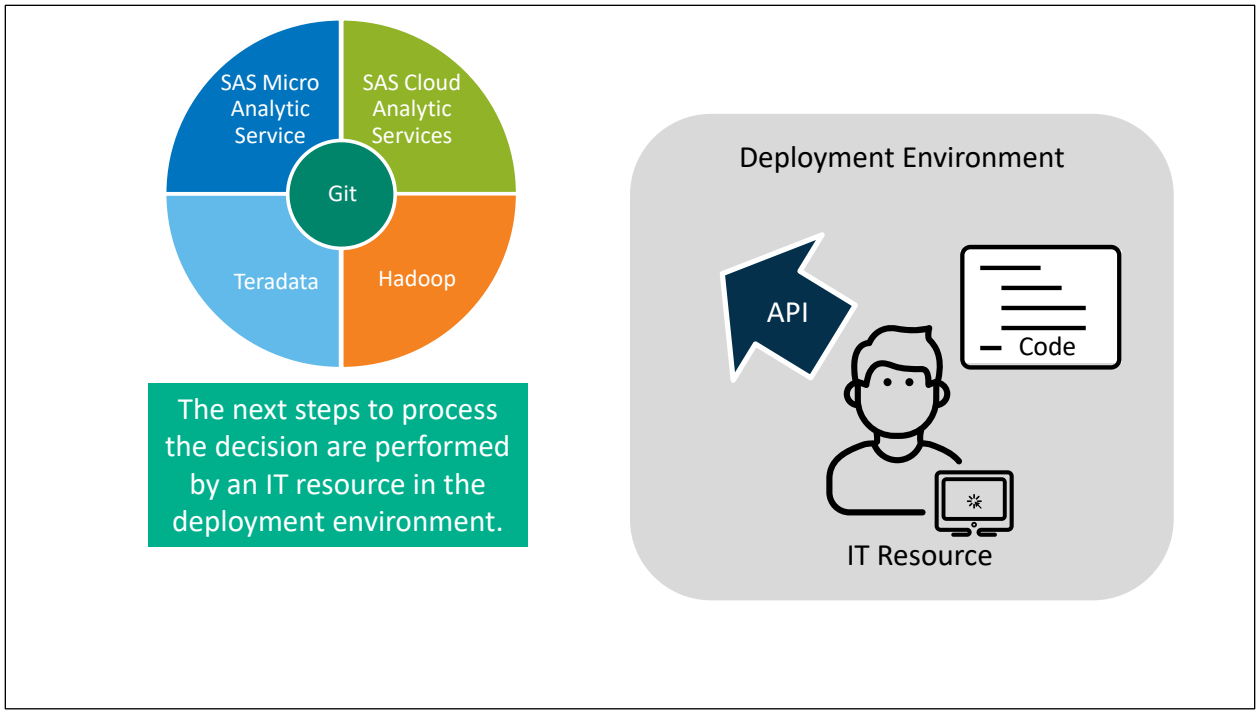


You execute a decision published to CAS by calling a native API using Python, SAS code, or other supported CAS language interfaces.

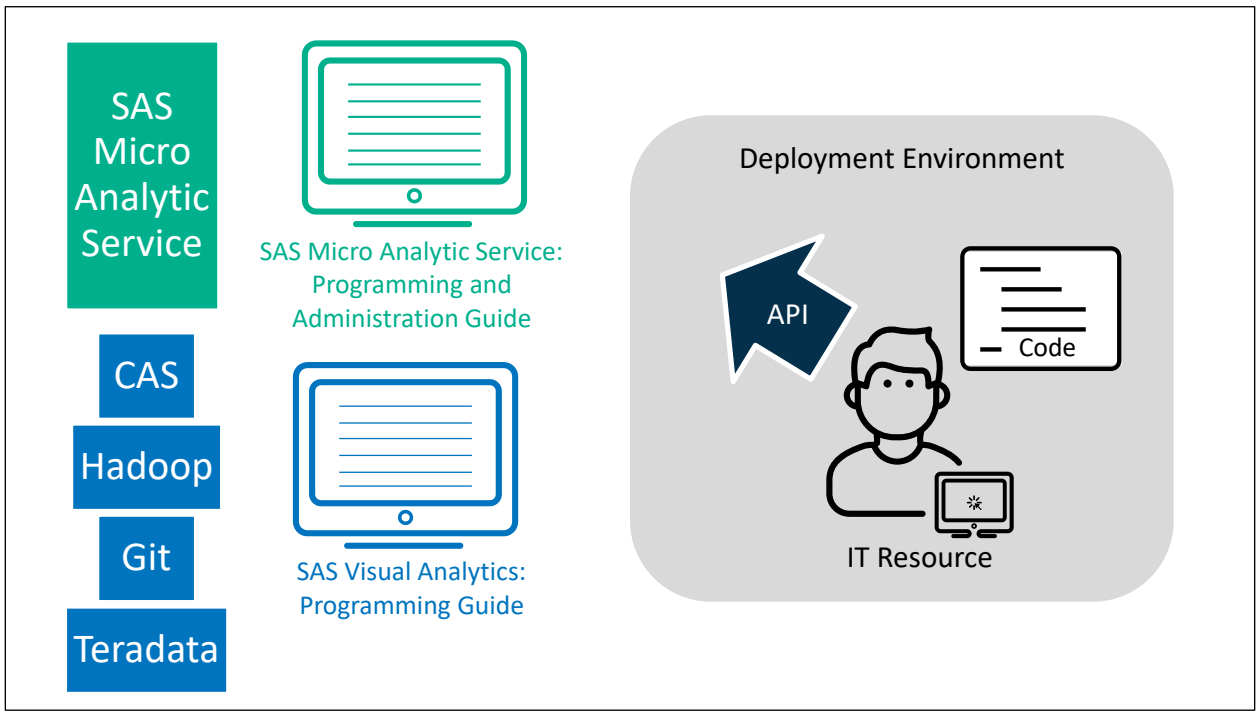


You execute decisions published to Hadoop or Teradata using code that is produced when the decision is published.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

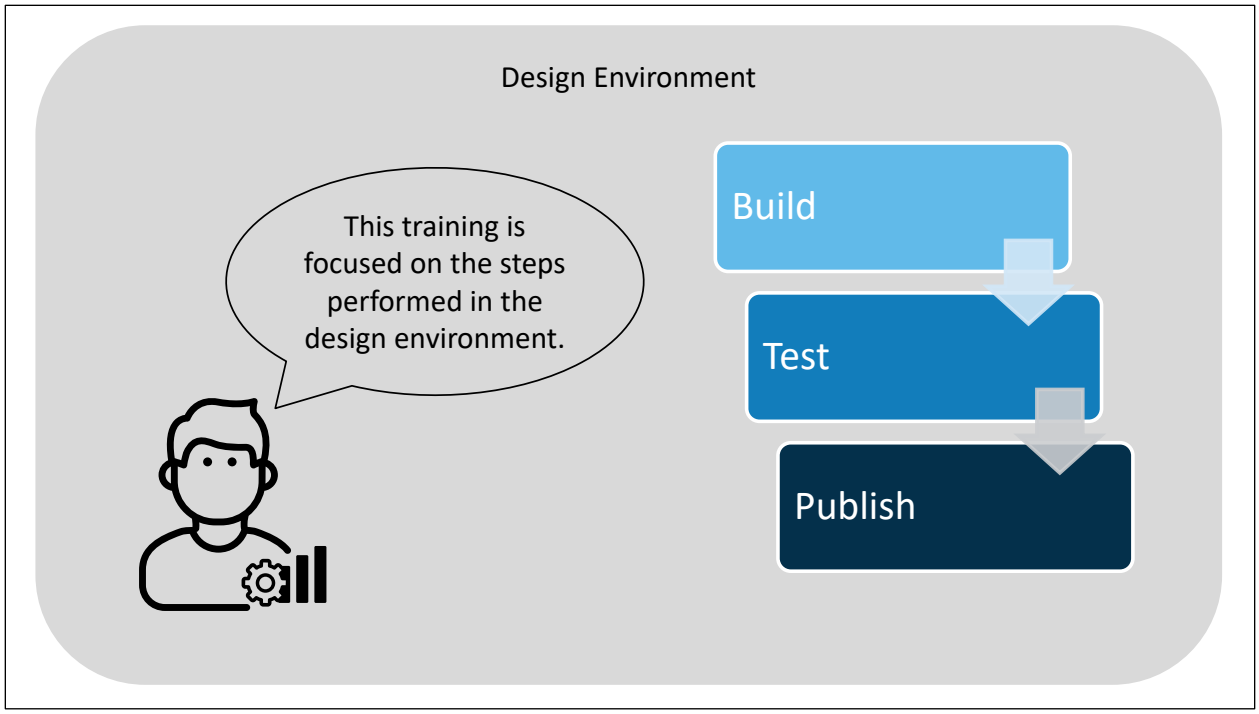


No matter which publishing destination you choose, the next steps needed to process the decision are performed by an IT resource in the deployment environment.

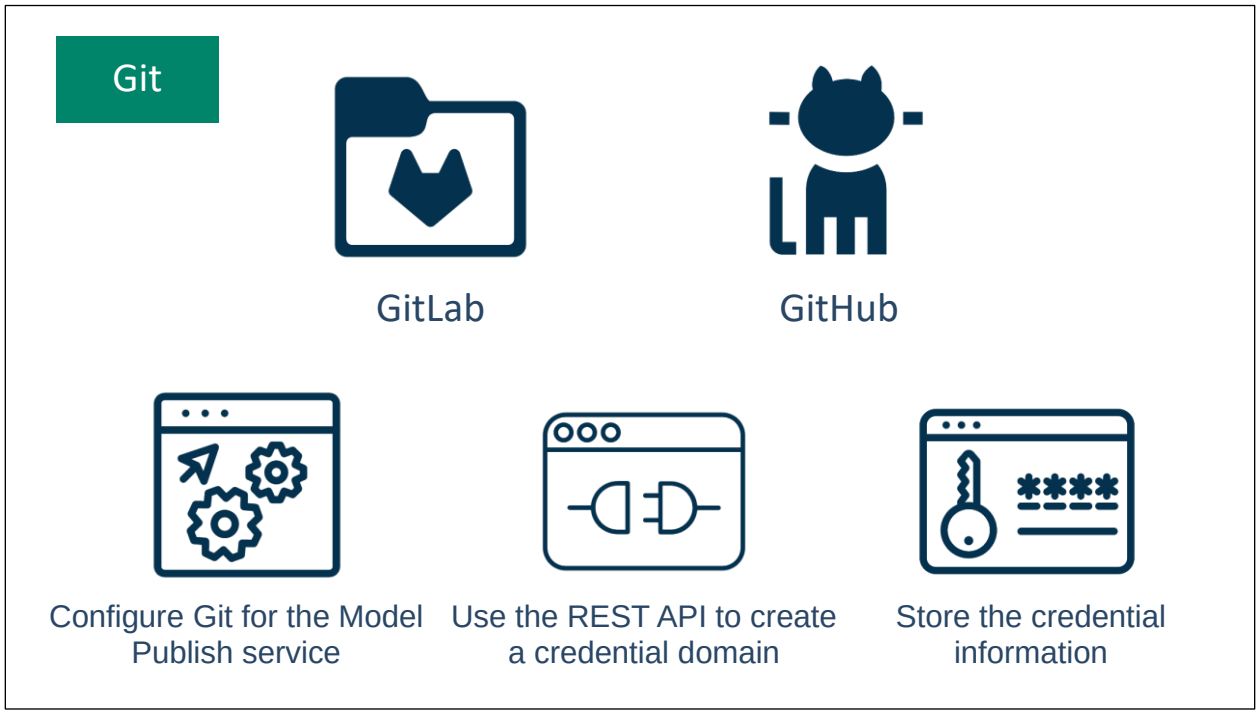


If you want to learn more about the next steps for the Micro Analytic Service, consult *SAS Micro Analytic Service: Programming and Administration Guide*. To learn more about the next steps for CAS, Hadoop, and Teradata, refer to *SAS Visual Analytics: Programming Guide*.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

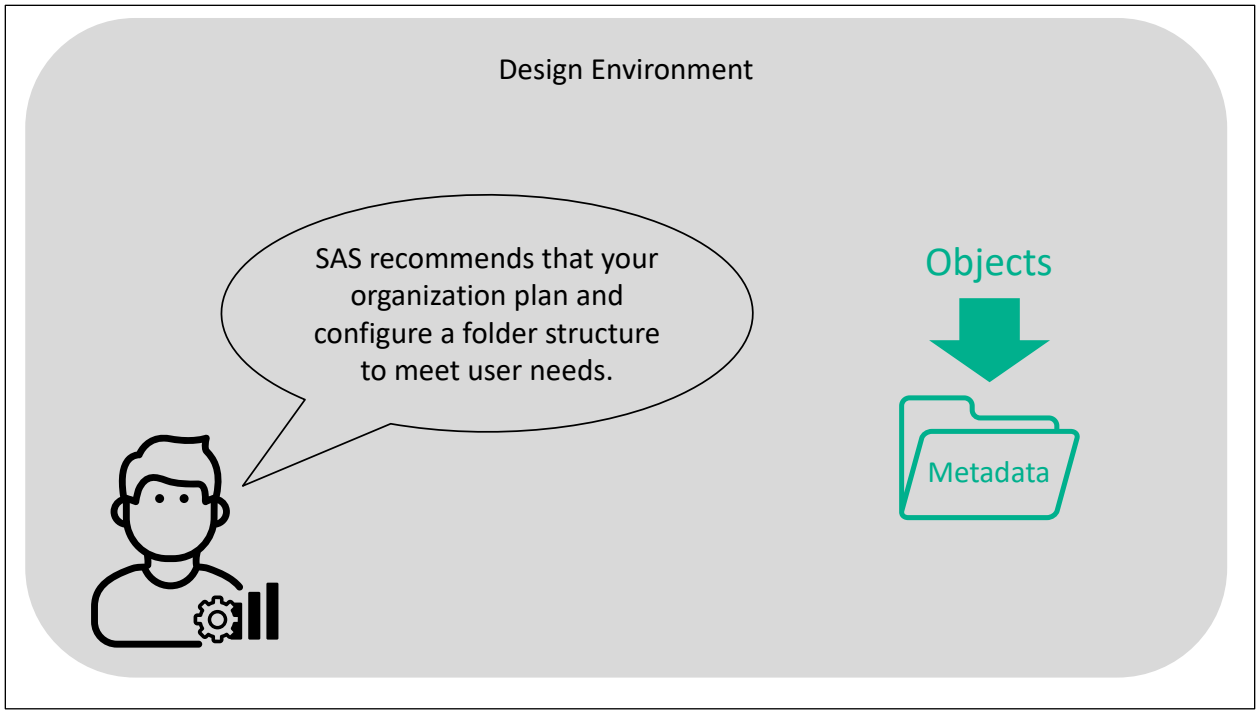


This training is focused on the steps performed in the design environment.

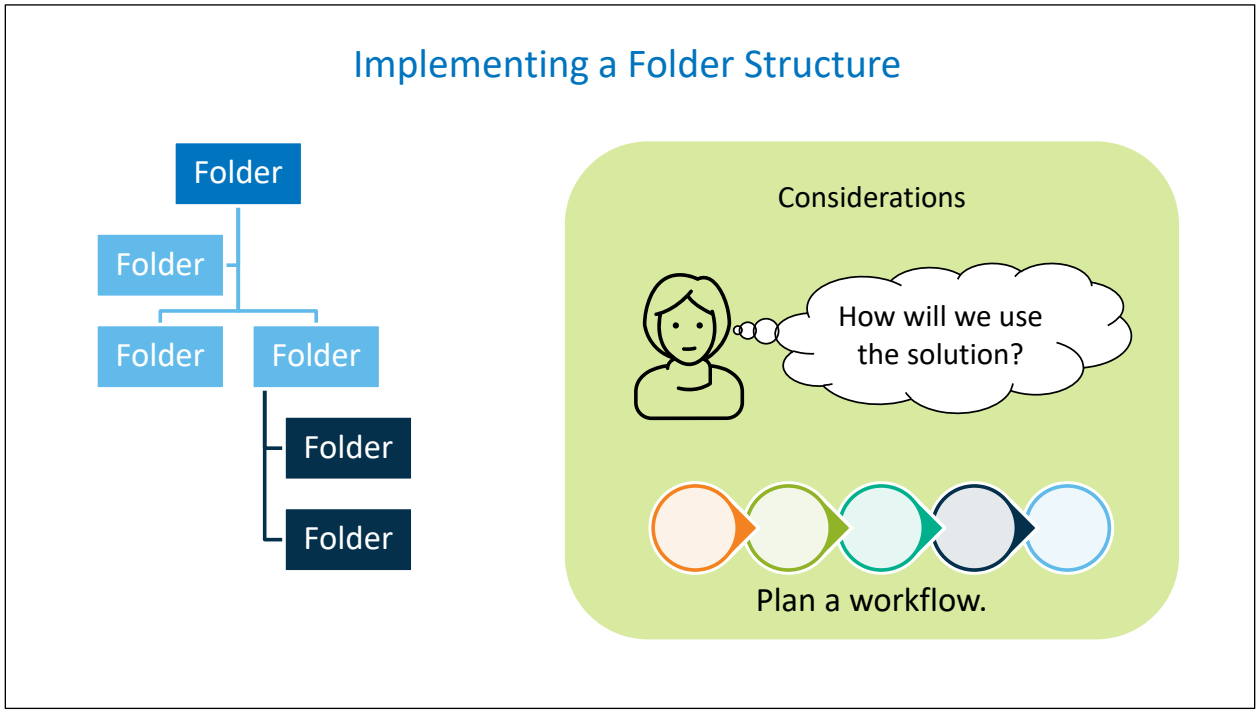


You can define a publishing destination for a Git repository that is located locally, or that is located on GitLab or GitHub.Before you can define a Git publishing destination, you must: 1. Configure Git for the Model Publish service. 2. Use the REST API to create a credential domain in the SAS Credentials service. 3. Store the credential information for accessing your cloud provider under the domain.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

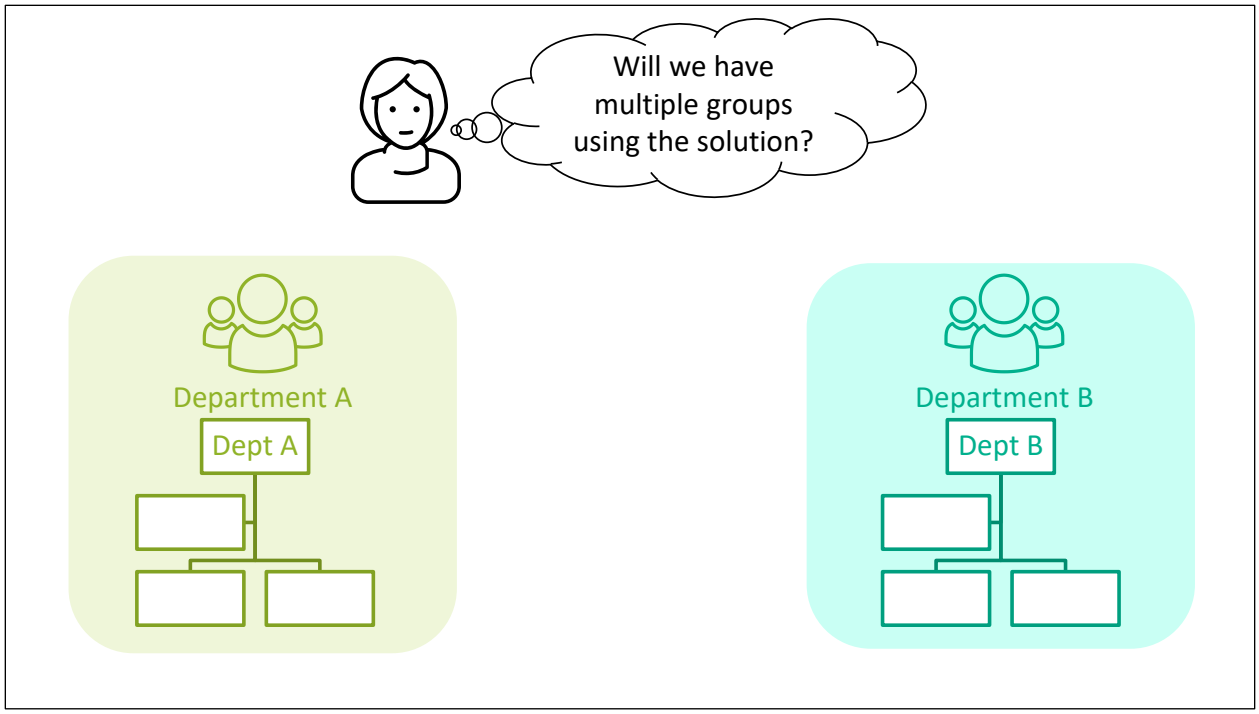


When you create objects in the design environment, you save them in metadata folders. SAS recommends that your organization plan and configure a folder structure to best meet the needs of individual users before beginning to work with SAS Intelligent Decisioning.

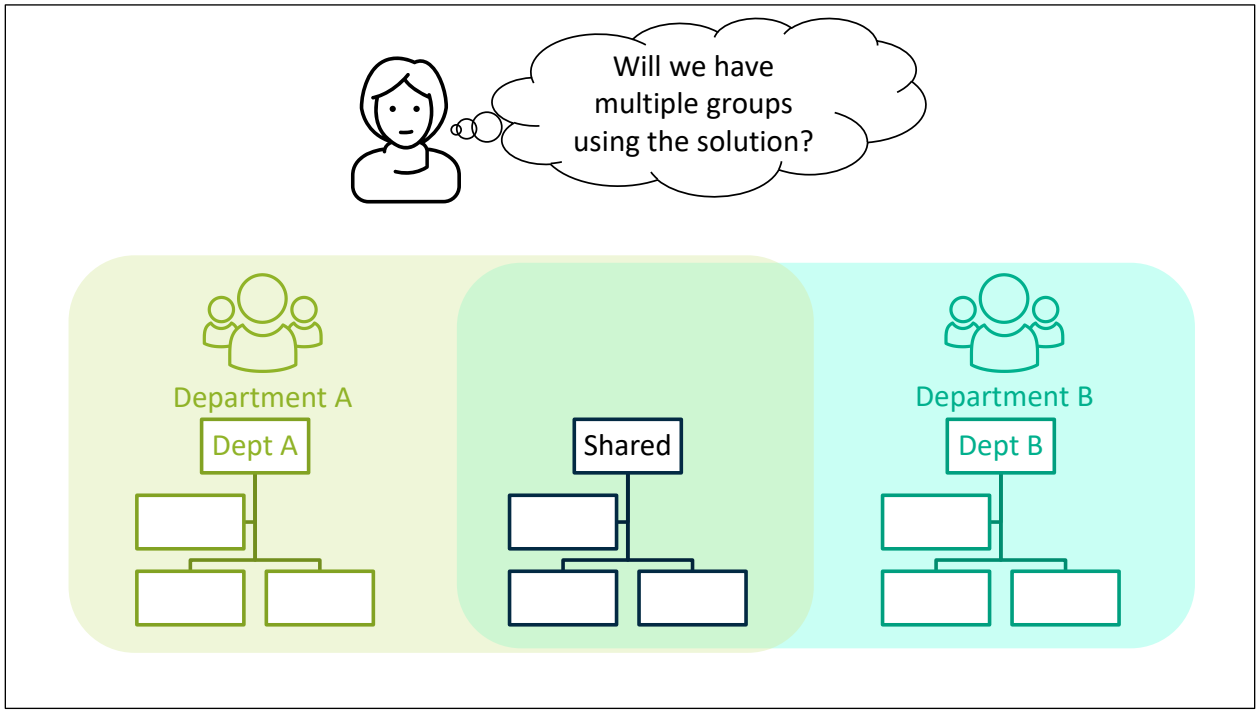


When implementing a folder structure for SAS Intelligent Decisioning, you should consider how you will use the solution at your site and plan a workflow for decision builders using the solution.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

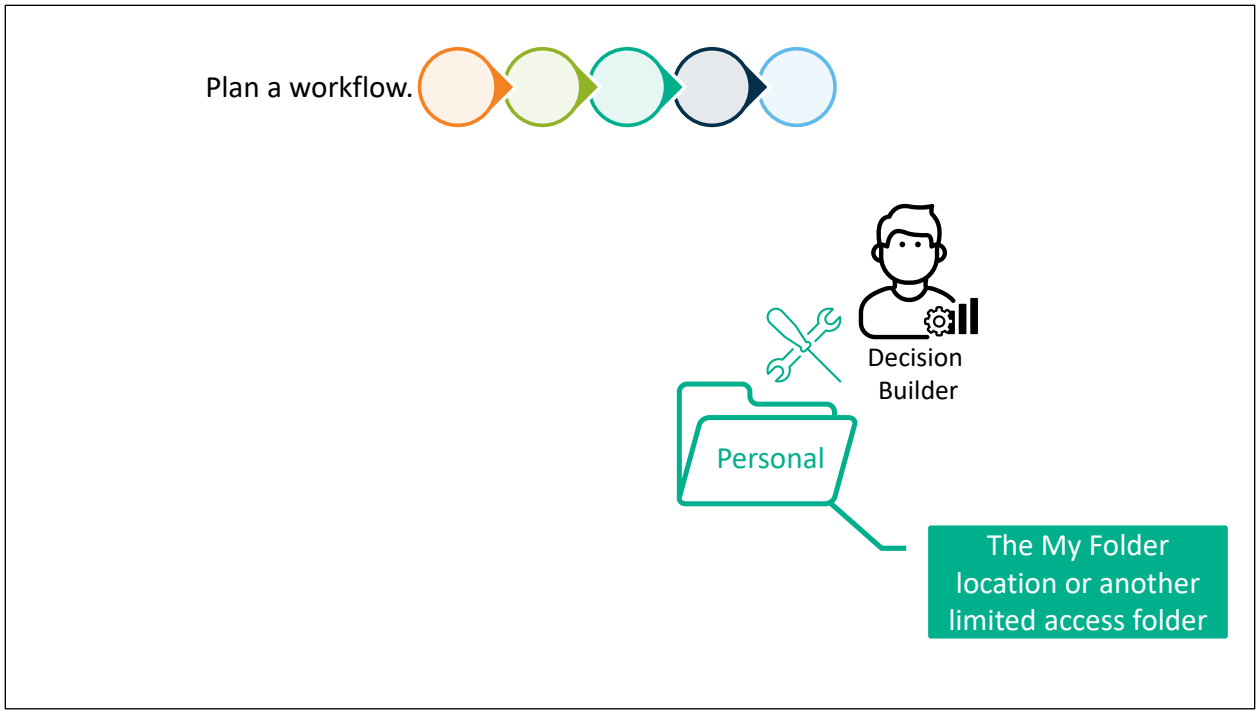


One consideration is whether you will have multiple groups using the solution. In this situation, you might want to set up separate folders for each group and grant folder access only to members of the appropriate group.

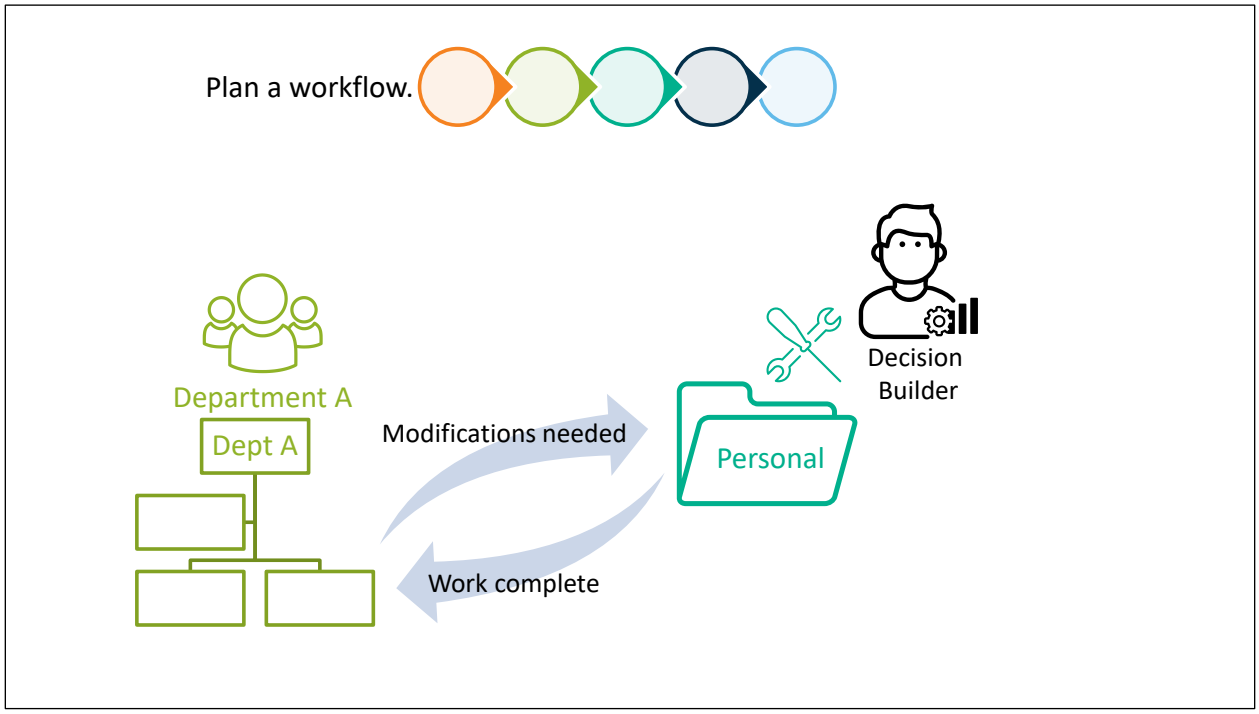


If the groups have some objects that they share, you could set up shared folders as well.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

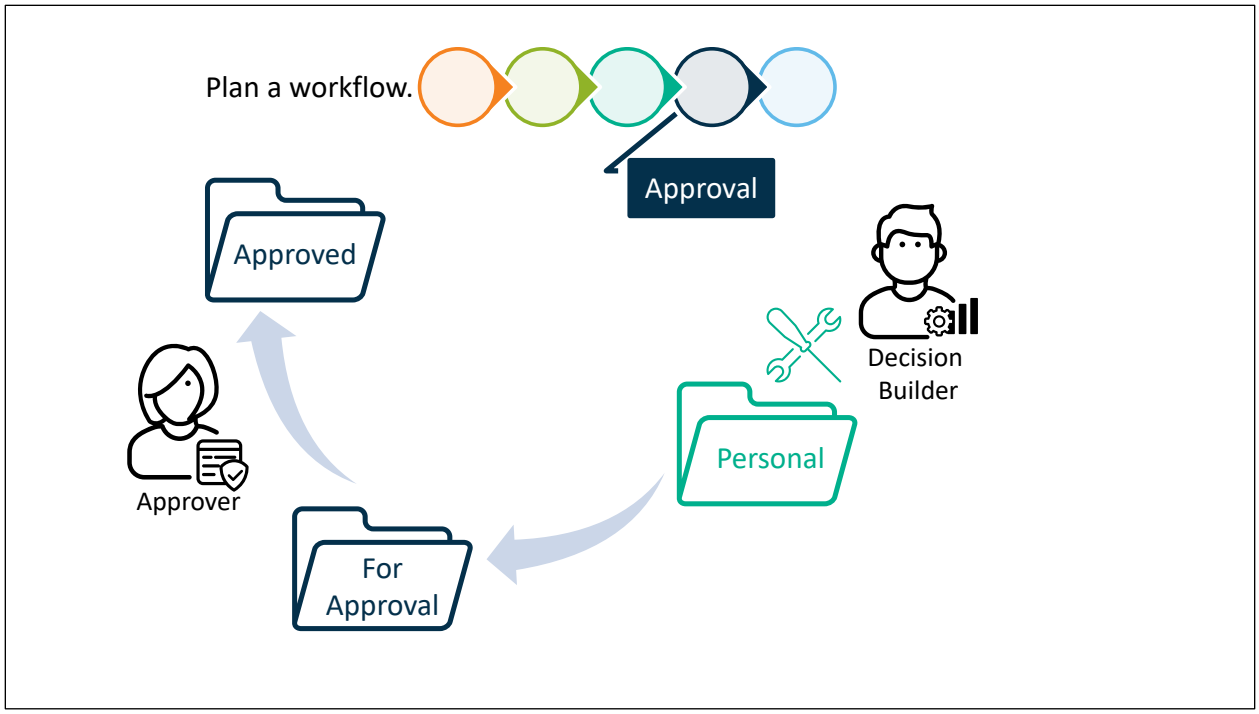


In terms of the workflow for individual users, they might actively work with objects stored in a personal or limited access folder to limit who else can see or work with the object. This folder could be the My Folder location that is set up automatically for each user and is accessible only to that user and administrative users. Or it could be another folder created for the purpose.

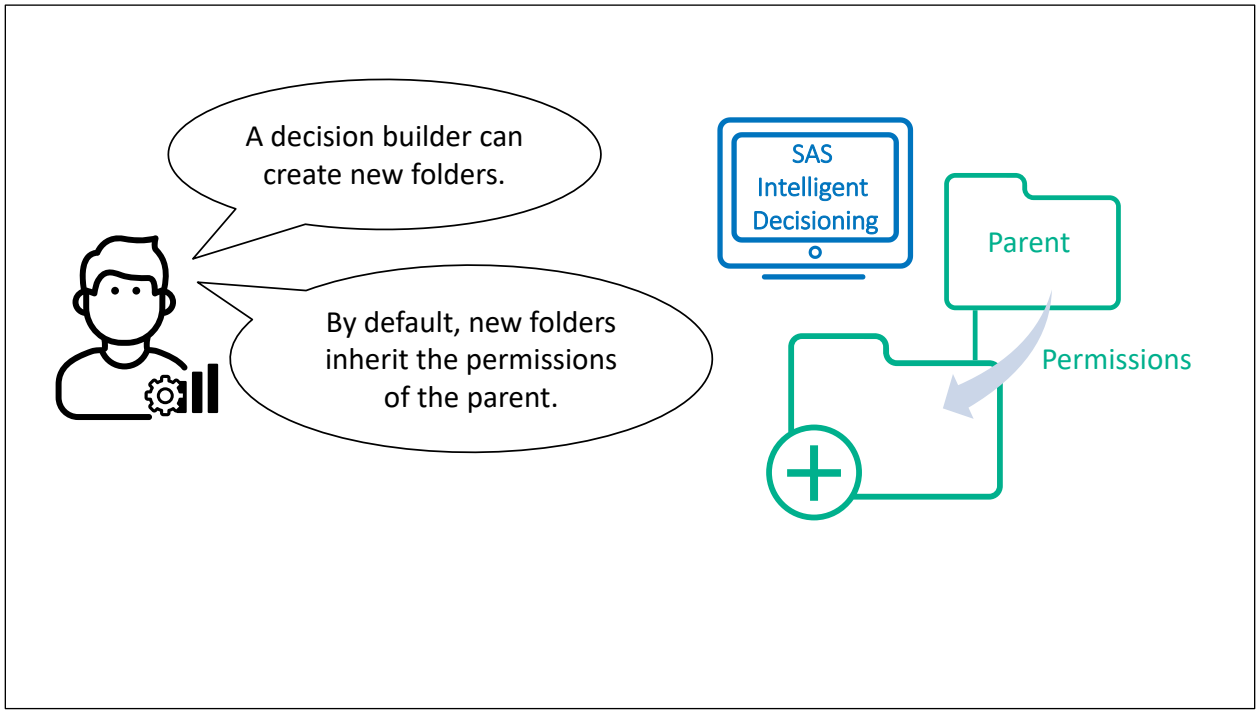


When their work is complete, they could move the object to an appropriate shared folder. Later, if they need to modify the object, they could move it back to the personal folder.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

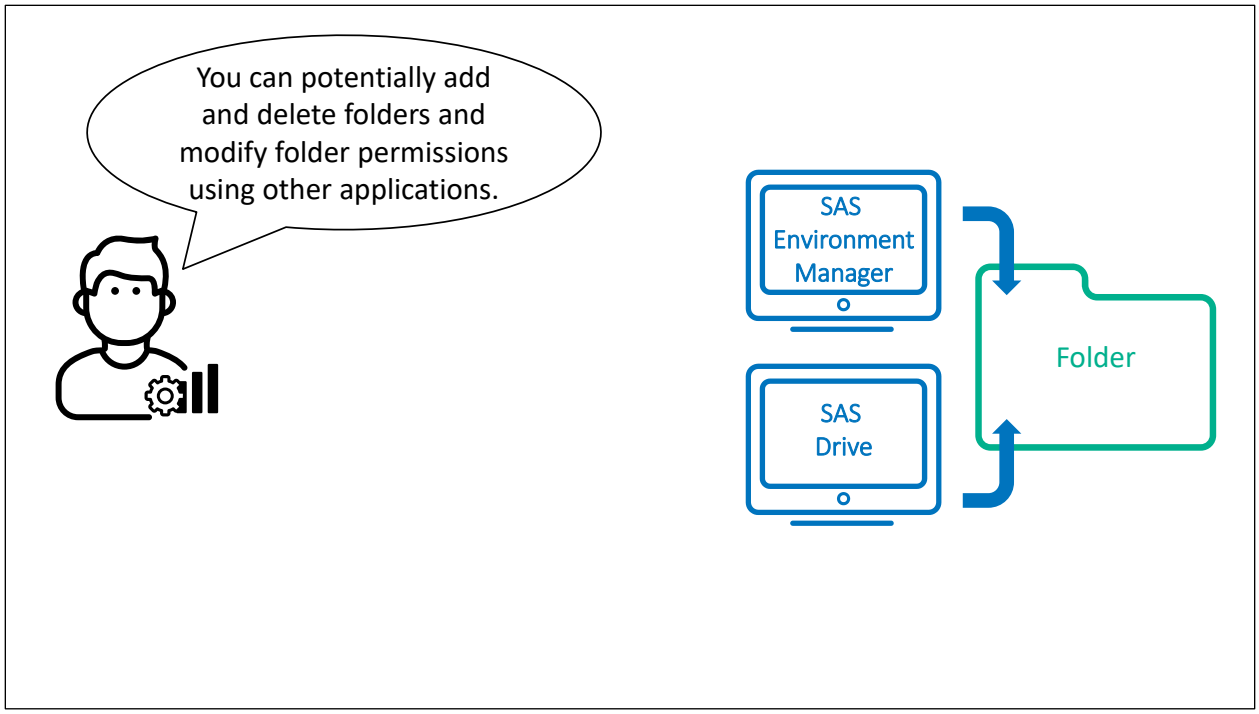


If your workflow includes an approval step, you might want to set up a folder structure to facilitate this process. When an object is complete and ready for approval, the decision builder moves it to a folder created for this purpose. An approver can then move the decision to a folder set up to contain approved objects. Only approvers have Write access to the approved folder.

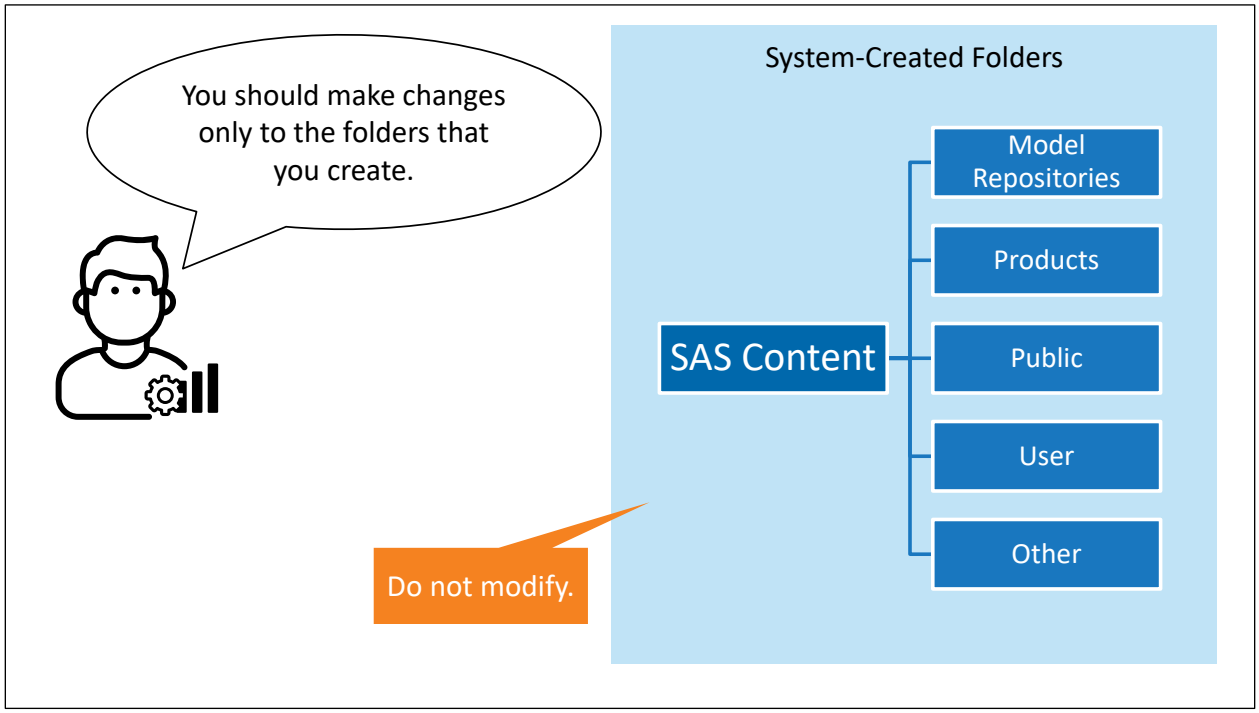


A decision builder can create new folders in SAS Intelligent Decisioning. By default, new folders inherit the permissions of the parent folder.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



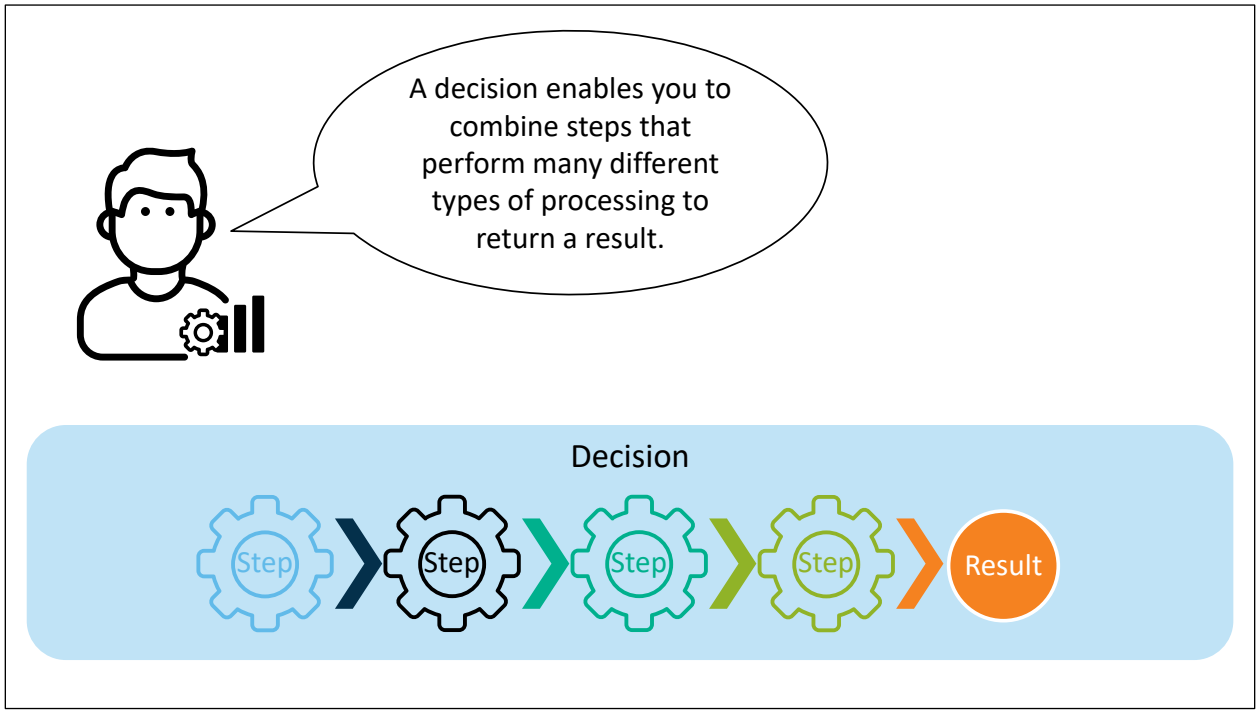
You can potentially add and delete folders and modify folder permissions using other applications such as SAS Environment Manager or SAS Drive. Your access to these applications and specific application capabilities is controlled by an administrator.



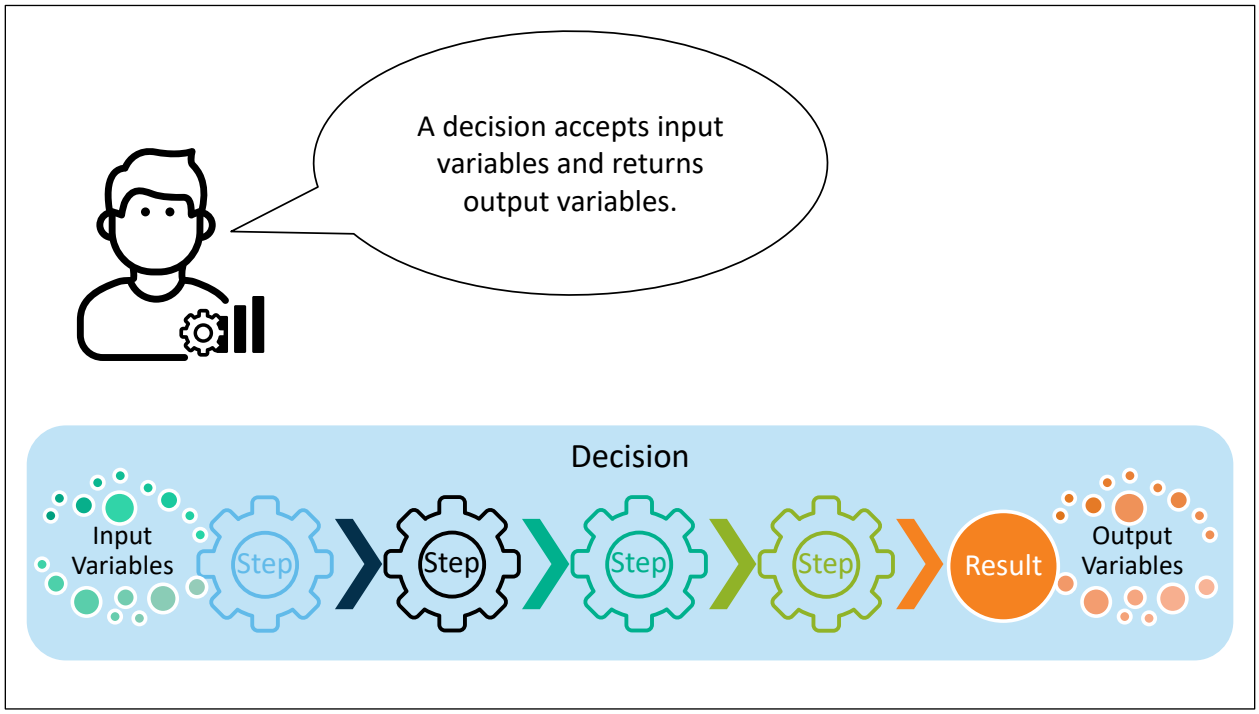
There are many system-created folders such as the Model Repositories, Products, Public, and User folders. These folders have special properties that should not be modified. You should make changes only to the folders that you create.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

1.2 Introduction to Decisions

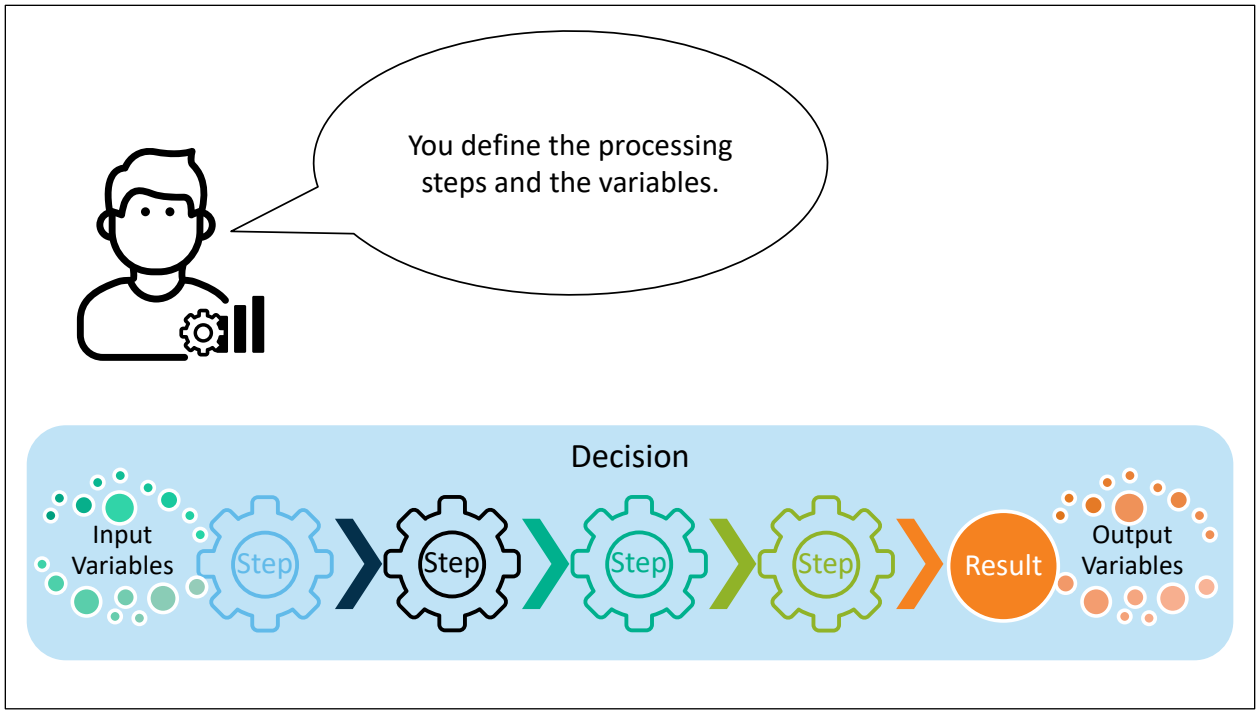


A decision enables you to create a single process that can combine steps that perform many different types of processing to return a result.

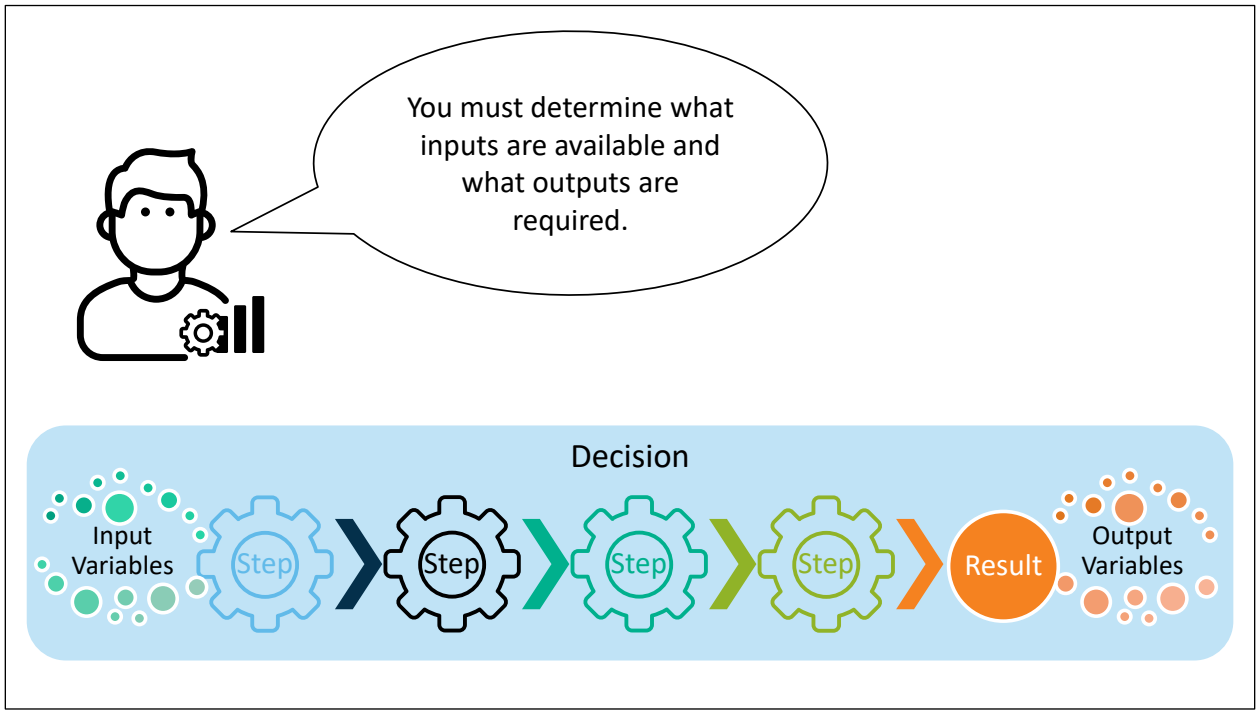


A decision accepts input variables and returns output variables.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

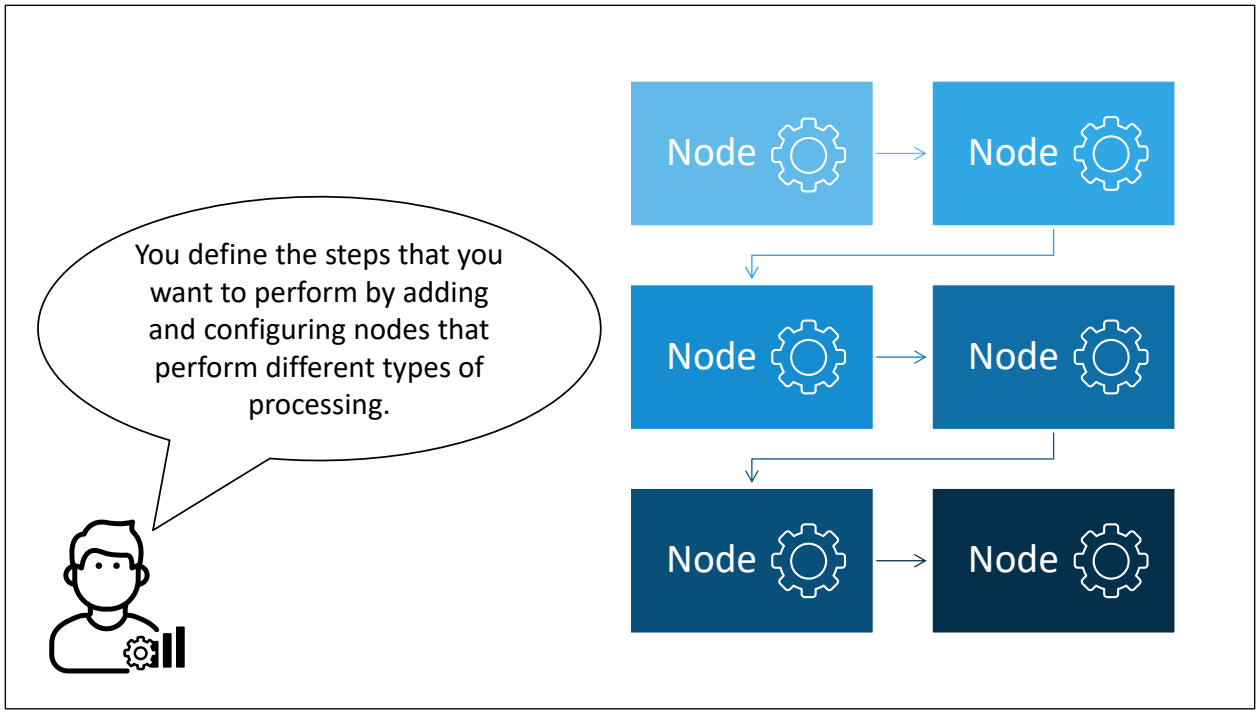


You define the processing steps and the variables.

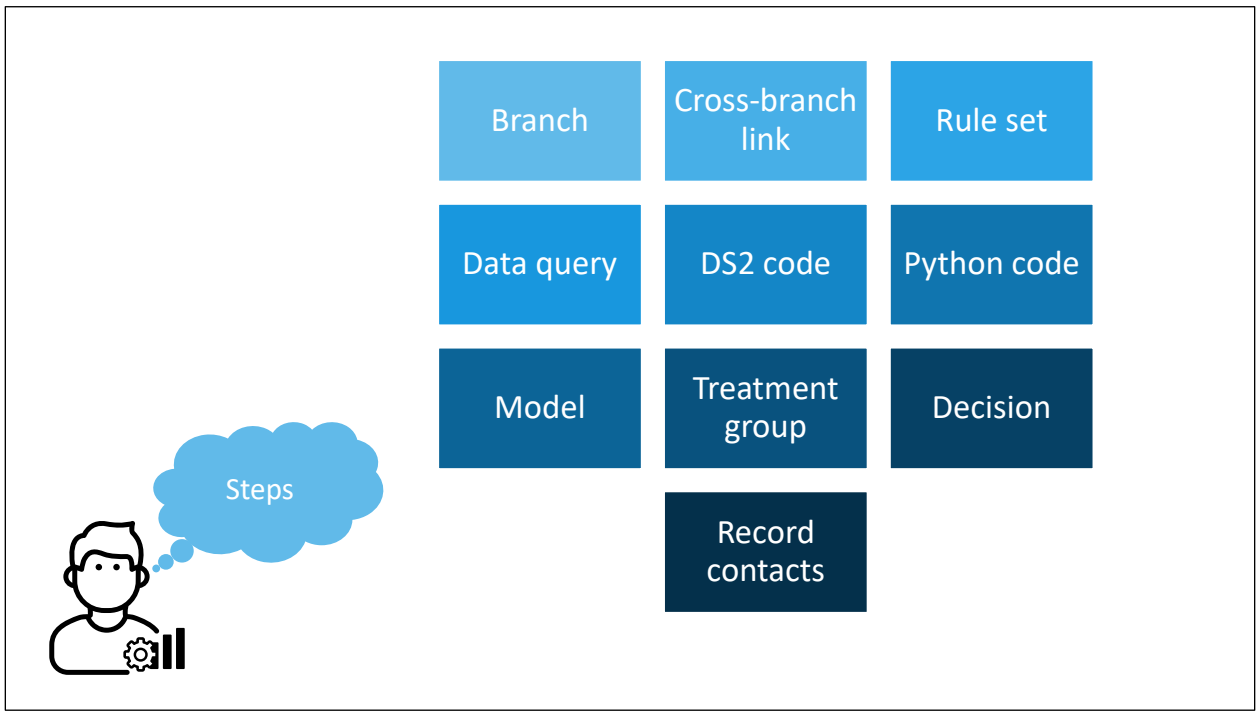


You begin by determining what inputs are available and what outputs are required.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

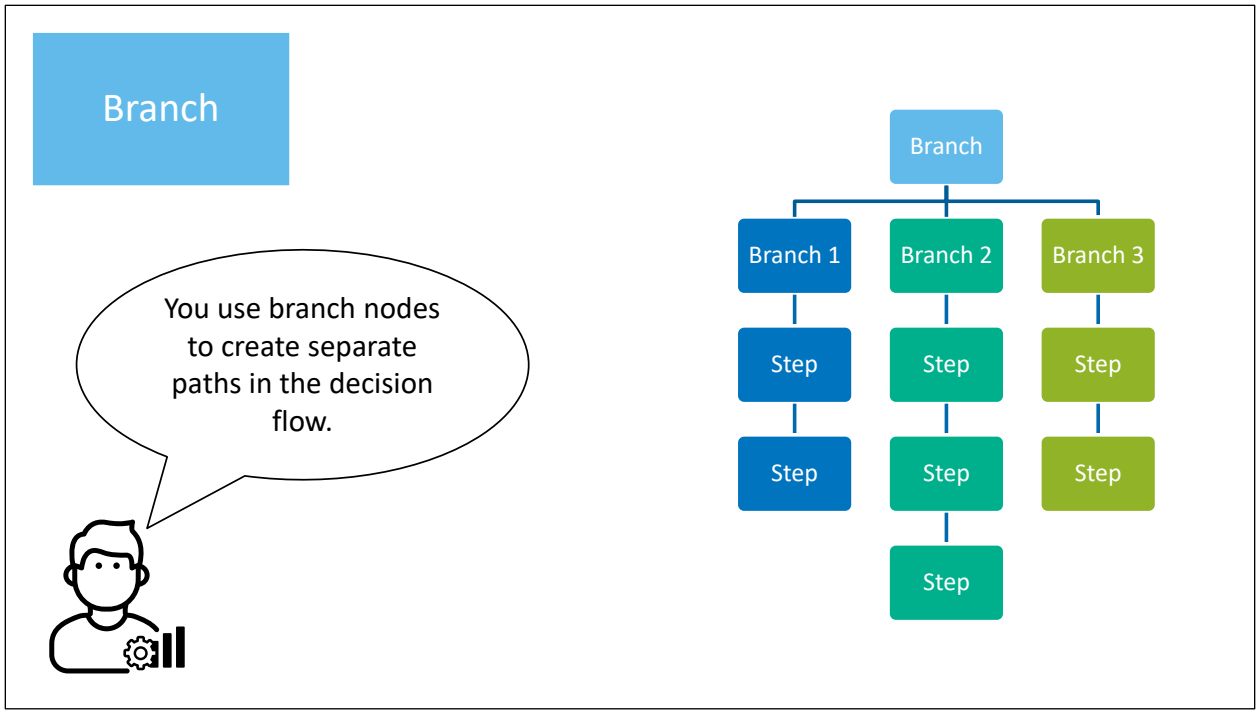


When you build a decision, you create a decision flow. You define the steps that you want to perform by adding and configuring nodes that perform different types of processing.

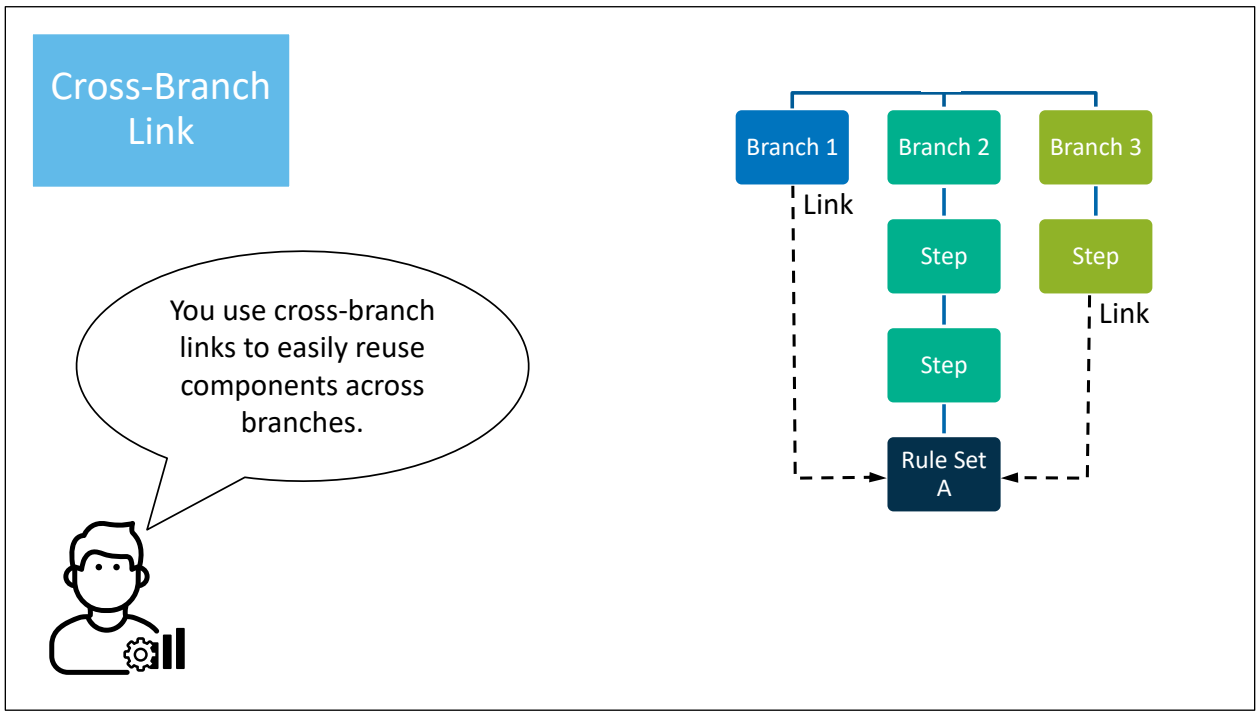


You can include nodes that perform many types of processing in a decision.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

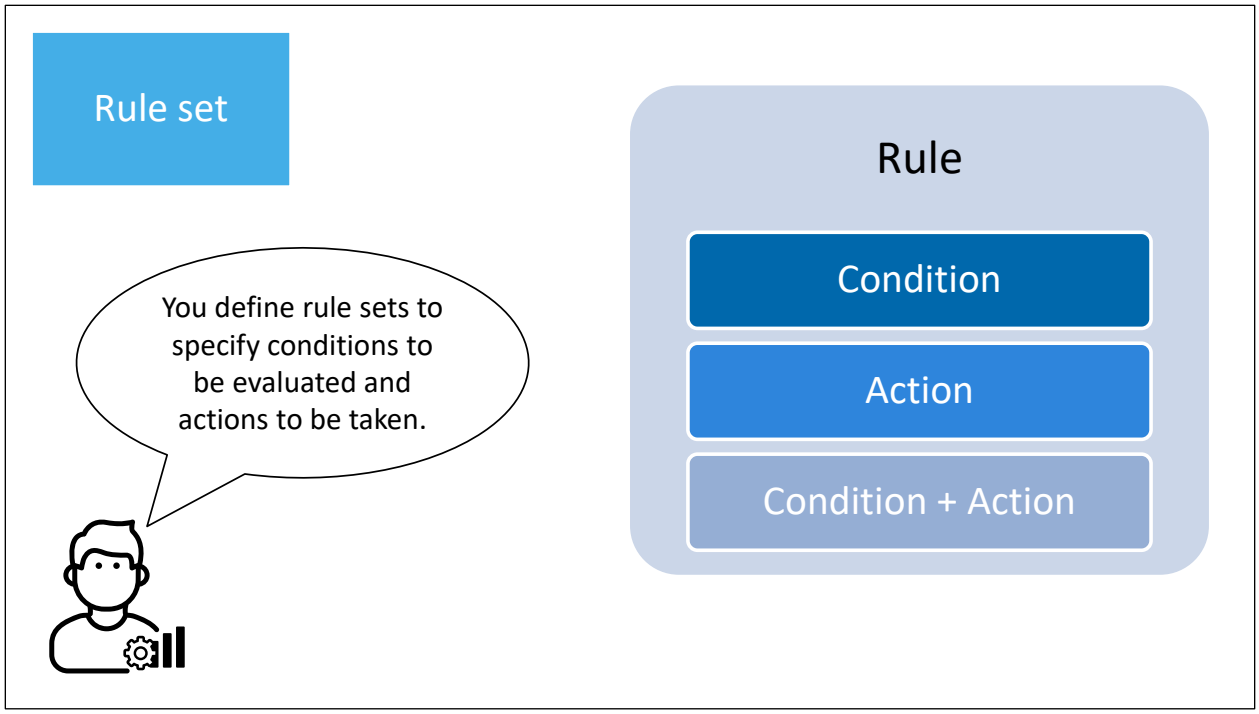


You can use branch nodes to create separate paths in the decision flow. Different processing can be performed for each path.

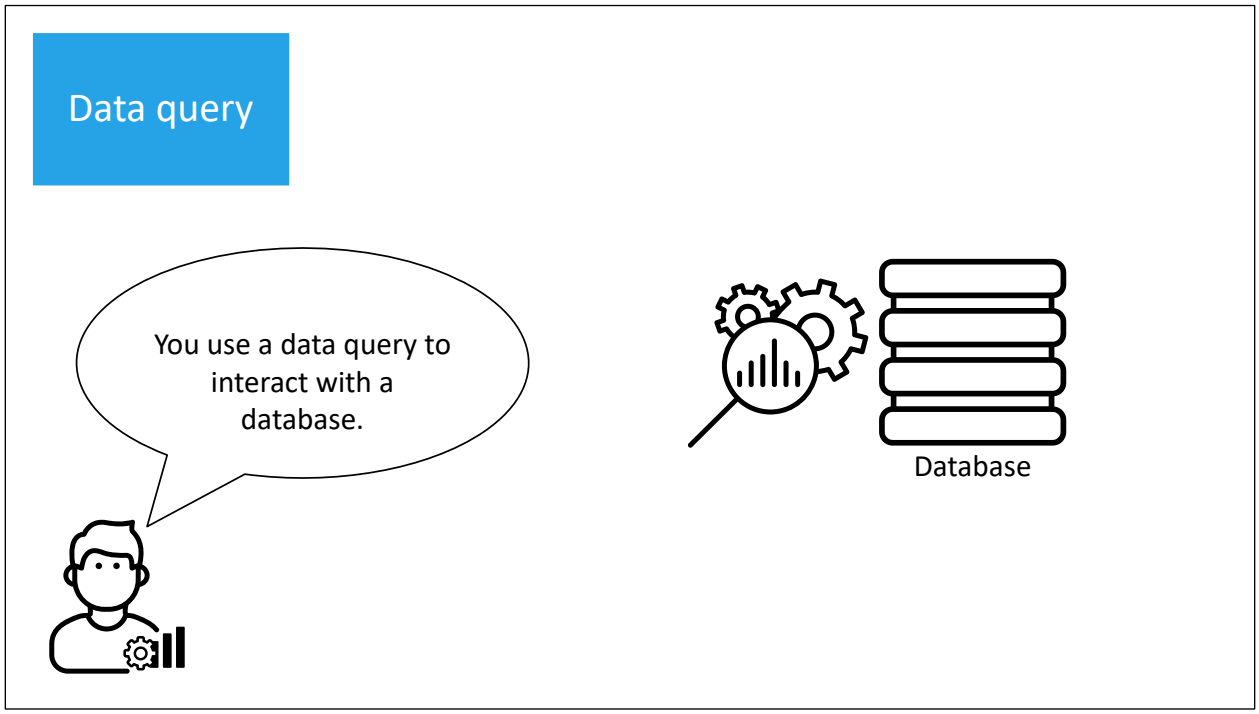


You use cross-branch links to easily reuse components across branches.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

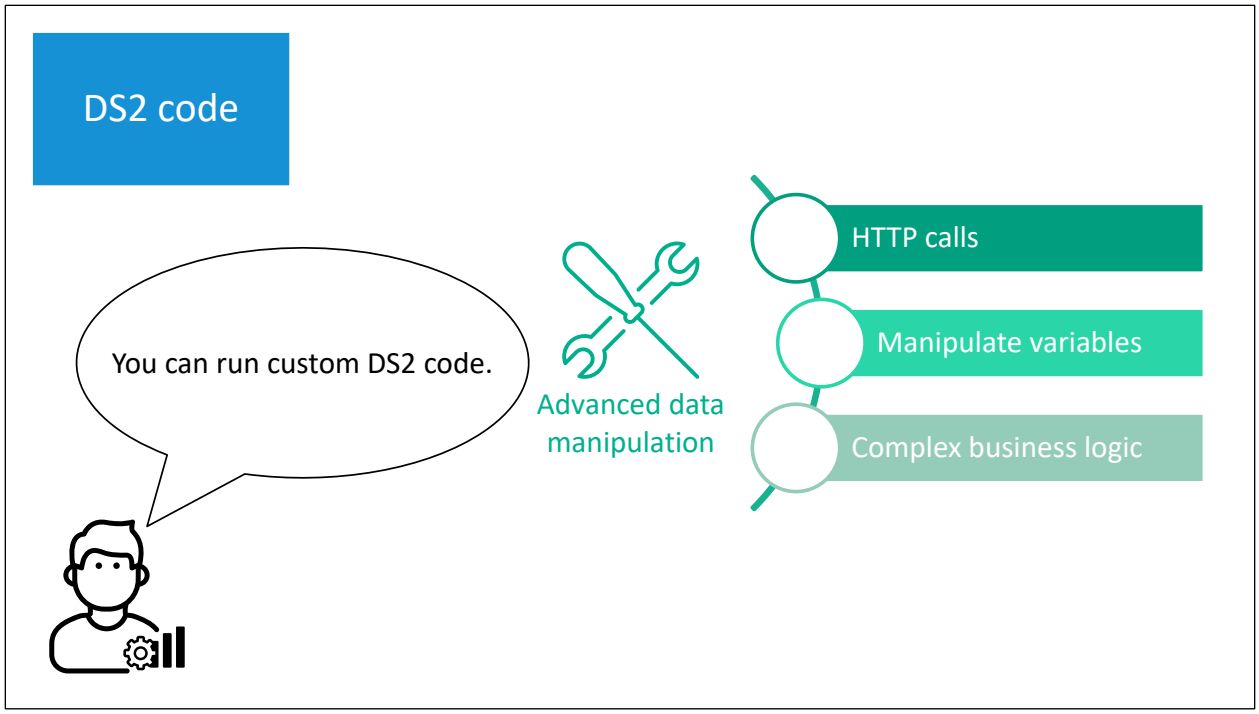


You can use rule sets to specify conditions to be evaluated and actions to be taken.

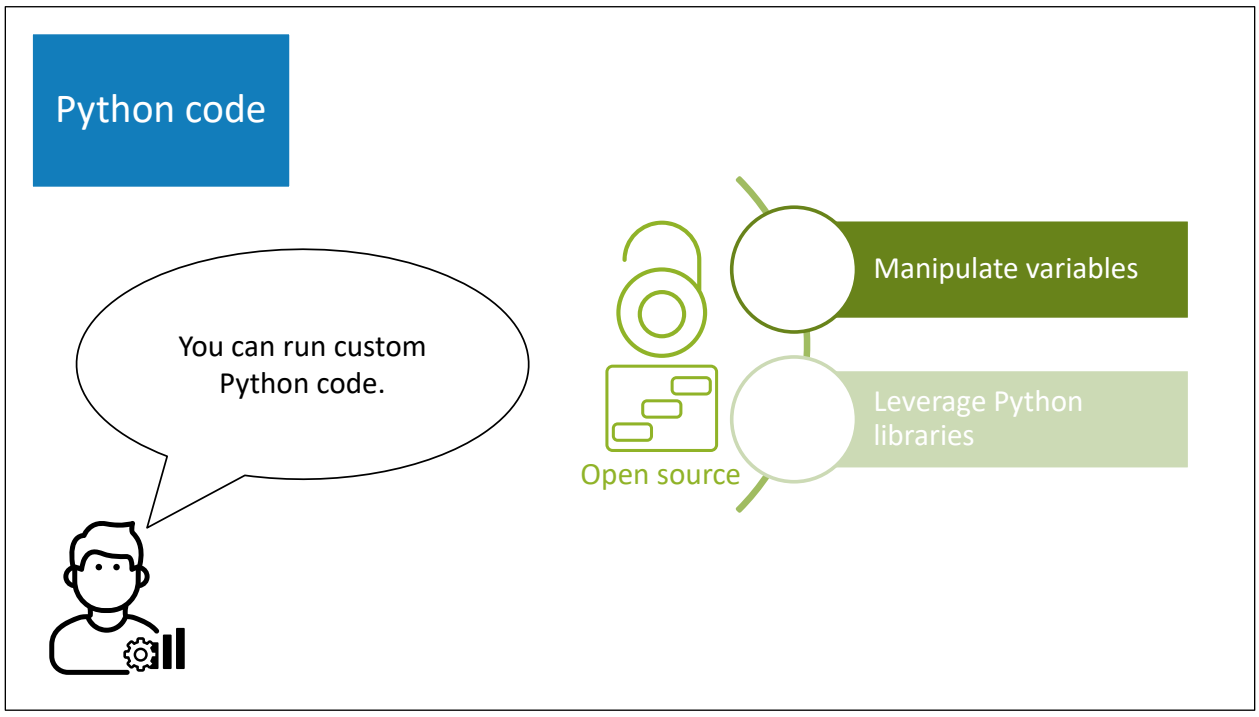


You can use a data query to interact with a database.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

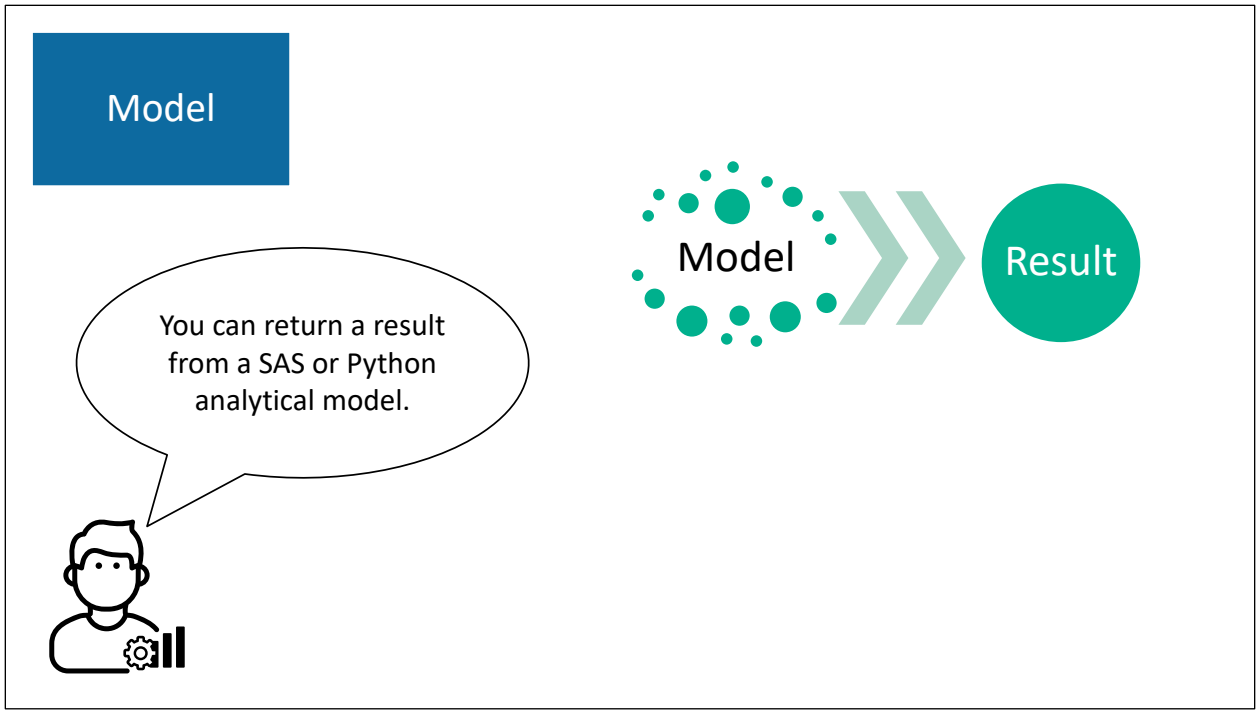


You can run custom DS2 code. DS2 is a SAS proprietary programming language that is appropriate for advanced data manipulation.

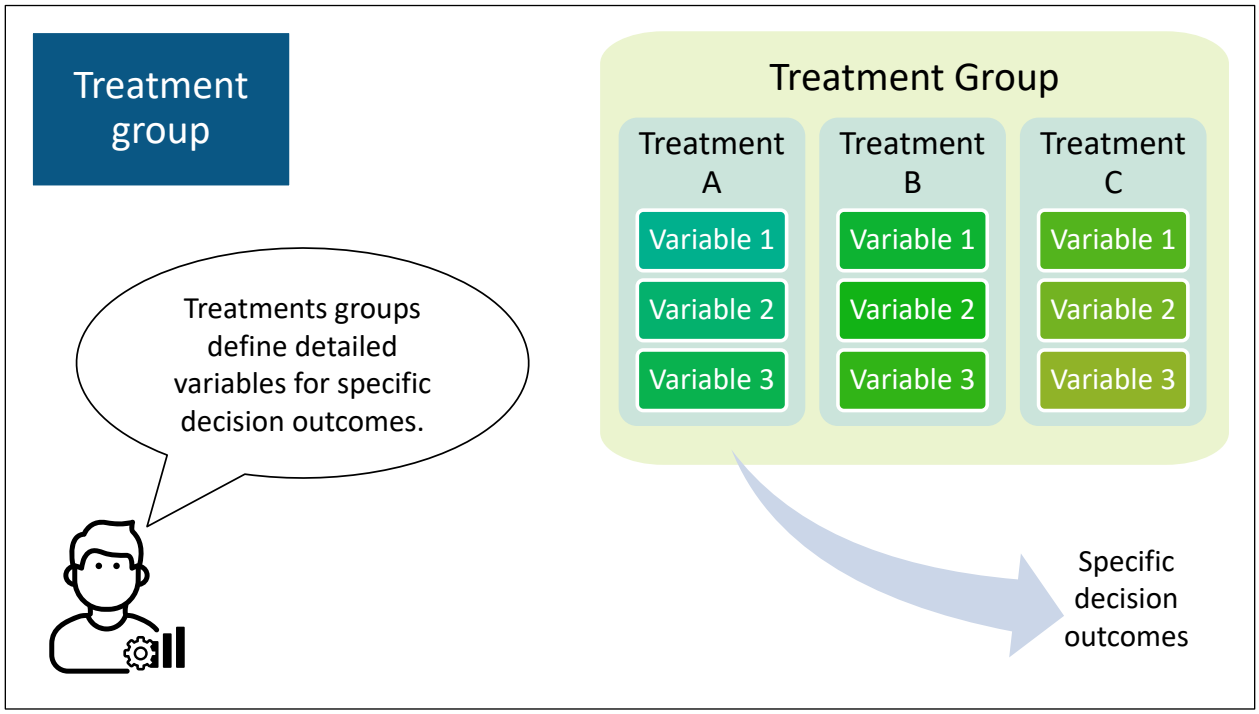


You can run custom Python code in a decision. Python is a popular open source language.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

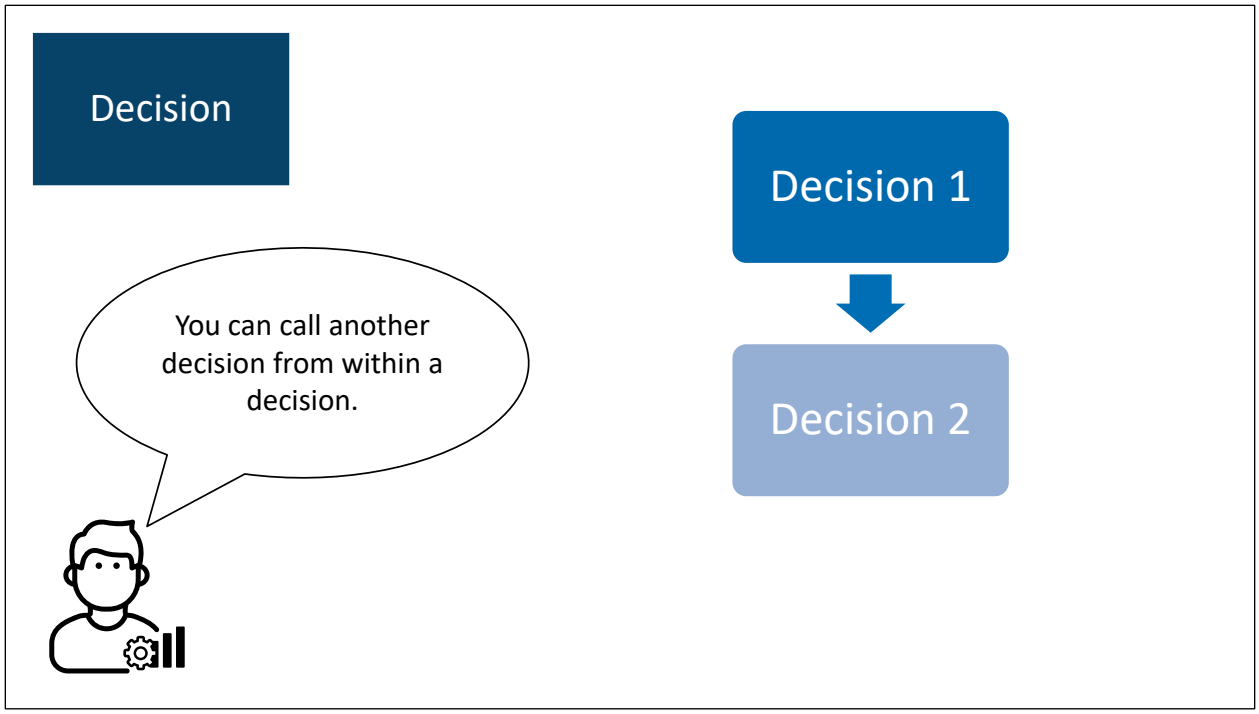


You can return a result from an analytical model. The model can be a SAS model or a Python model.

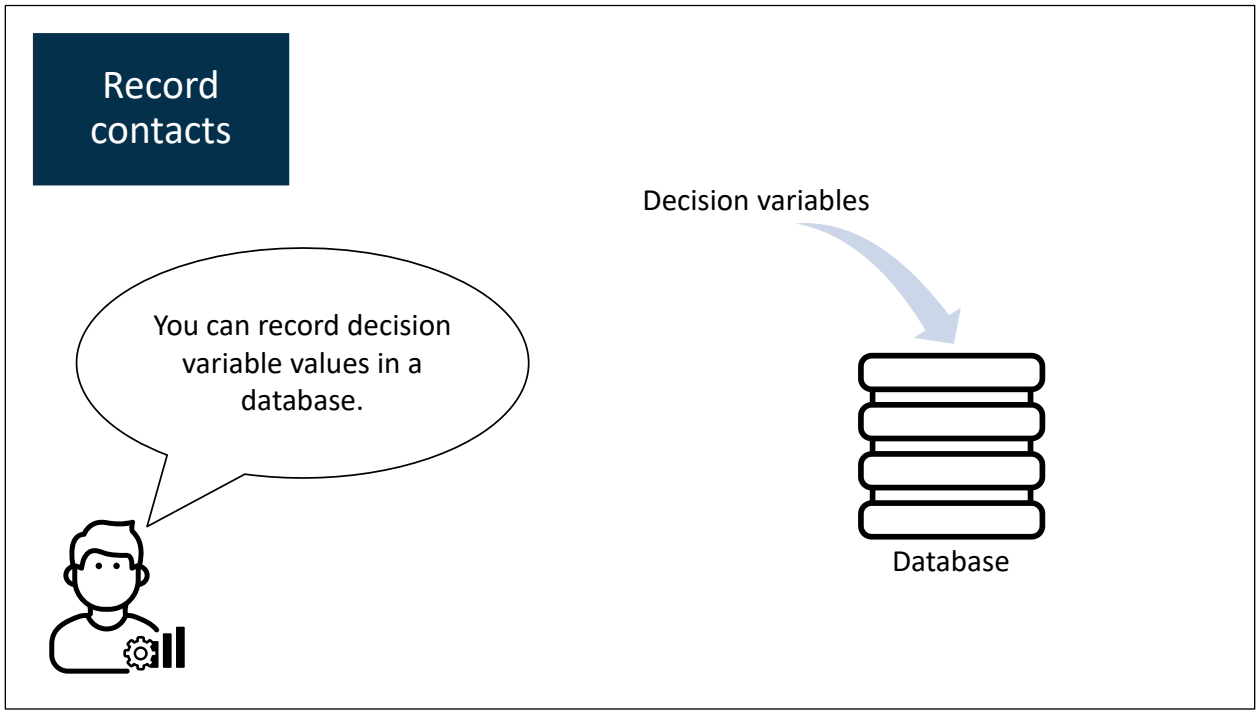


You can use treatments and treatment groups that define detailed variables corresponding to specific decision outcomes. These can be reused across multiple decision flows, promoting consistency and efficiency.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

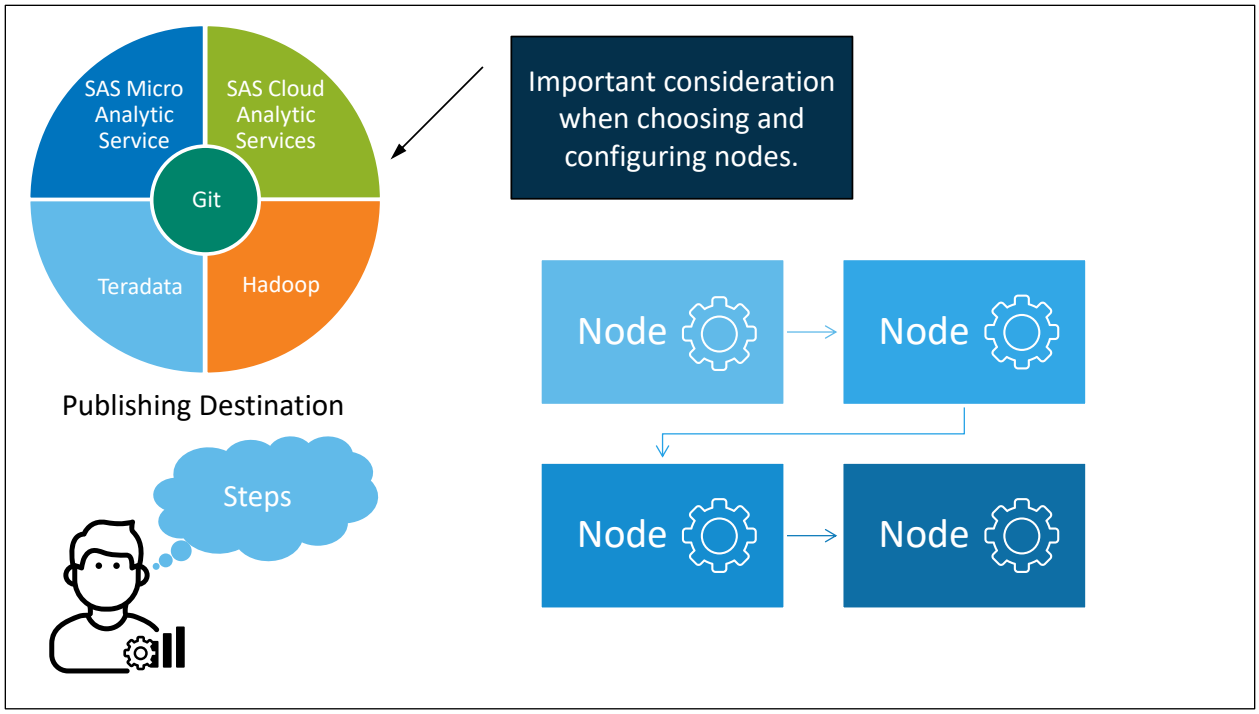


You can call another decision from within a decision.

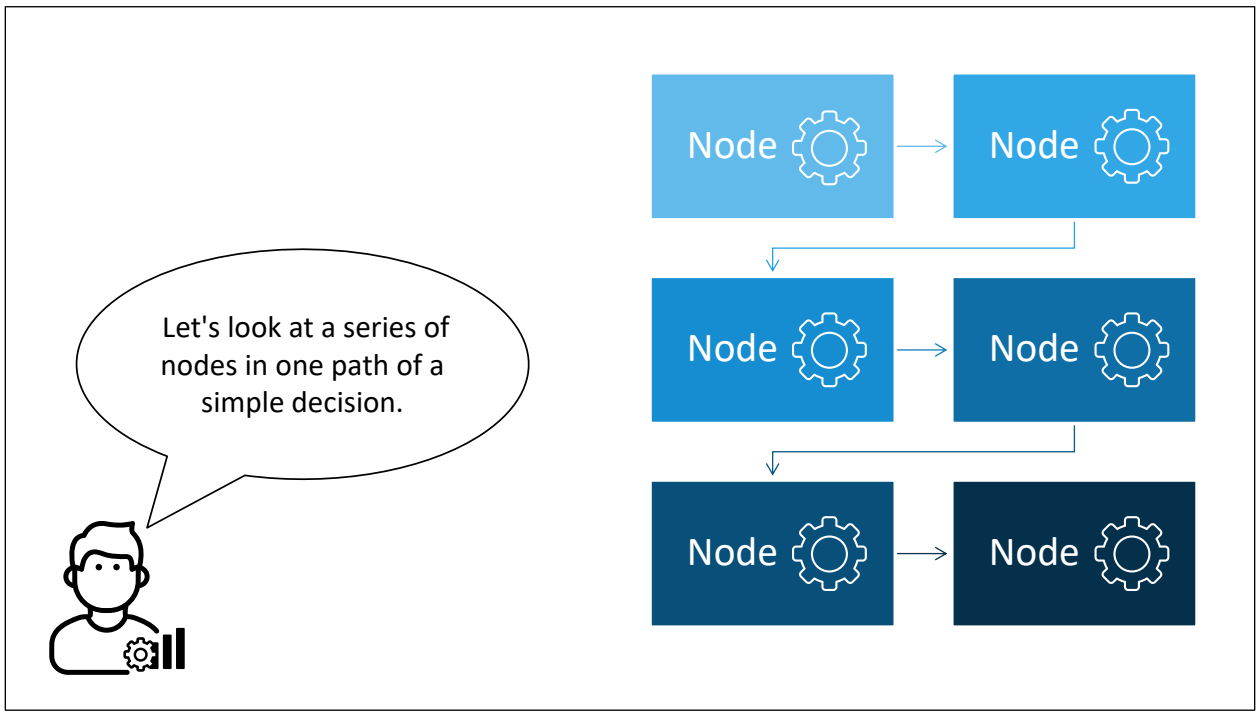


You can use a record contacts node to record decision variable values in a database.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

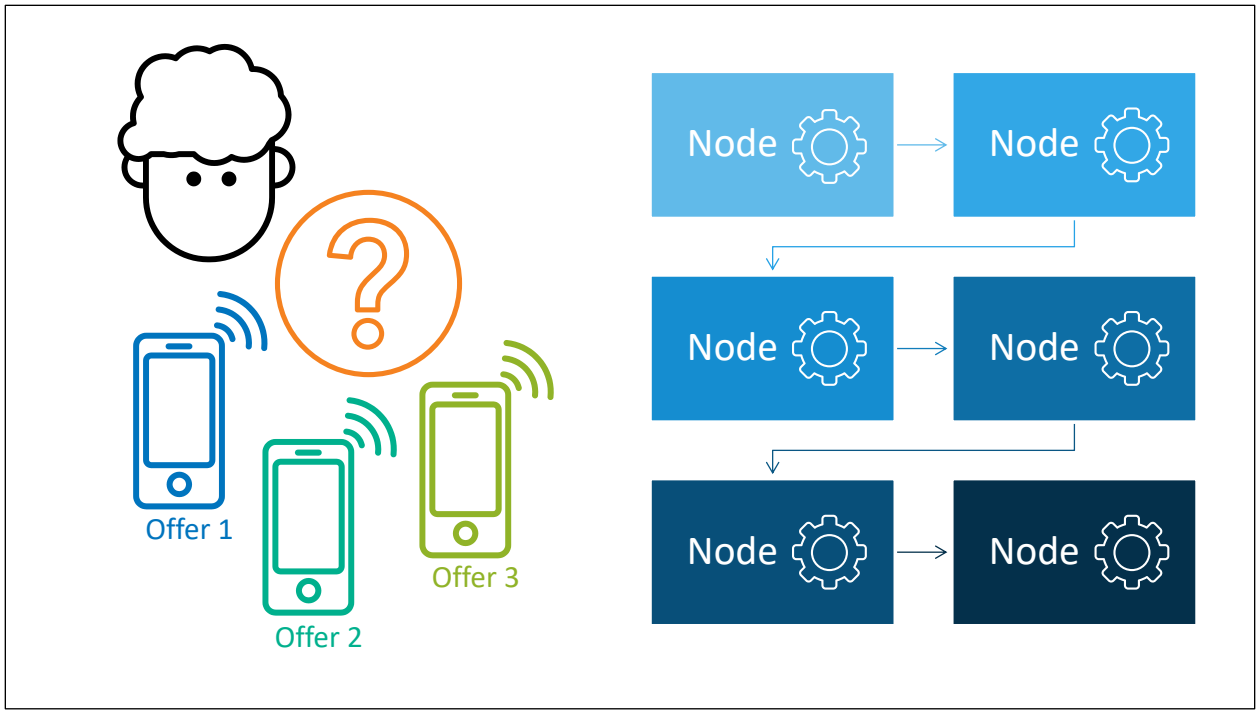


Your intended publishing destination is an important consideration when choosing and configuring nodes in your decision. Details for specific nodes are discussed in later lessons.

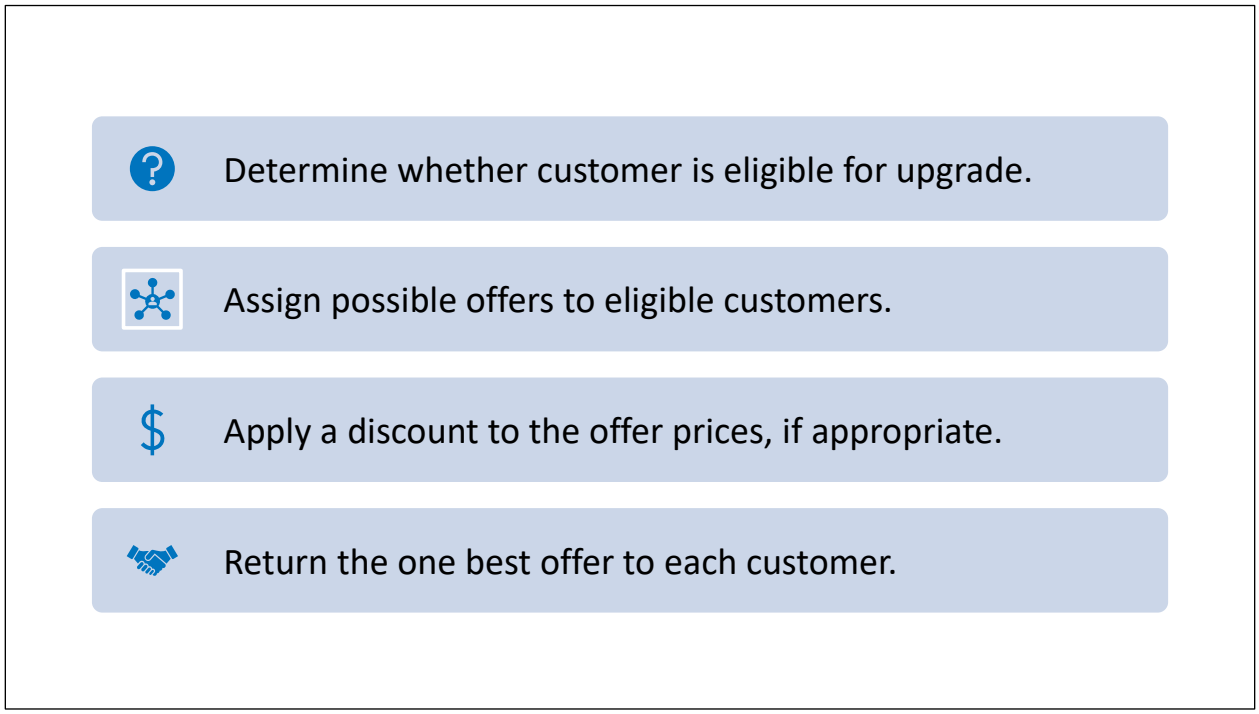


Let's take a look at an example of one path in a simple decision and the specific nodes that it uses.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

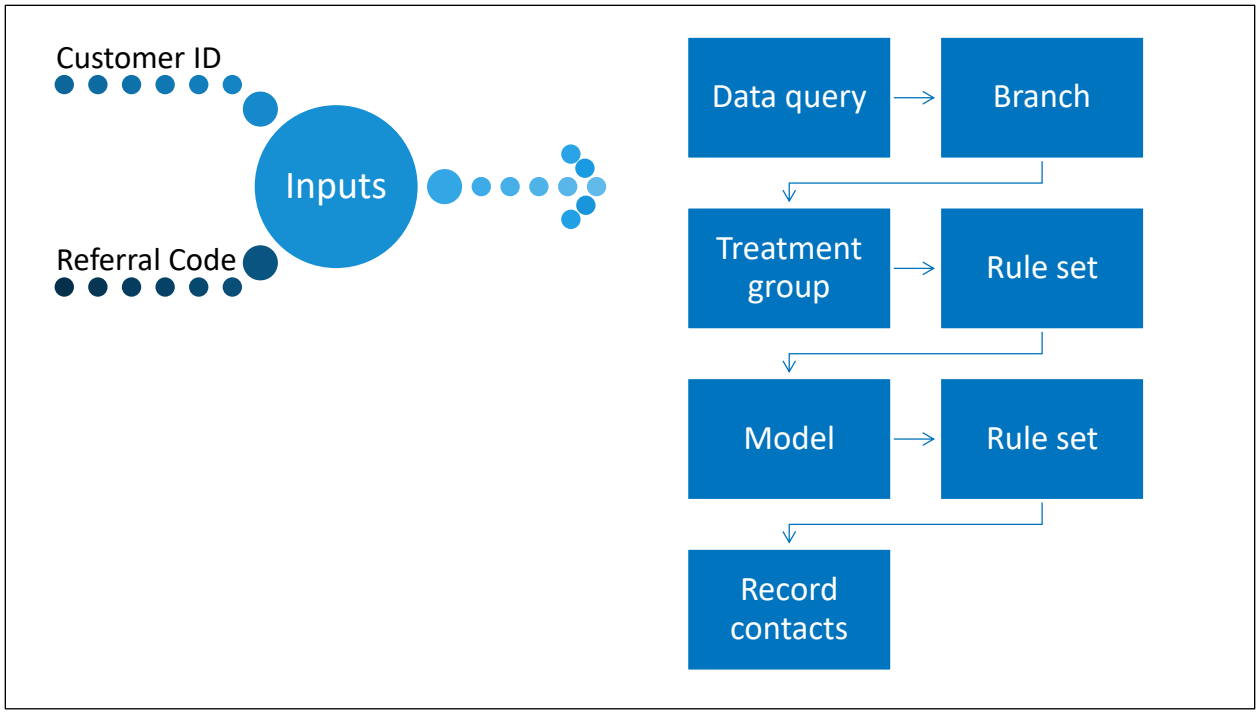


The purpose of the decision is to determine what type of cell phone package to offer to a customer.

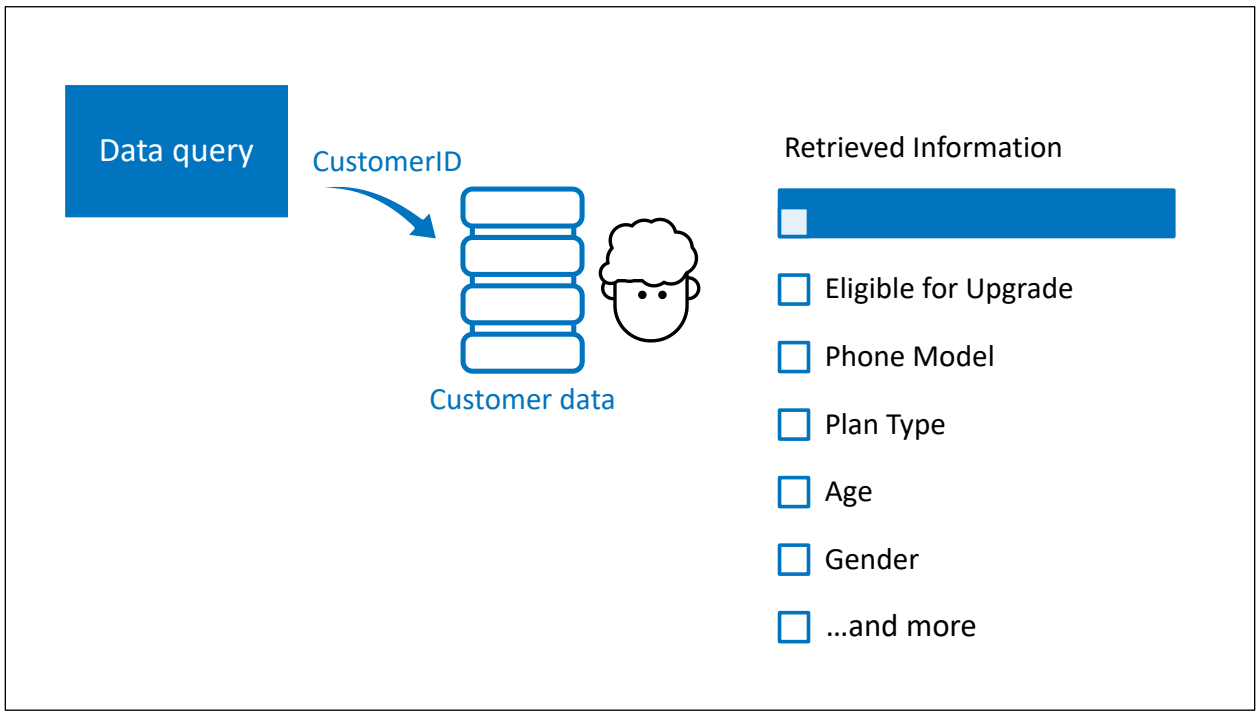


At a high level, the decision will determine whether the customer is eligible for a device upgrade. If so, the decision assigns possible offers to the customer and applies a discount to the price of the phone, if appropriate. Finally, it returns the one offer that is determined to be the best fit to the customer.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

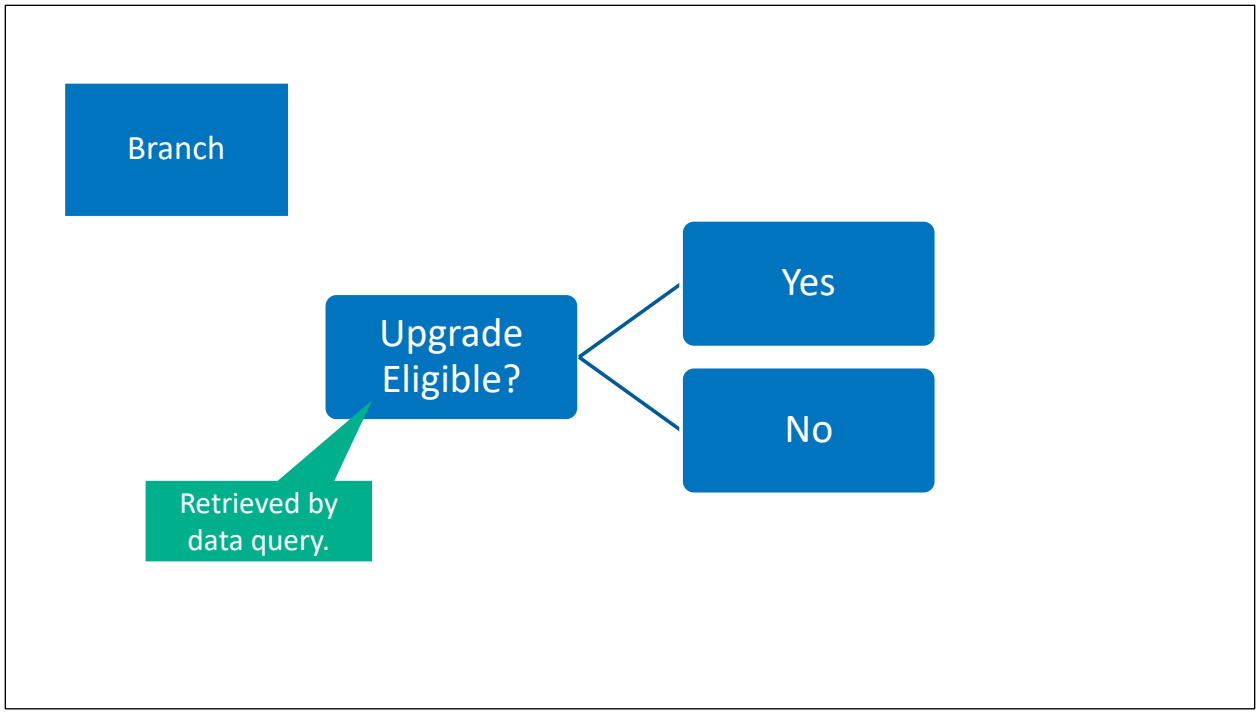


The inputs to the decision are the customer's ID and a referral code. The decision contains a series of nodes to perform the needed processing.

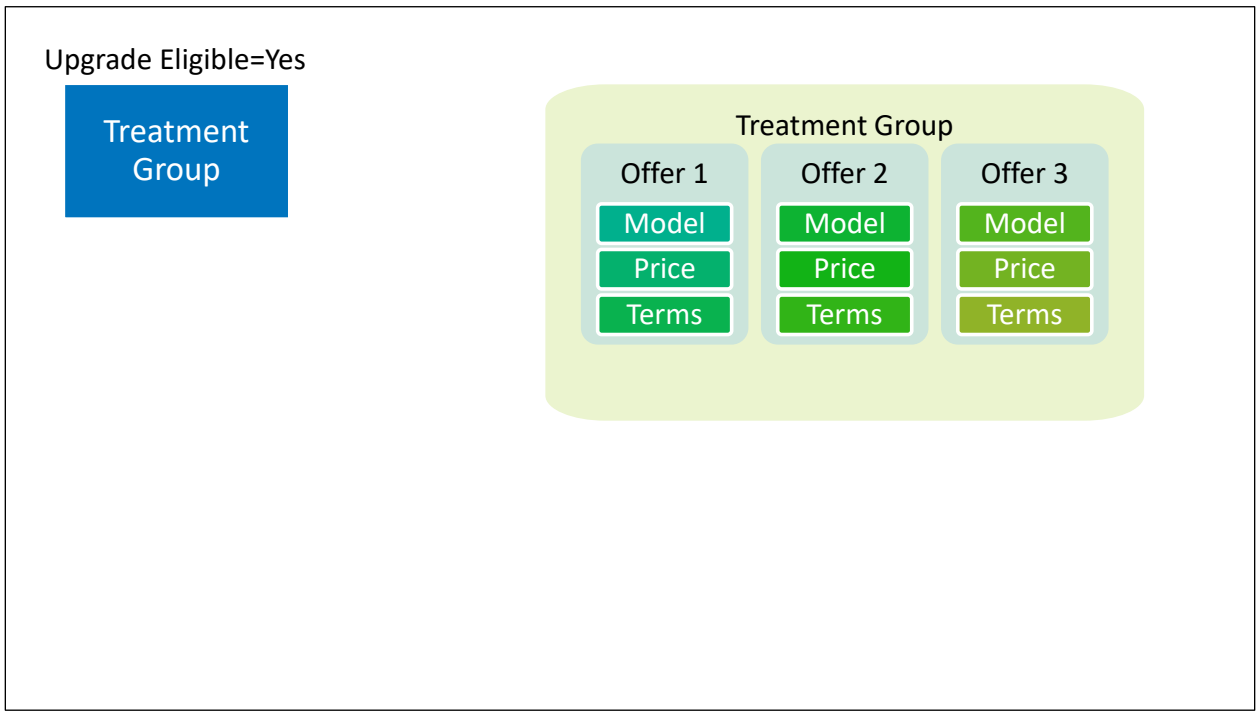


The first node in the decision flow is a data query. It uses the customer ID that was input to the decision to retrieve information about the customer and their current phone plan to use in the decision.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

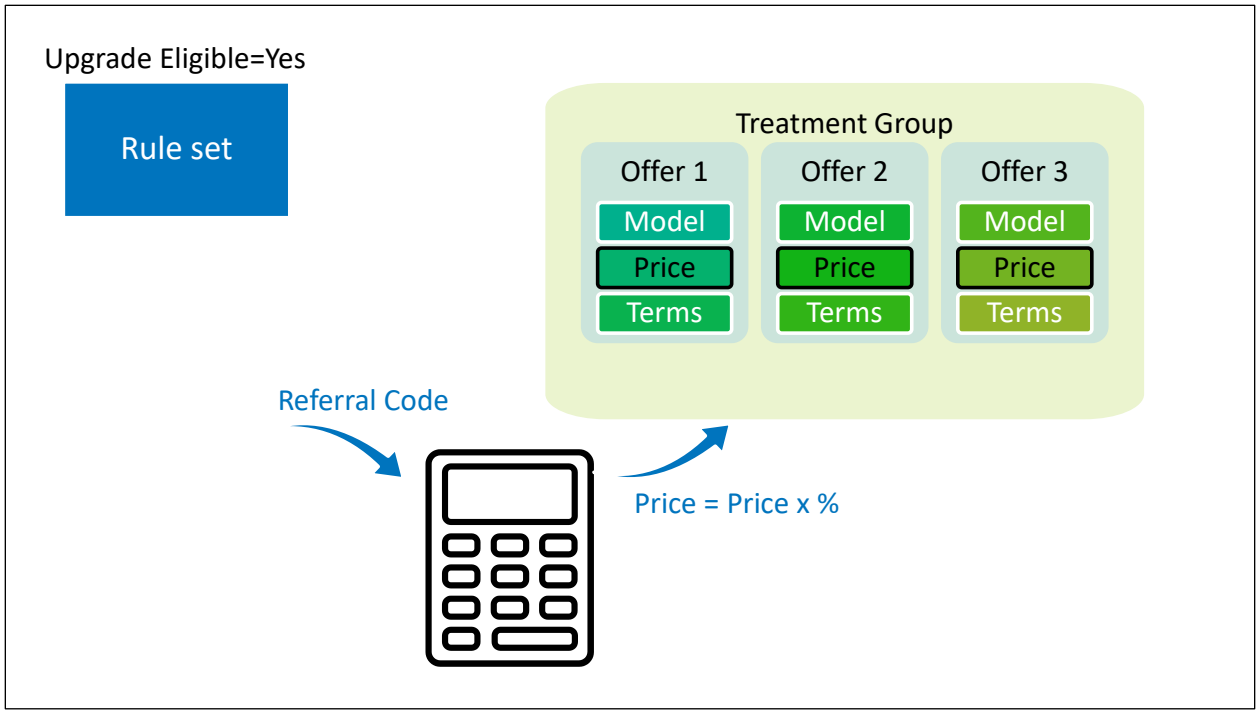


The next node in the decision is a branch node. Branching is based on whether the customer is eligible for a device upgrade. The eligibility information was retrieved in the data query.

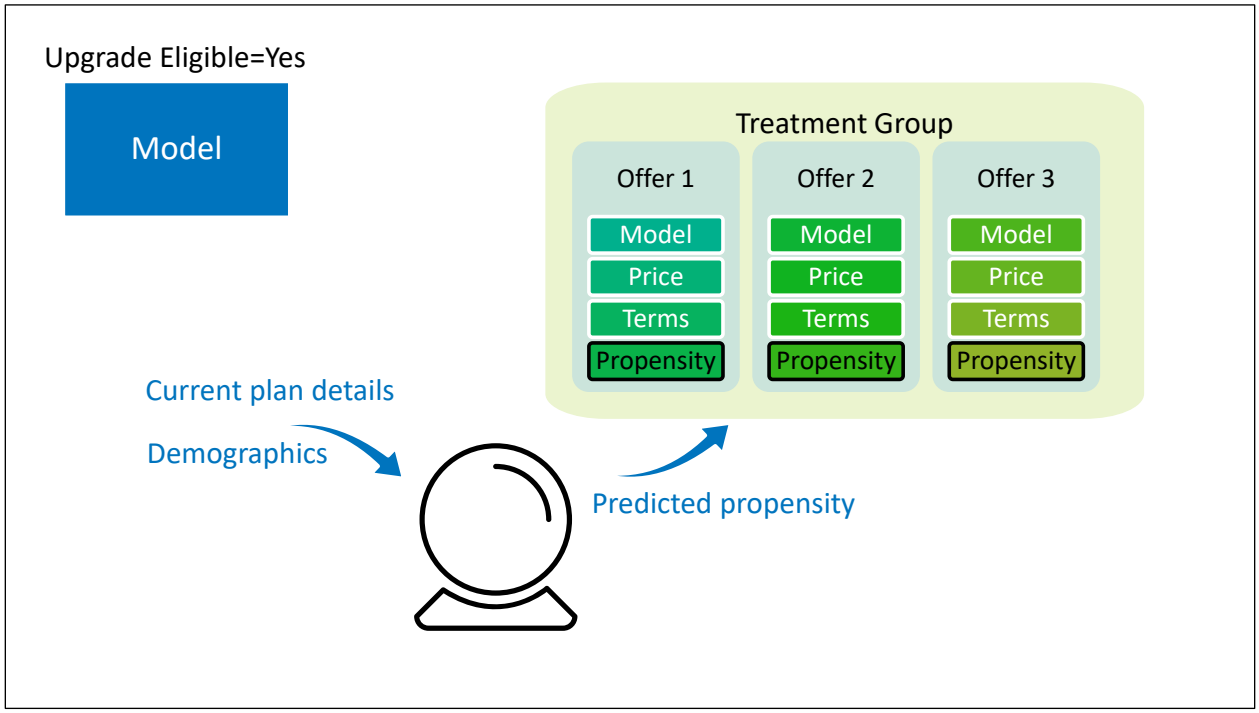


On the Yes path of the branch node, the next node is a treatment group. The treatment group is made up of treatments that describe the specifics of potential cell phone offers to make to the customer including the model, price, and terms.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

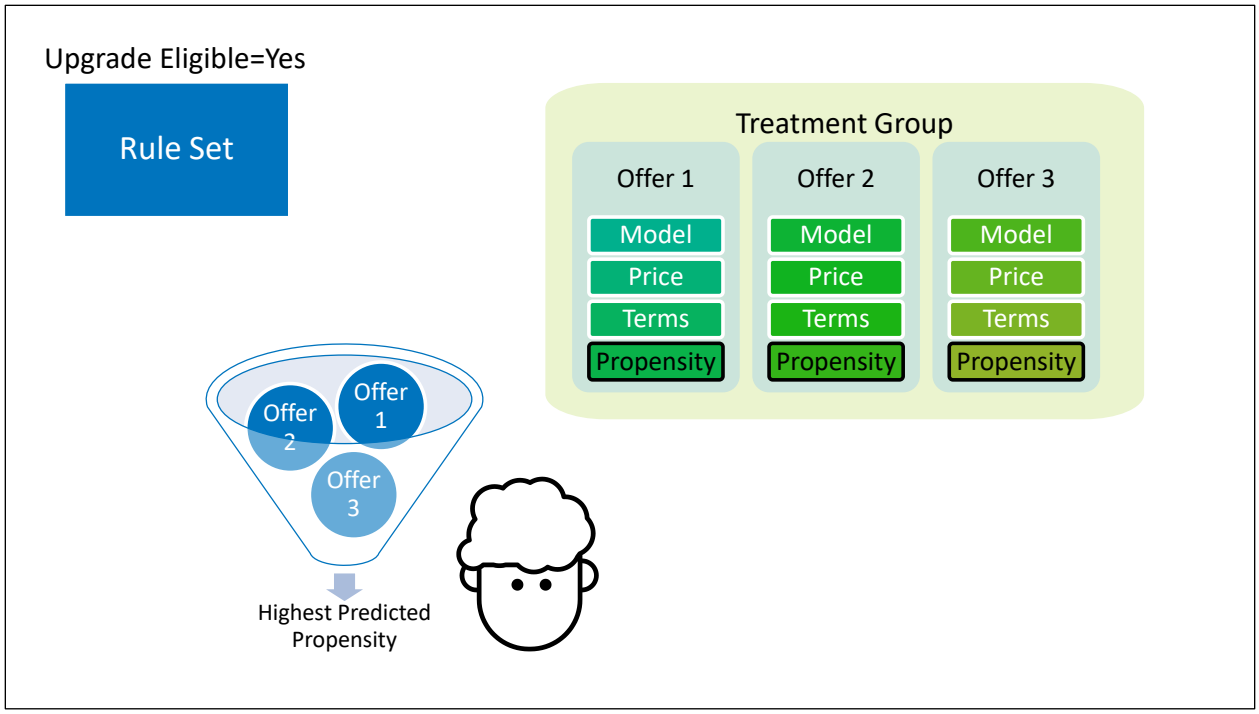


The next node is a rule set. The referral code that was an input to the decision determines the discount that the customer gets on any phone purchase. The rule set determines the discounted price for each offer based on the referral code.

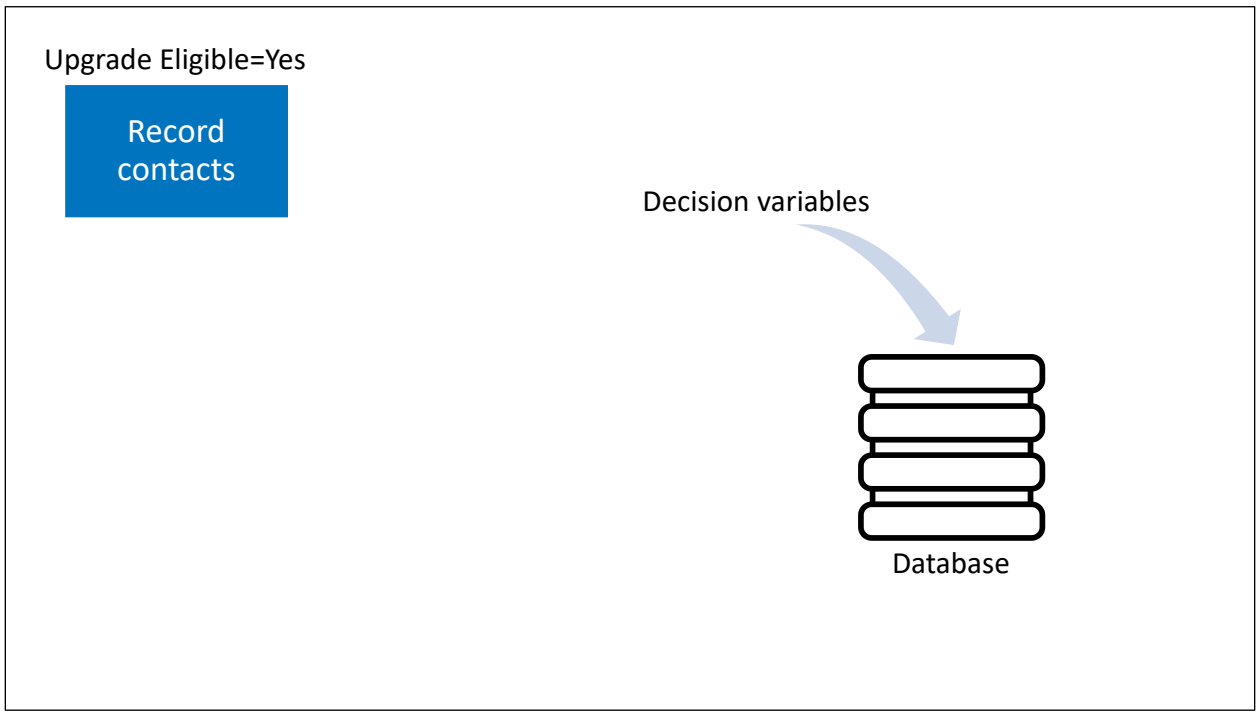


The next node is a model. The model calculates the predicted propensity of acceptance of each offer for the customer using the demographic and phone plan information that was retrieved by the data query.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The next node is another rule set. It determines which offer has the highest predicted propensity of acceptance for each customer, so that only this best offer is returned by the decision.



The final node is a record contacts node. It records the details of the treatment group and other selected variables in a database.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Working in SAS Intelligent Decisioning

This demonstration explores the SAS Intelligent Decisioning user interface.

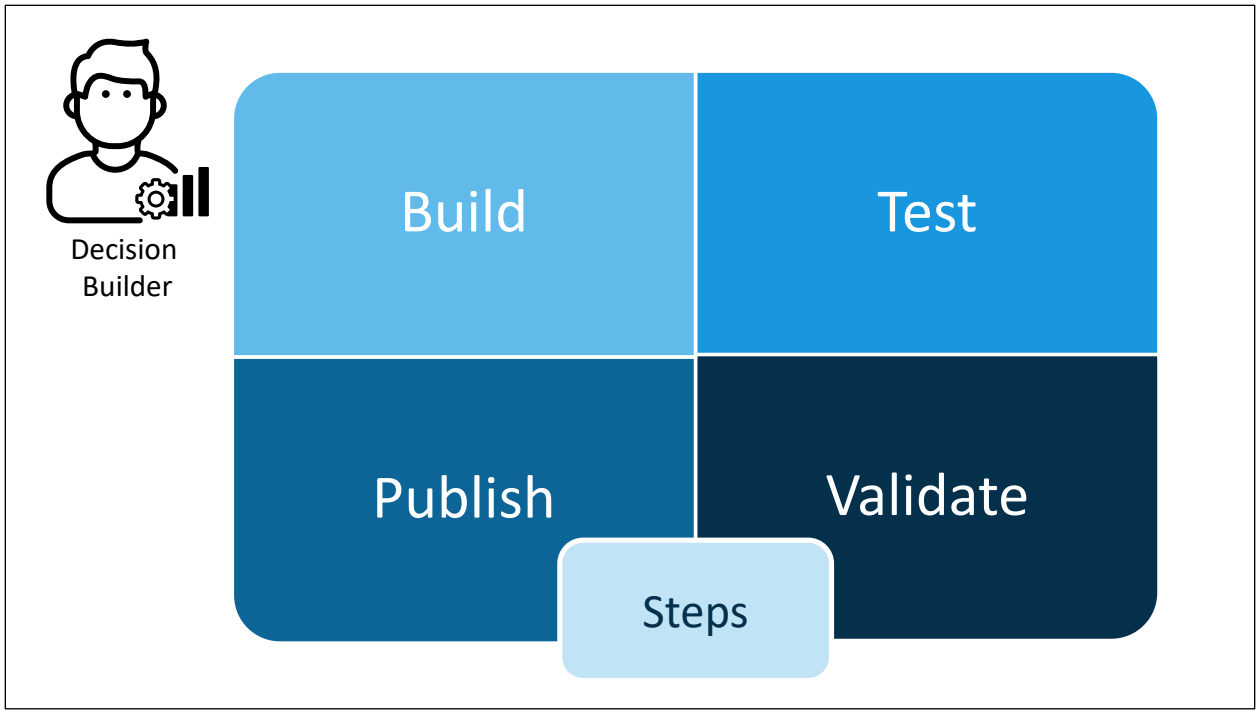
Copyright © SAS Institute Inc. All rights reserved.

Practice

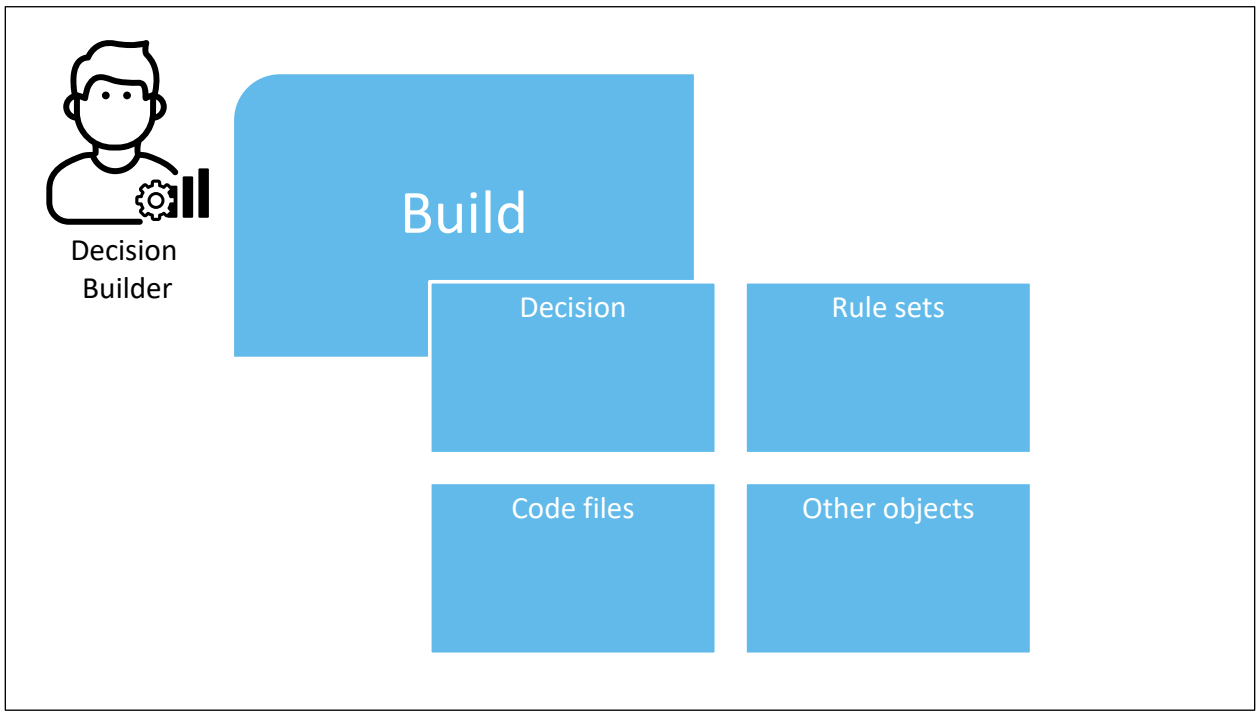
This practice walks through logging on to SAS Intelligent Decisioning and explores the user interface.

Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

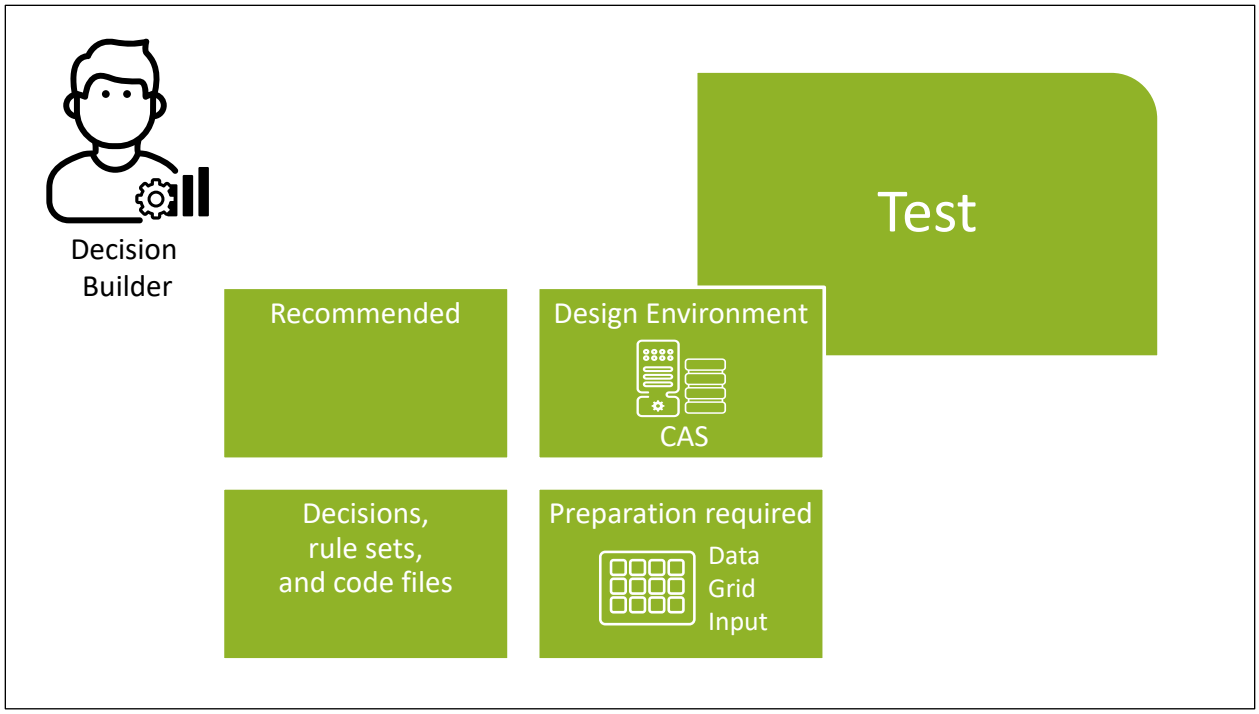


There are four main categories of activities when you develop a decision: Build, Test, Publish, and Validate.

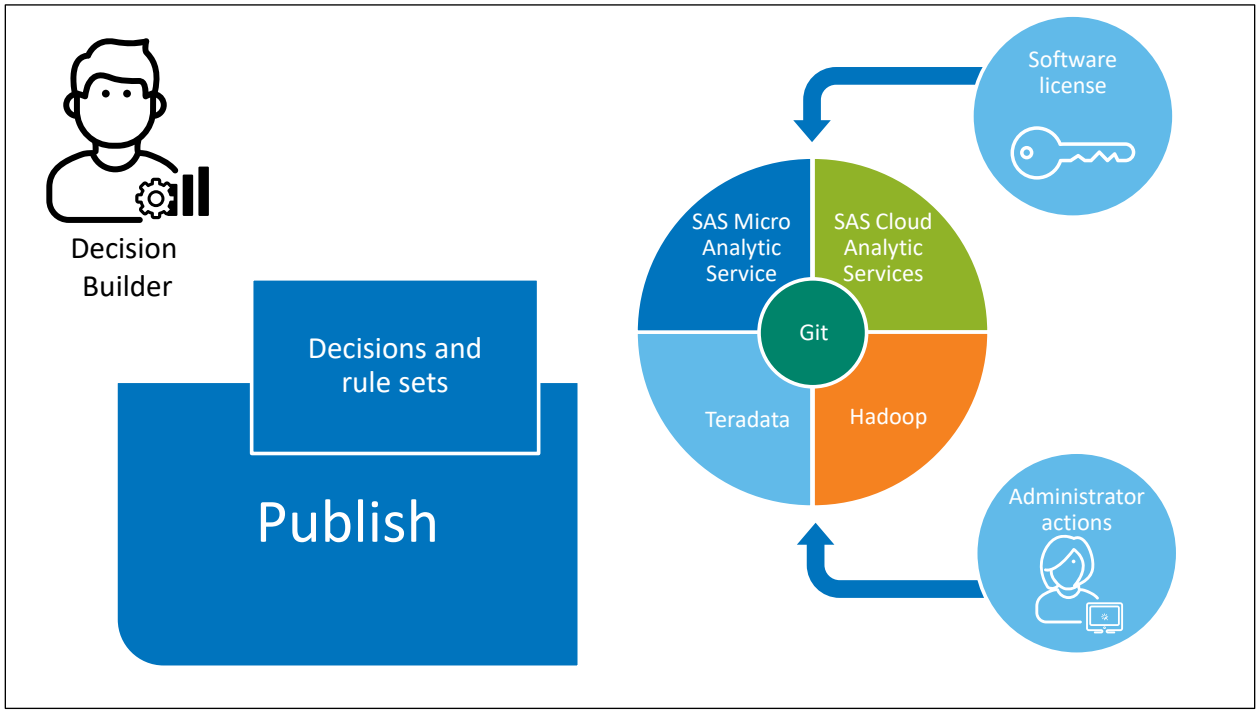


You begin by building the decision itself along with any objects that it uses, such as rule sets or code files.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

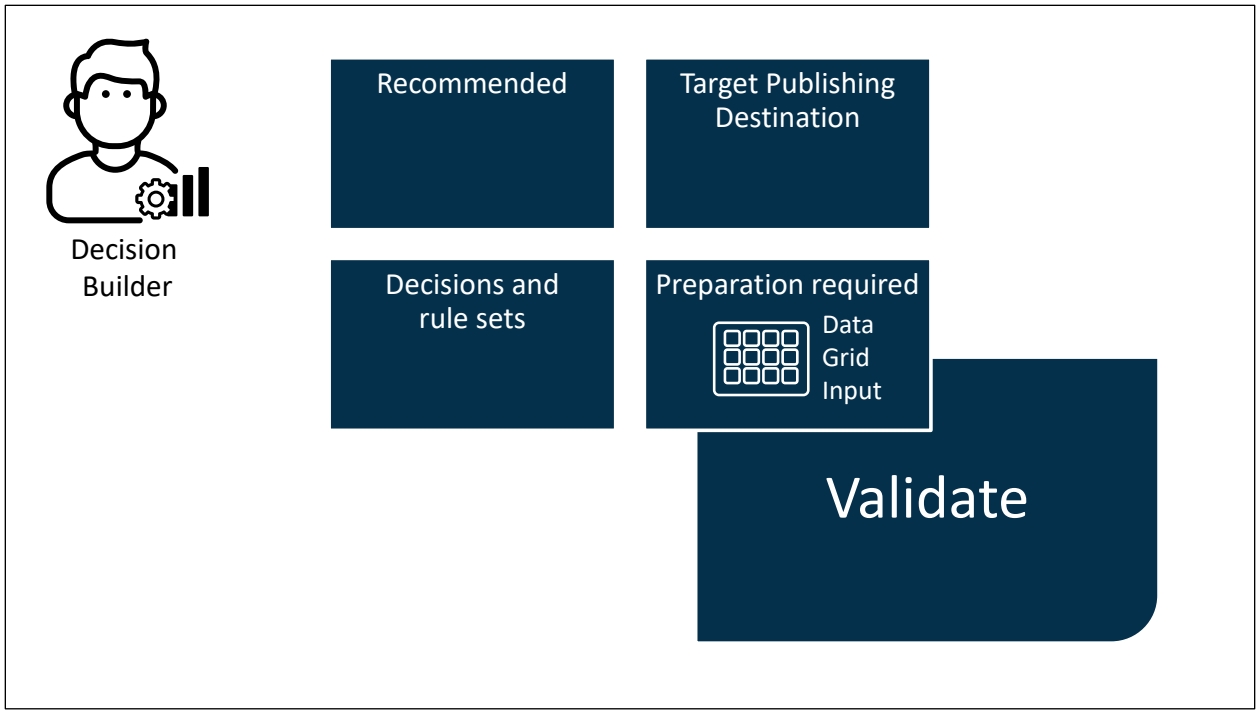


Testing is recommended, but not required. Testing is performed in the design environment using CAS. You can test decisions, rule sets, and code files in SAS Intelligent Decisioning. If your decision or rule set includes input data grid variables, you must perform steps to prepare the data for testing.

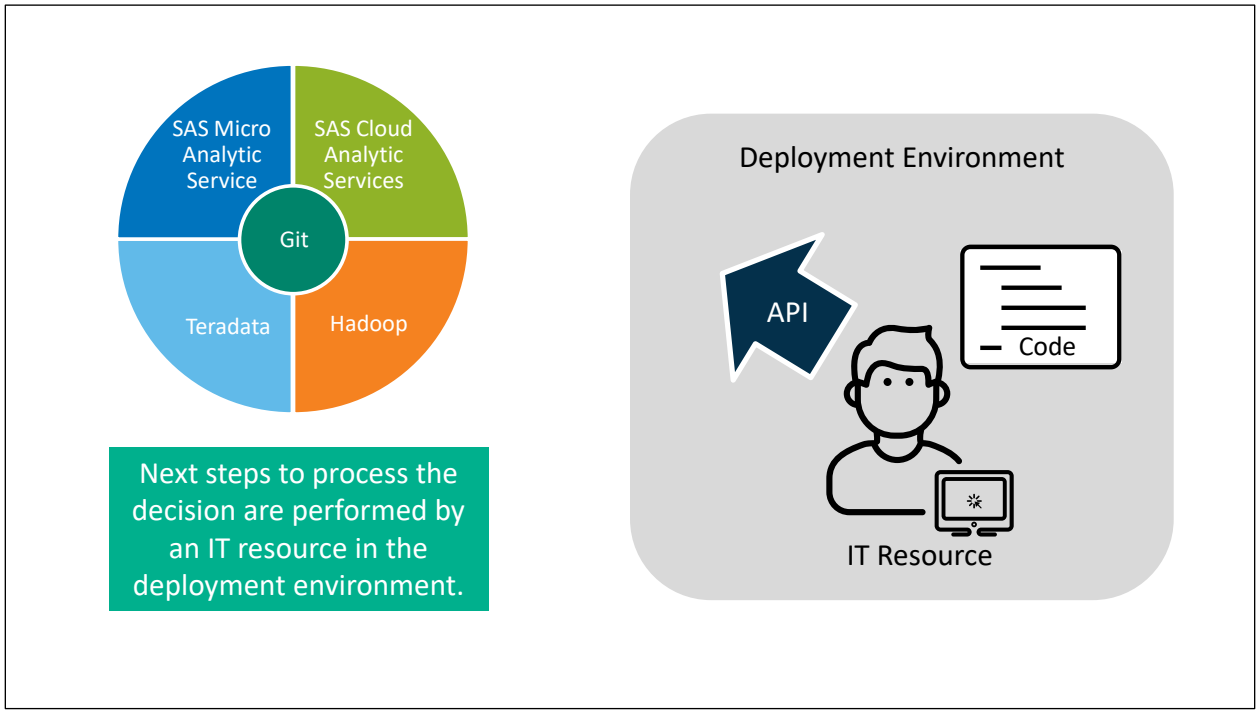


When you are ready to produce results, you publish the decision or rule set. Recall that the publishing destinations available to you depend on your software license and the actions taken by your administrator.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

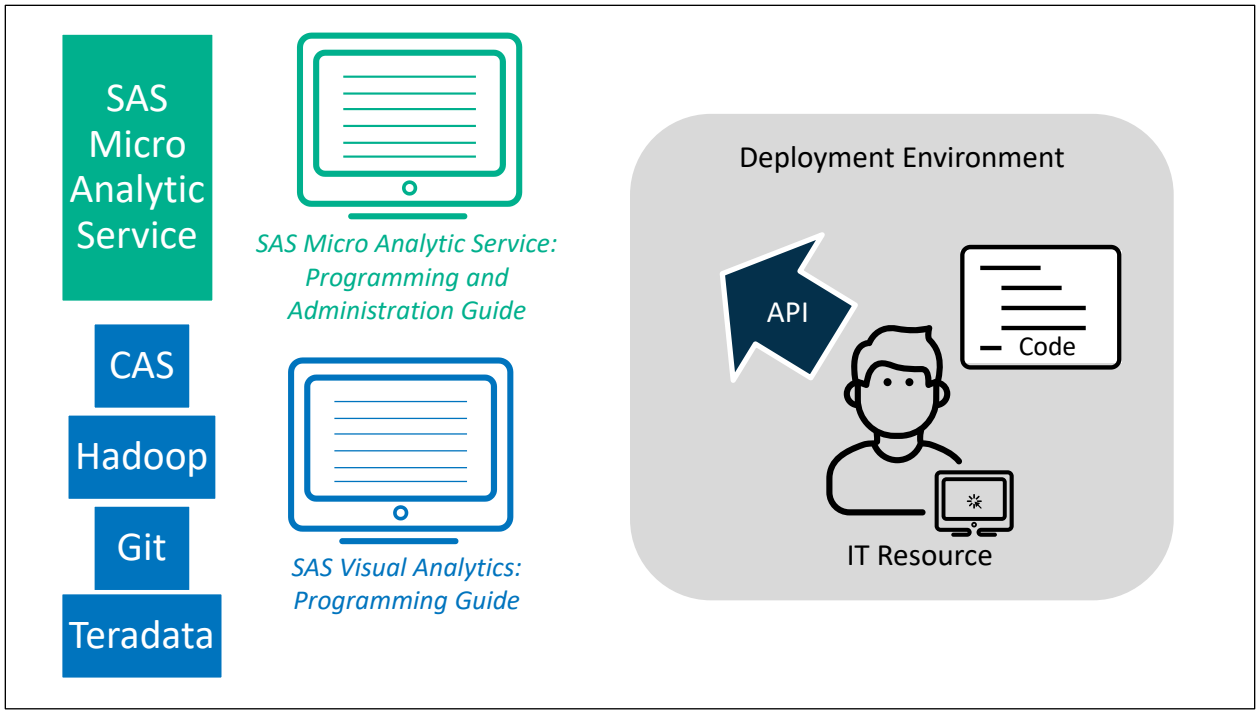


After publication, you can validate decisions and rule sets. This step is recommended, but not required. Validation is performed in the target publishing destination. If your decision or rule set includes input data grid variables and you are publishing to CAS, Teradata, Git, or Hadoop you must perform steps to prepare the data for processing. Refer to the lesson about variables to learn more.



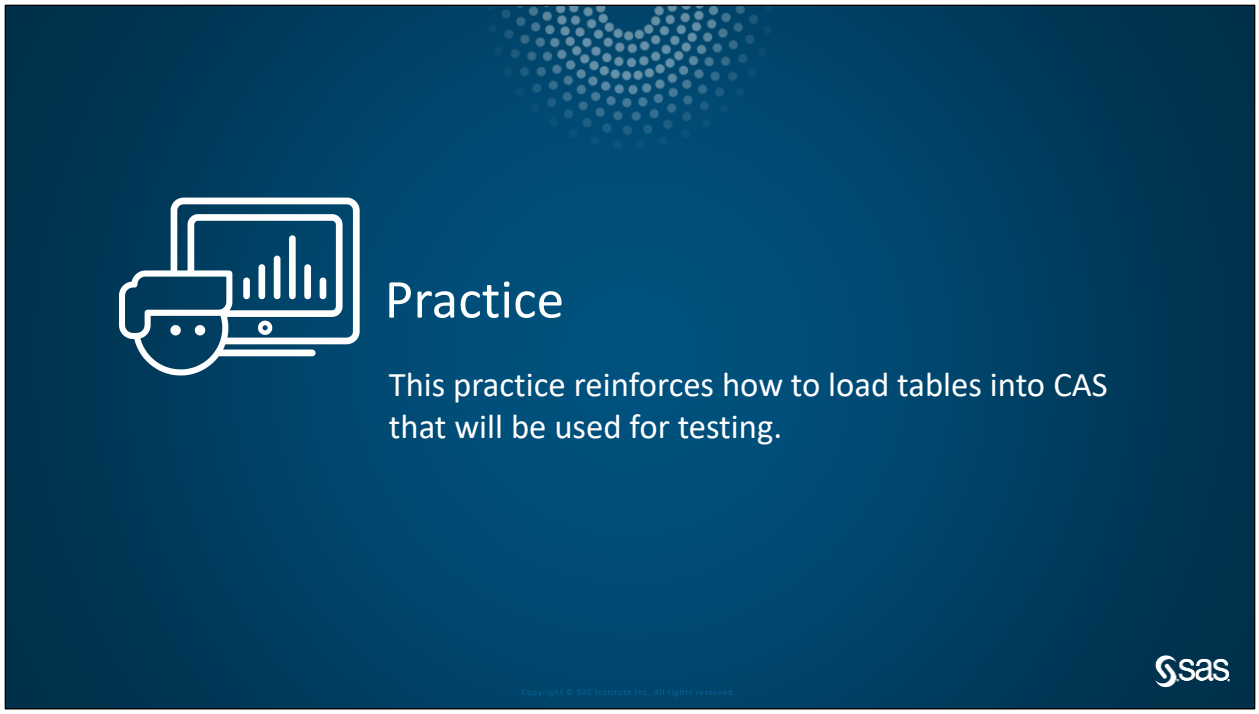
Recall that no matter which publishing destination you choose, the next steps needed to process the decision are performed by an IT resource in the deployment environment.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



If you want to learn more about the next steps for the Micro Analytic Service, consult *SAS Micro Analytic Service: Programming and Administration Guide*. To learn more about the next steps for CAS, Hadoop, Git or Teradata, refer to *SAS Visual Analytics: Programming Guide*.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

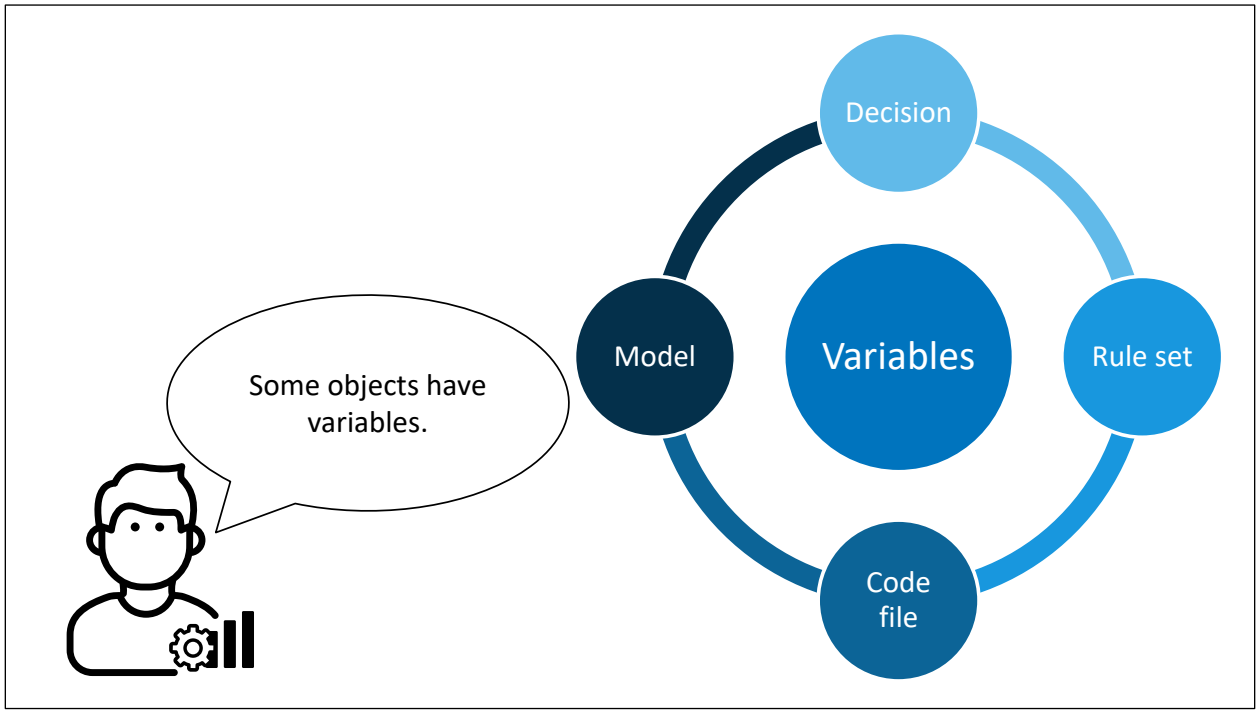


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

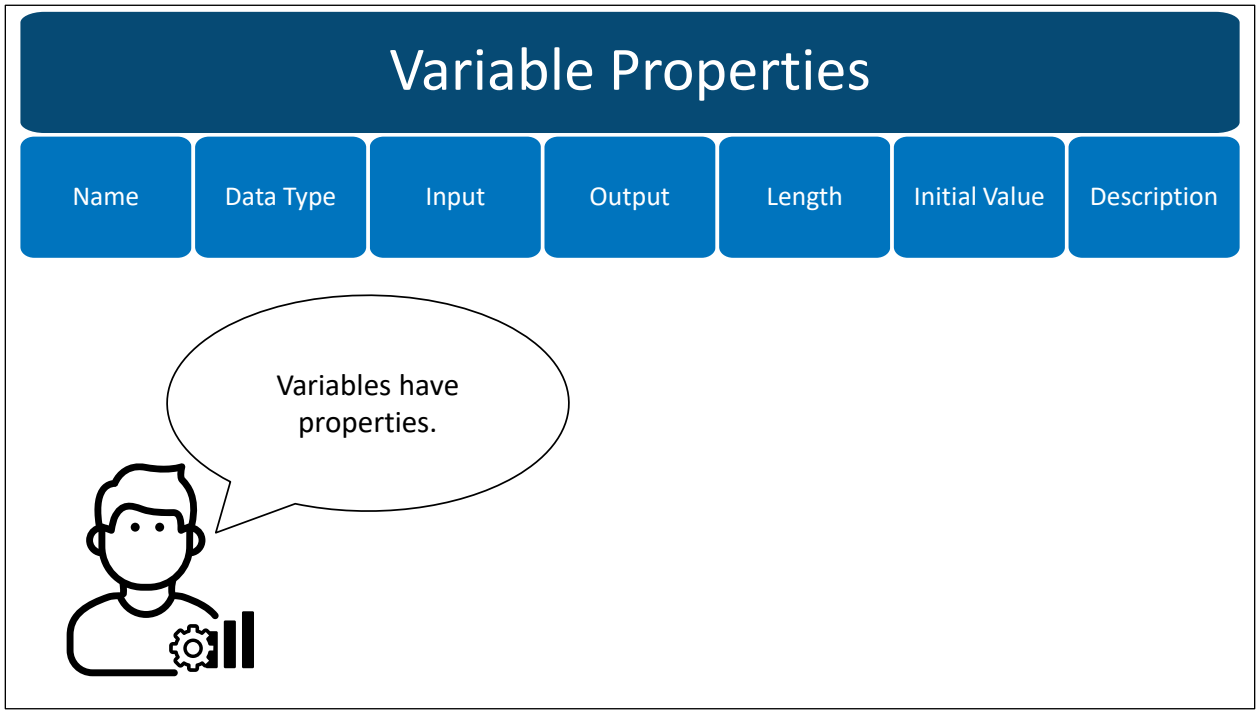
Lesson 2 Decision Basics

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.1 Variables

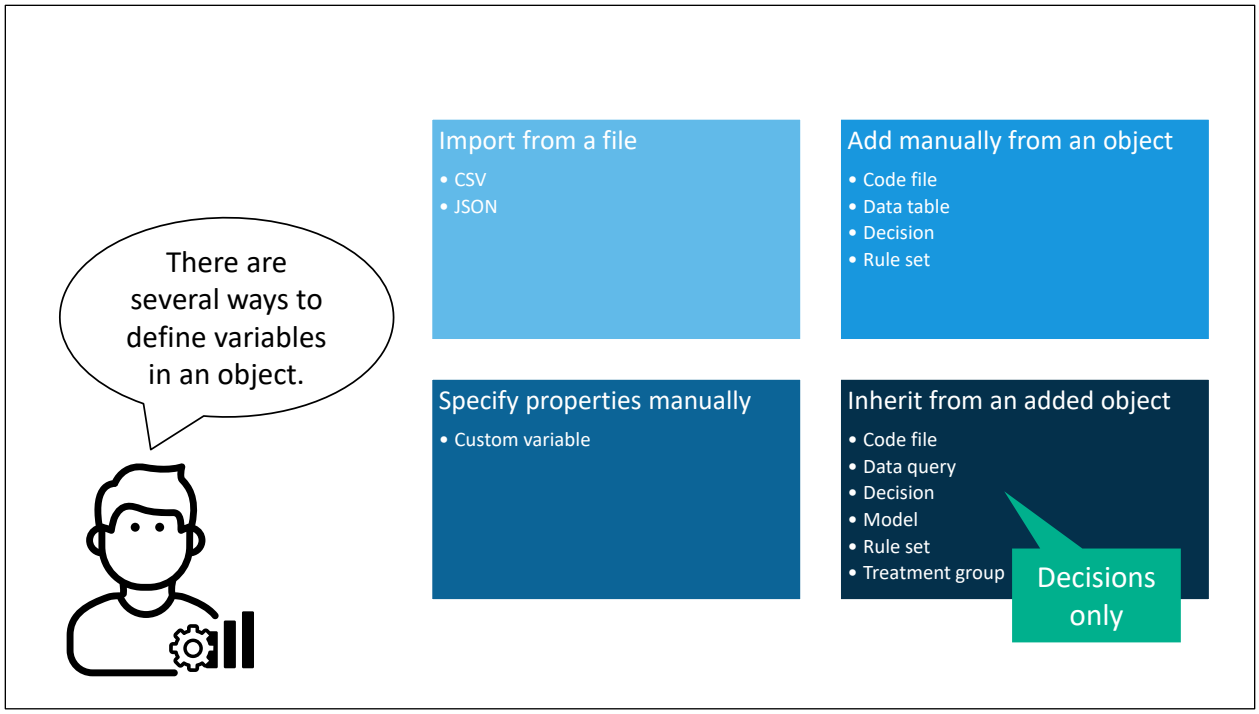


Decisions and some objects that are used in decisions have variables.

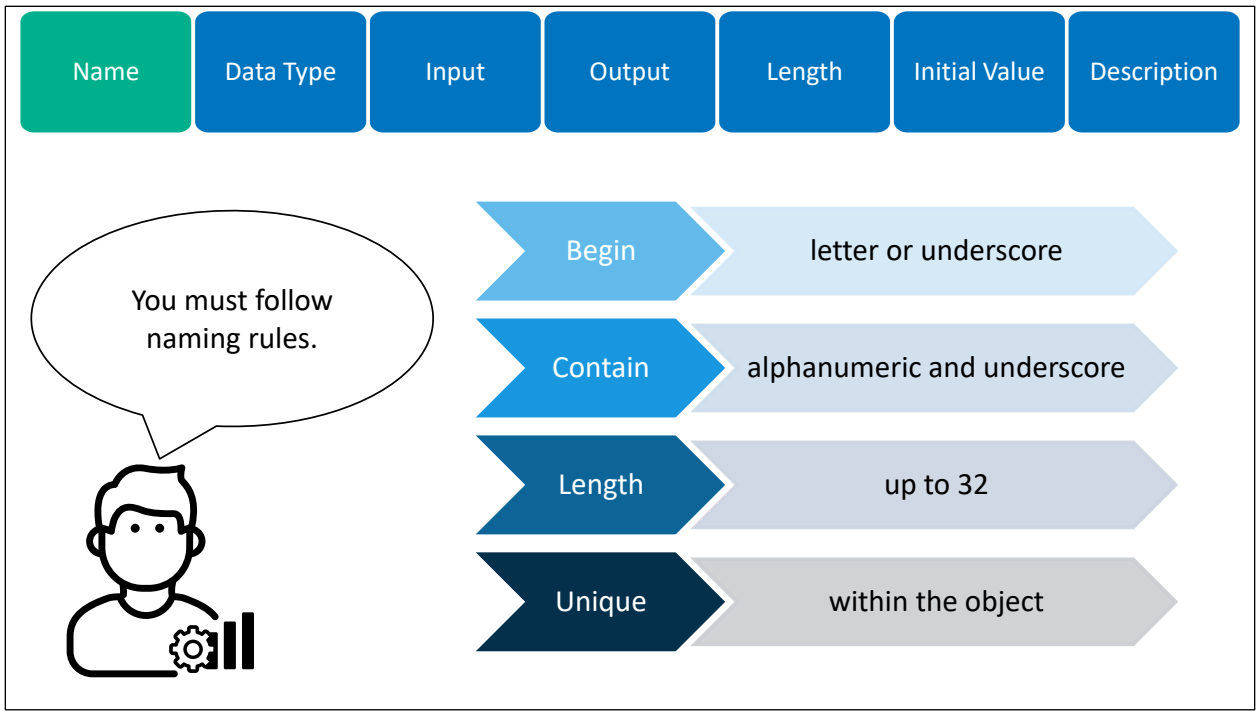


Variables have the properties shown here.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

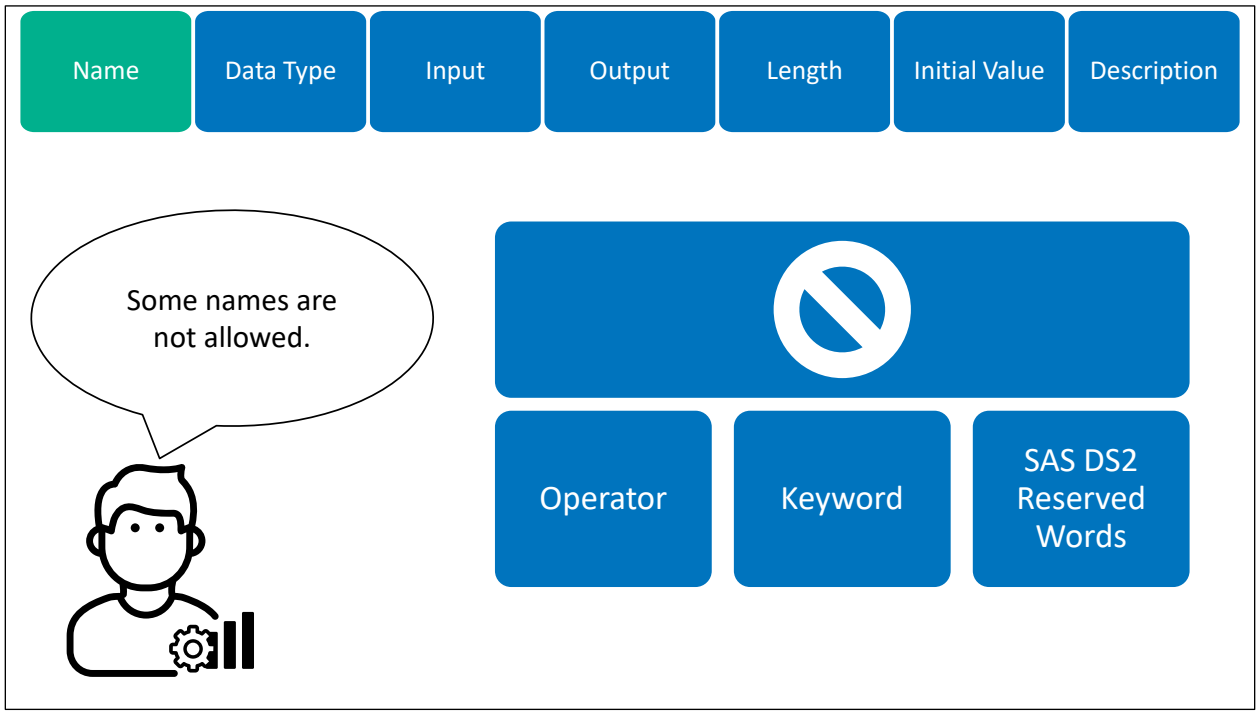


There are several ways to define variables in an object. You can define variables by importing their properties from a CSV file or JSON file. You can add variables from a code file, data table, decision, or rule set. You can define variables and their properties manually. For decisions only, any variables defined for an object (such as a rule set) are defined for the decision when you add that object to a decision.



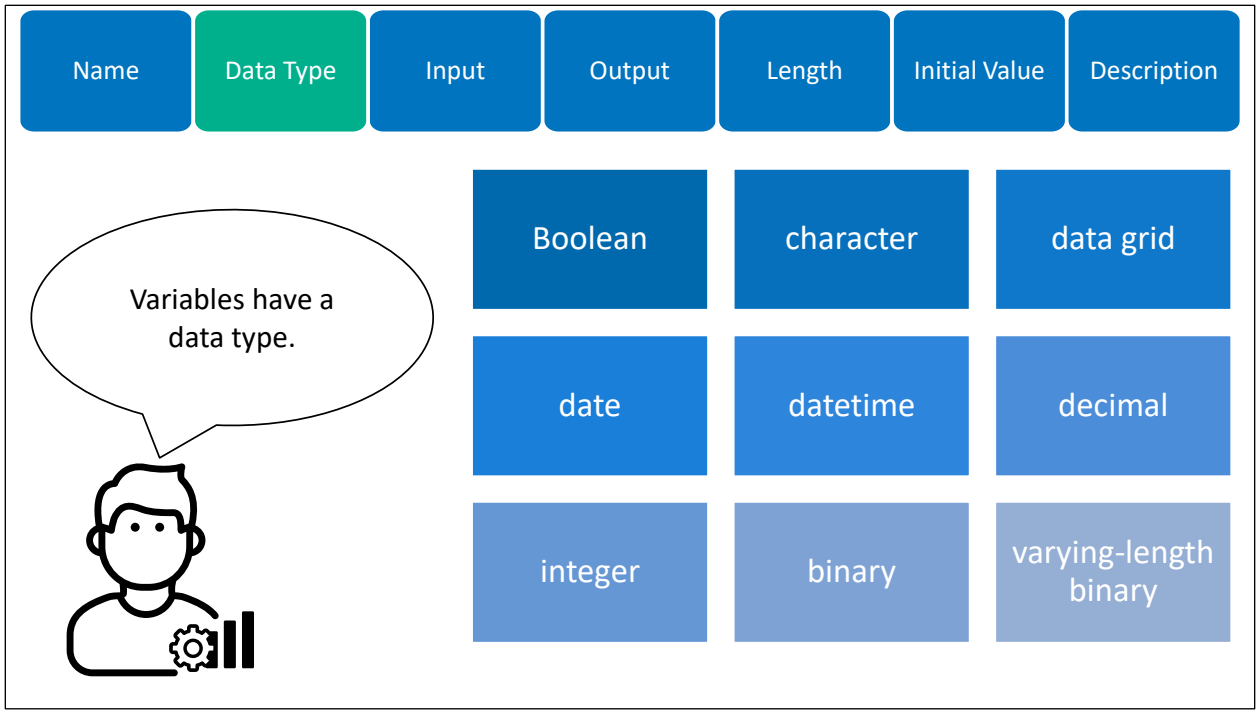
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Variable names must start with a letter or an underscore, and can contain only alphanumeric characters and the underscore. Names can be up to 32 characters long and must be unique within the object.

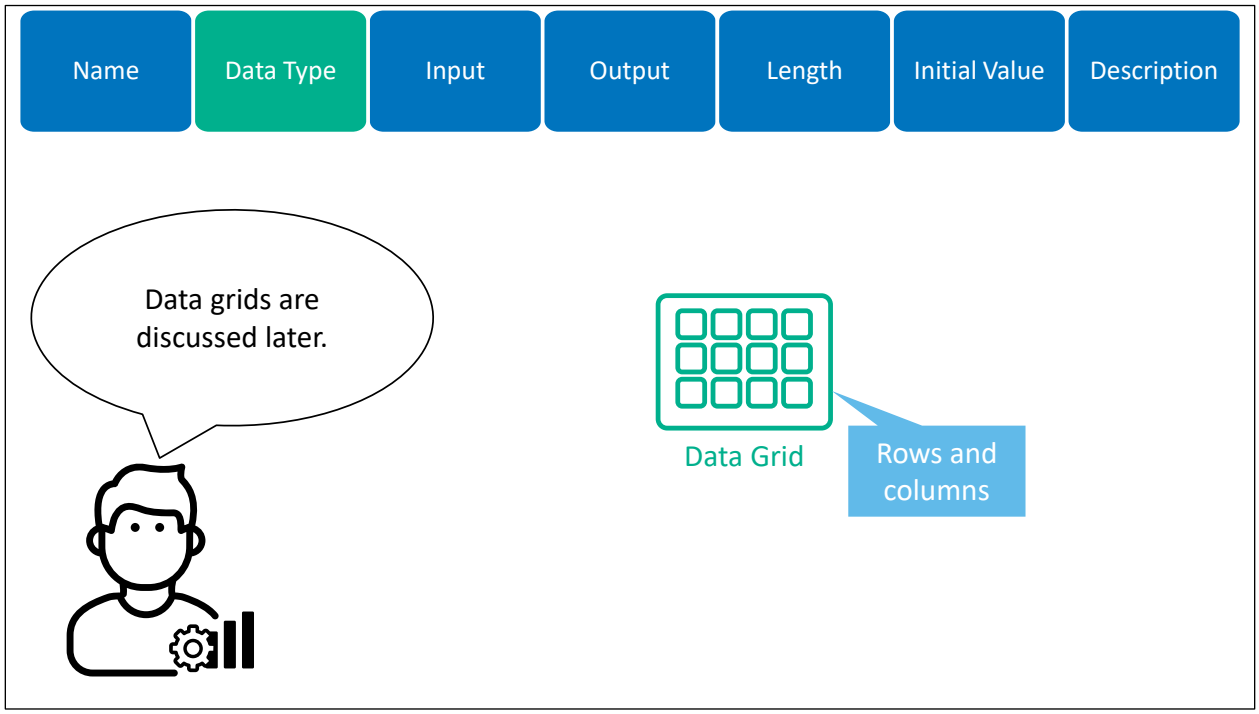


Some names are not allowed. You cannot use operators, keywords, or reserved words for the SAS DS2 language. Refer to *SAS Intelligent Decisioning: User's Guide* and *SAS DS2 Programmer's Guide* for more information. You cannot manually add a variable with a name that is not allowed. If you import a variable with a name that is not allowed, you receive an error message, and the variable is not imported successfully.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

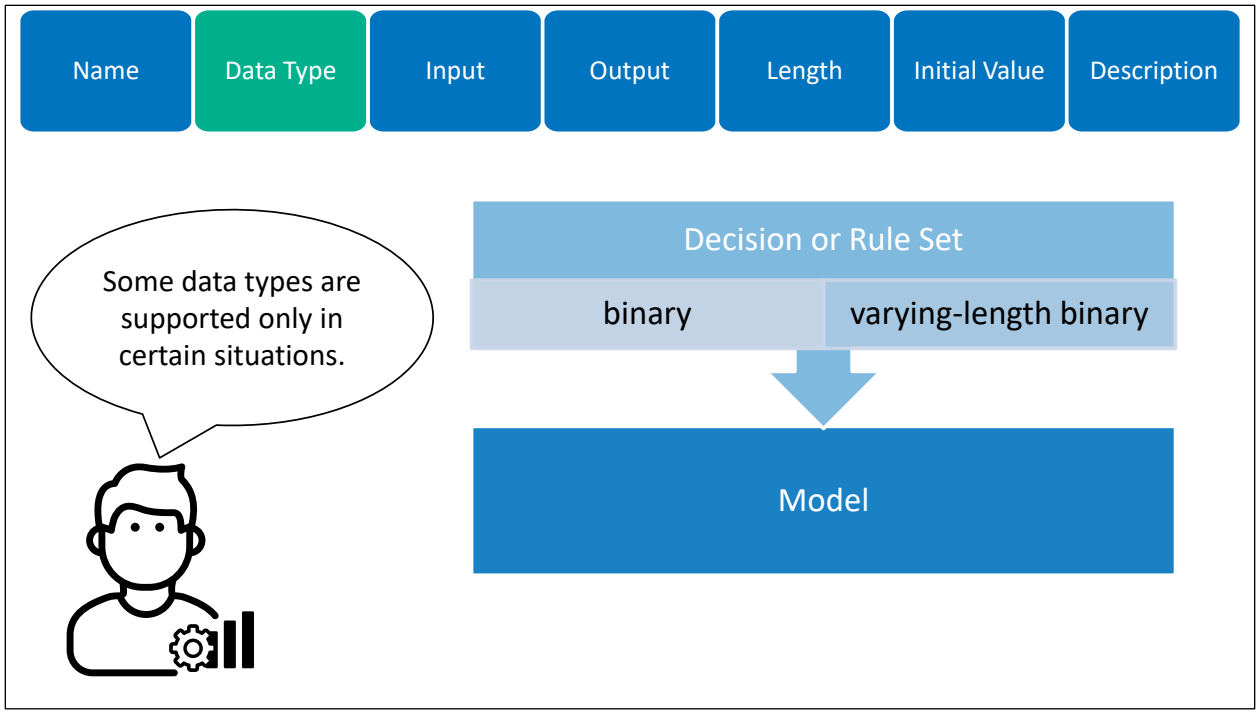


Variables can have the data types listed here.

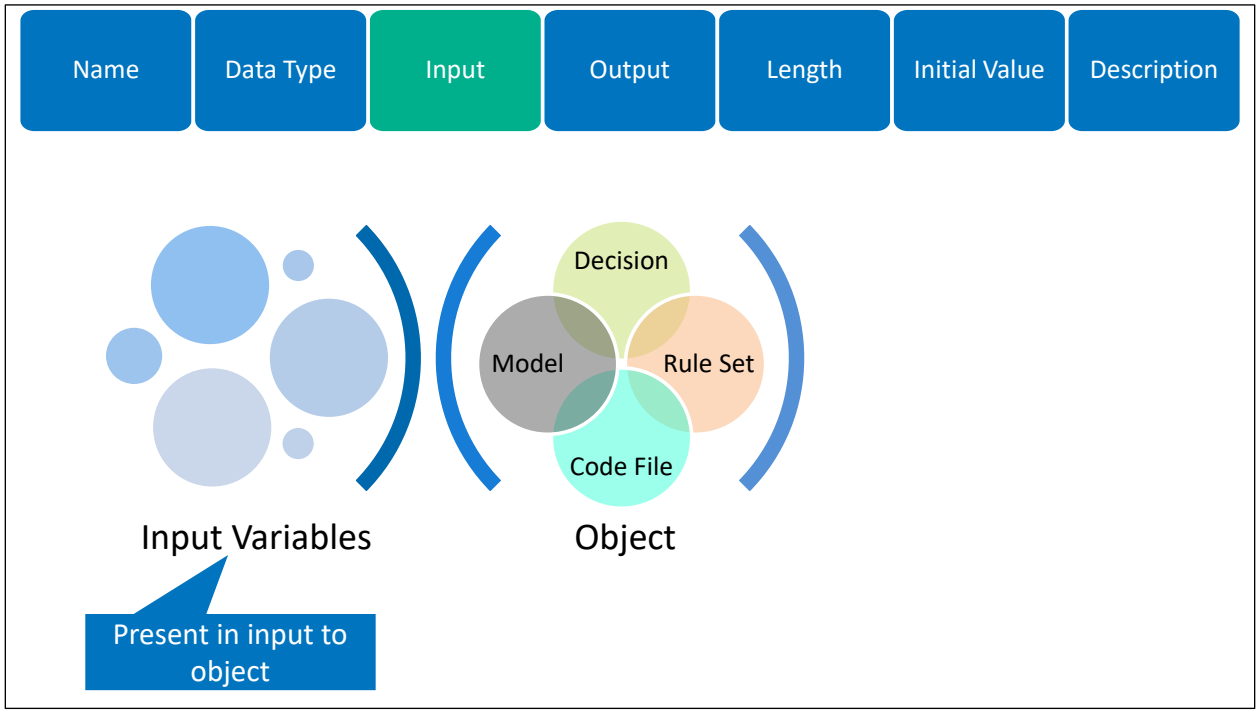


A data grid variable is a table, with rows and columns. Data grids are discussed in more detail later.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

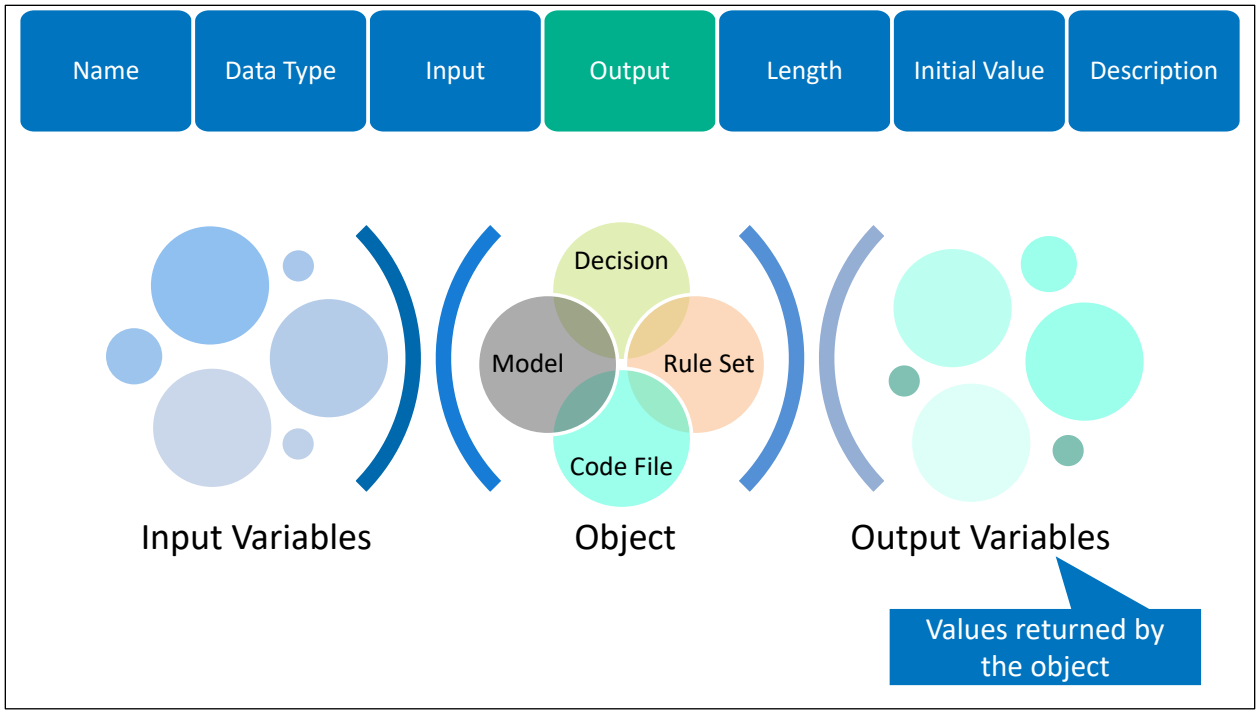


Binary and varying-length binary variables are supported in decisions and rule sets only. Binary variables are supported only as input variables or temporary variables in order to support models that require binary data.

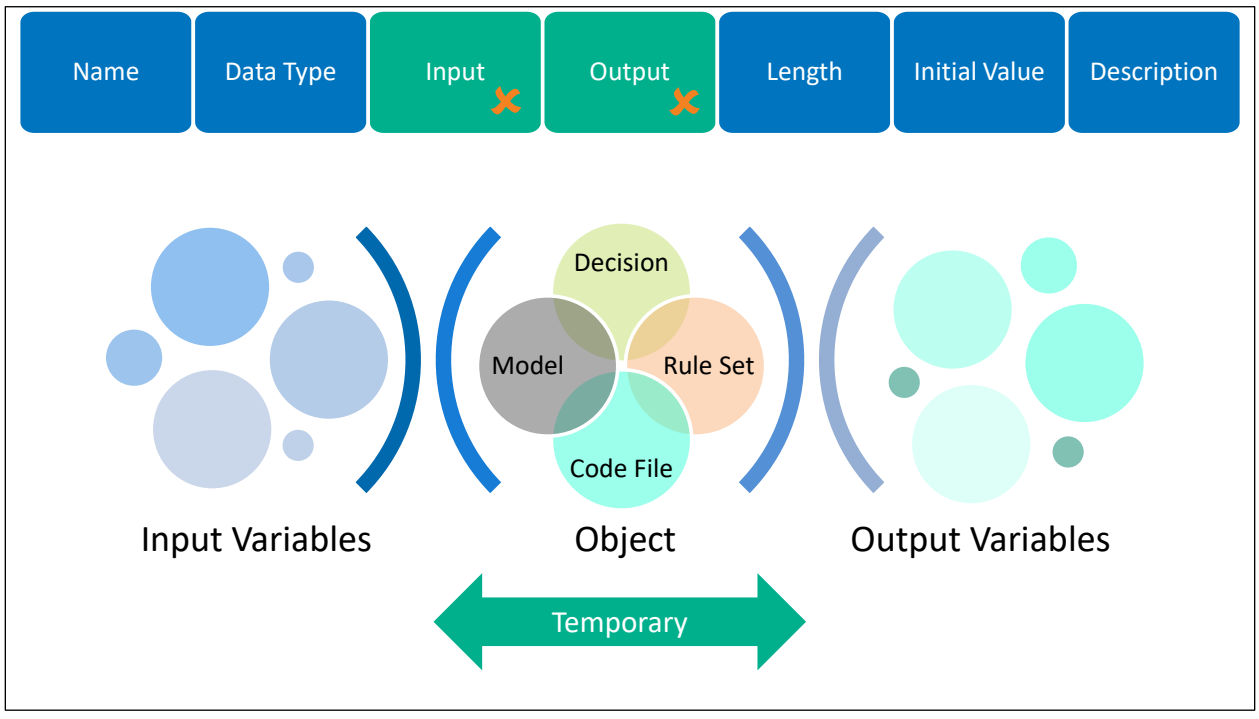


Variables with the input property are present in the input to the object. Therefore, you should make sure that this property is enabled only for variables that will be available to the object. Specifically for decisions, all input variables must exist in the input data when the decision is deployed.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

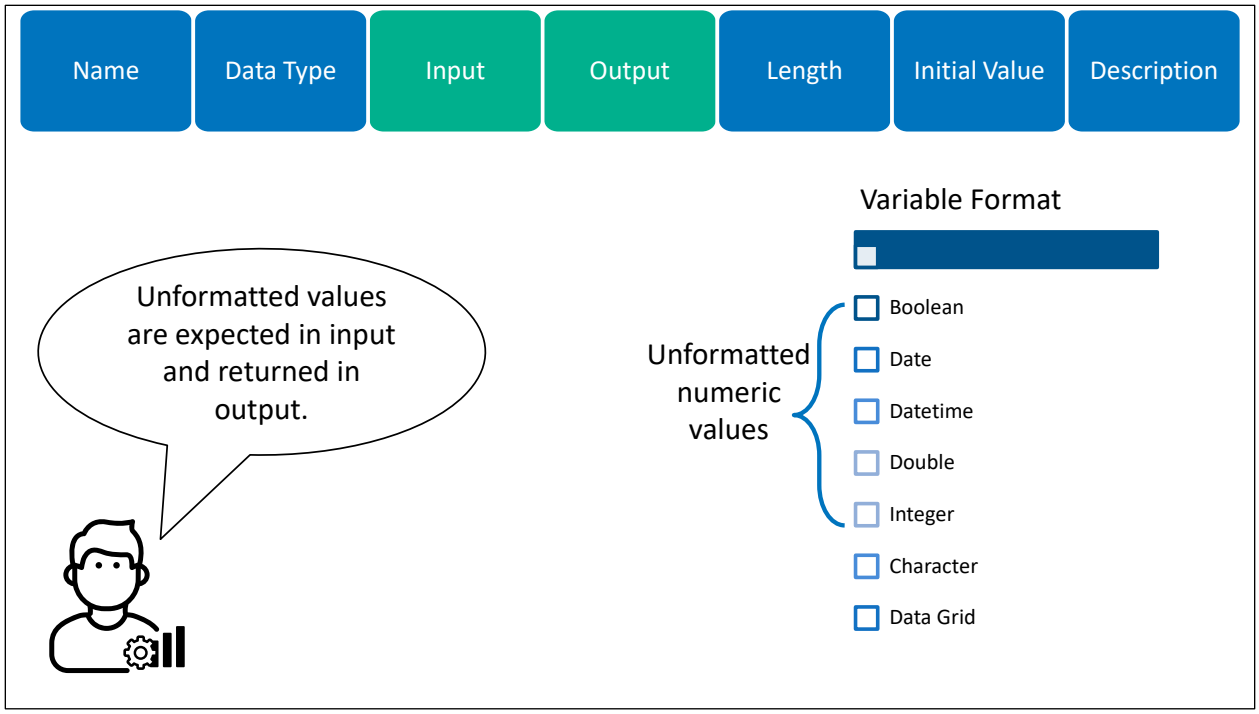


Variables with the output property are returned by the object. Specifically for decisions, output variable values are written to the output that is created when a decision is run.

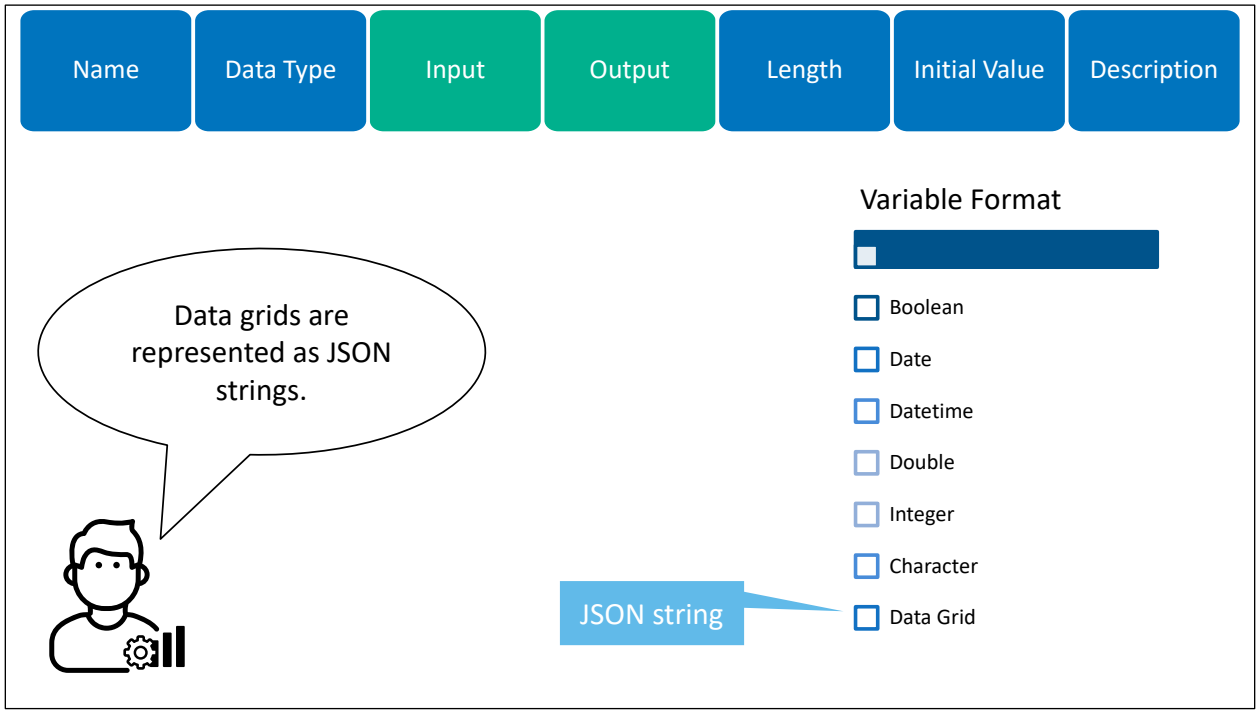


If a variable has neither the input nor the output property, then it is temporary. It exists only while the object is being processed.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

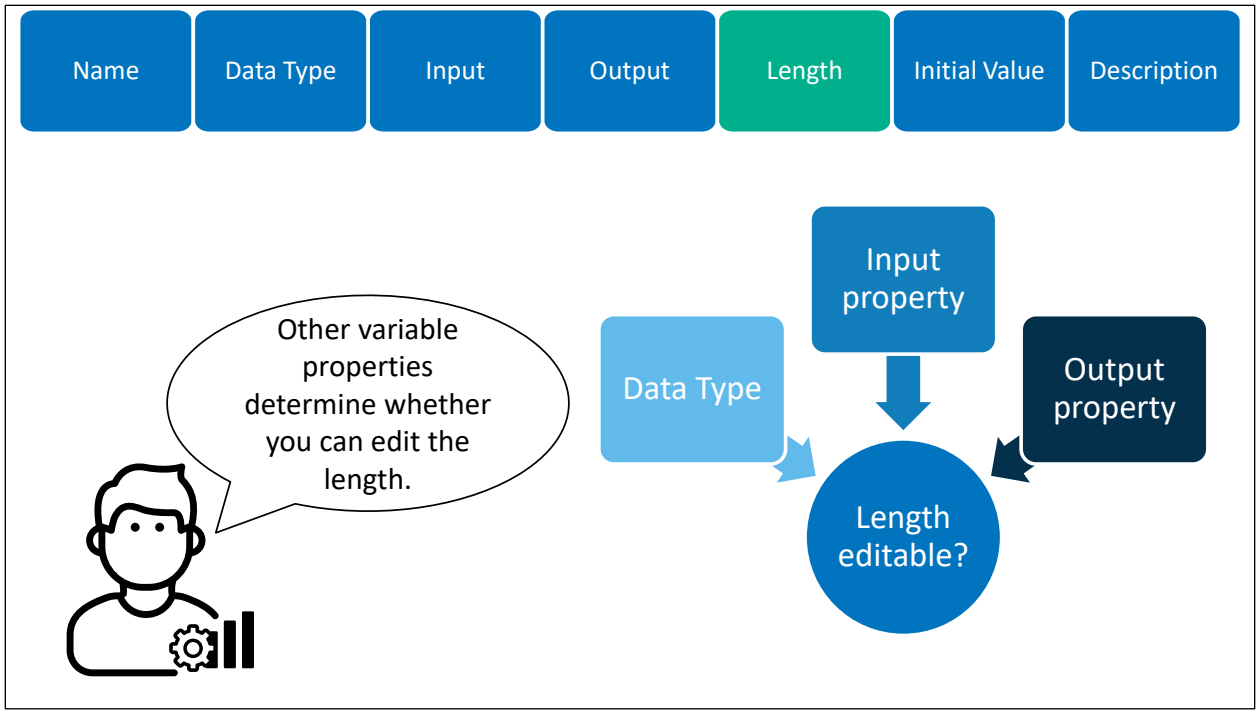


You need to consider the formatting of input and output data for decisions and rule sets. Unformatted values are expected in input and returned in output. You might need to perform extra steps (such as passing in a calendar date as a string and converting to a date) to perform the processing needed for your decision.

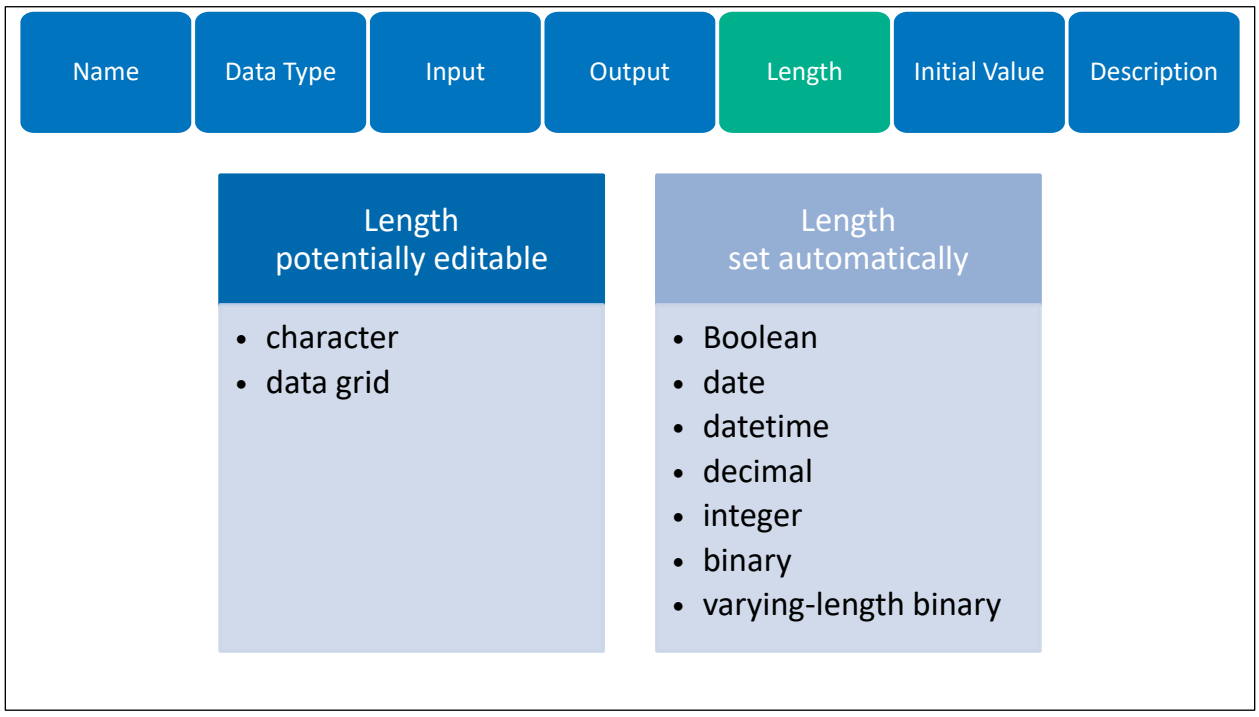


Data grids are represented as JSON strings. Data grids and JSON strings are discussed in more detail in a later lesson.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

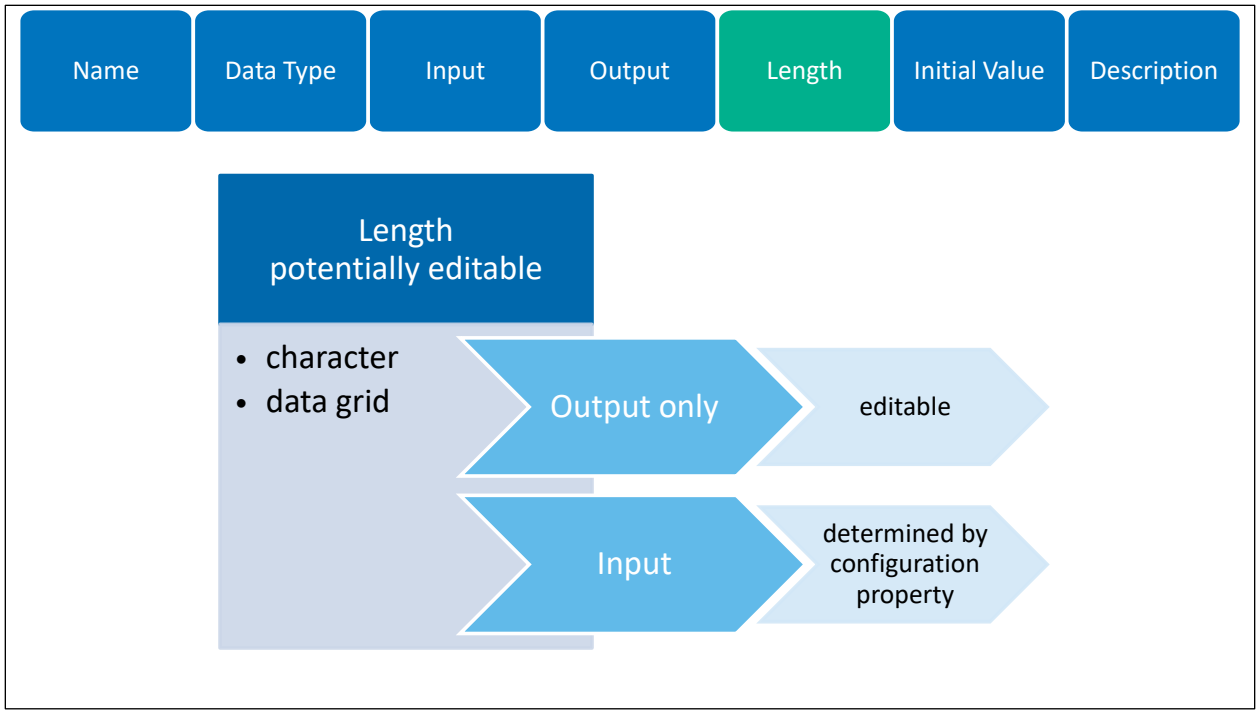


The data type, input, and output properties of the variable determine whether you can edit the length setting.

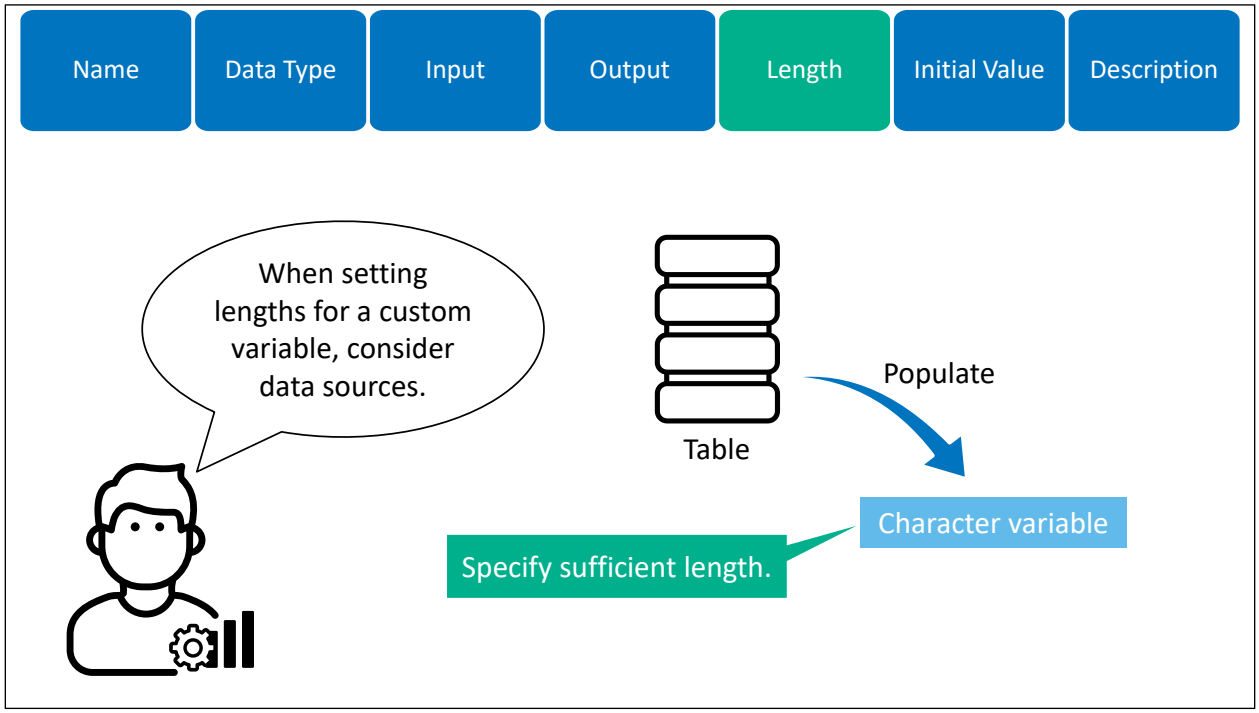


The length property is potentially editable for character and data grid variables only. Lengths for all other data types are set automatically.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

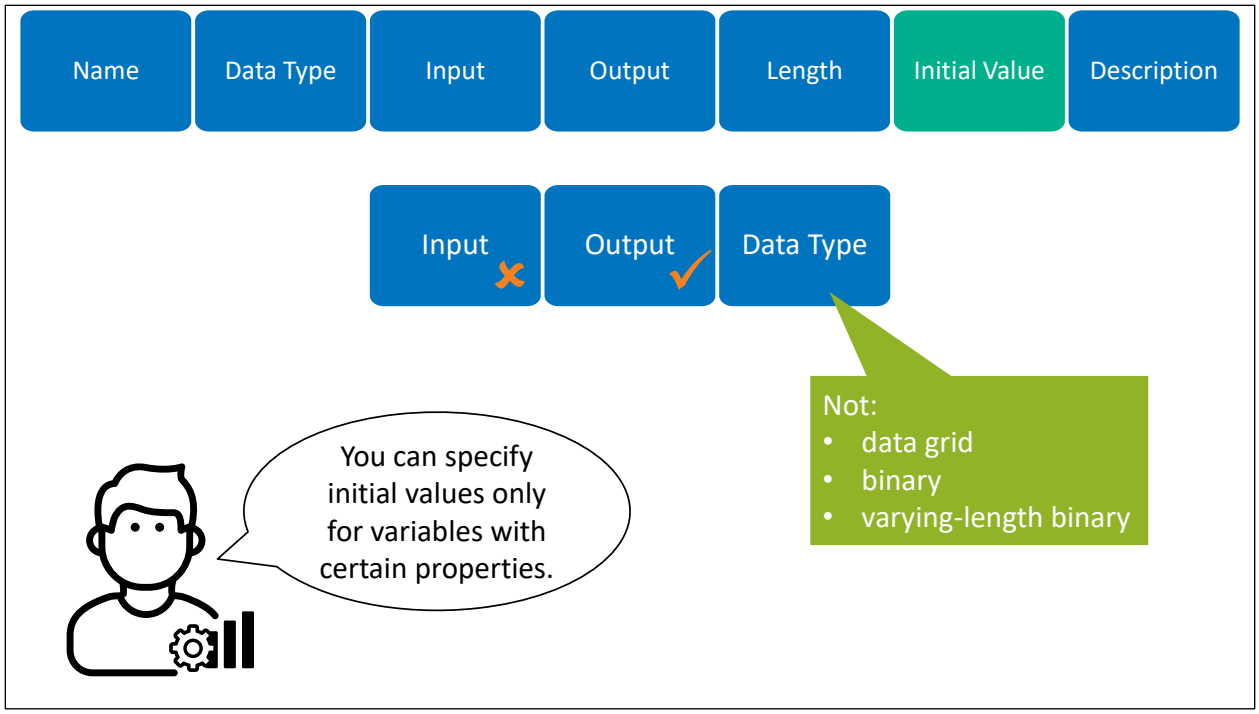


If a character or data grid variable is output-only, the length is editable. If it has the input property, a configuration property determines whether you can edit the length. Refer to the *SAS Intelligent Decisioning User's Guide* for more information.

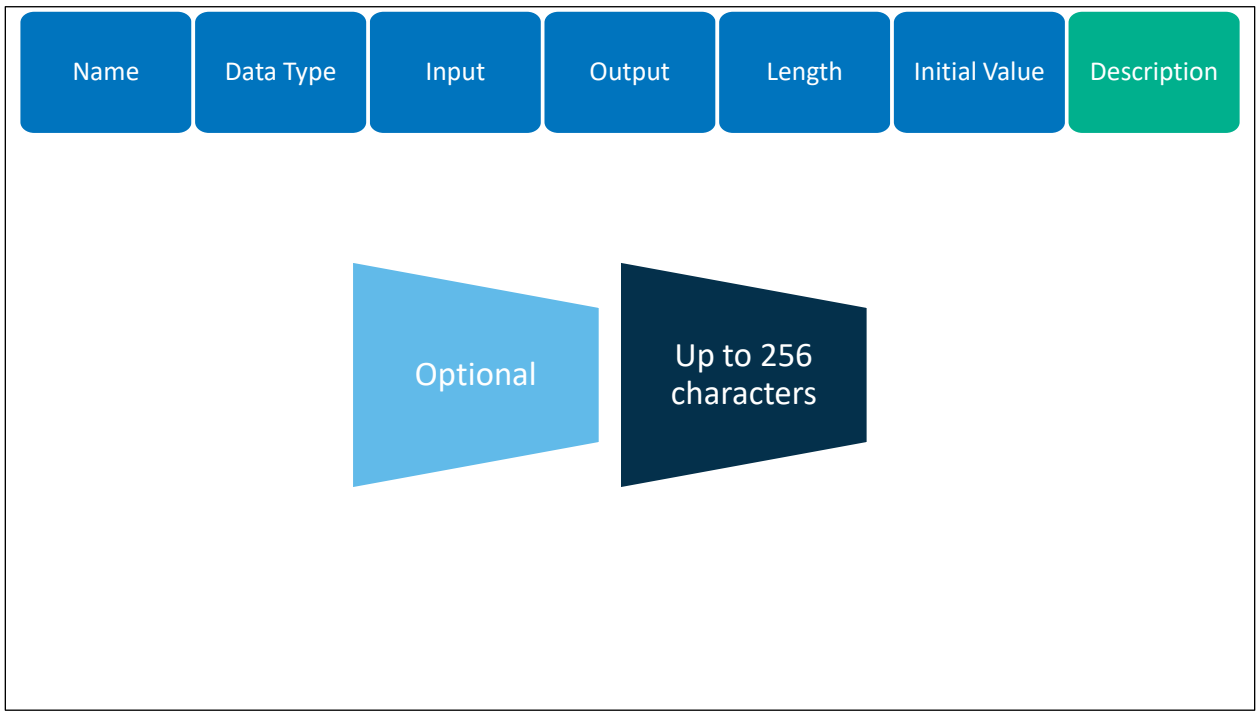


When setting variable lengths for a custom variable, be sure to consider any data sources that you will be using to populate the variable value. Be sure that the length is sufficient to store the value.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

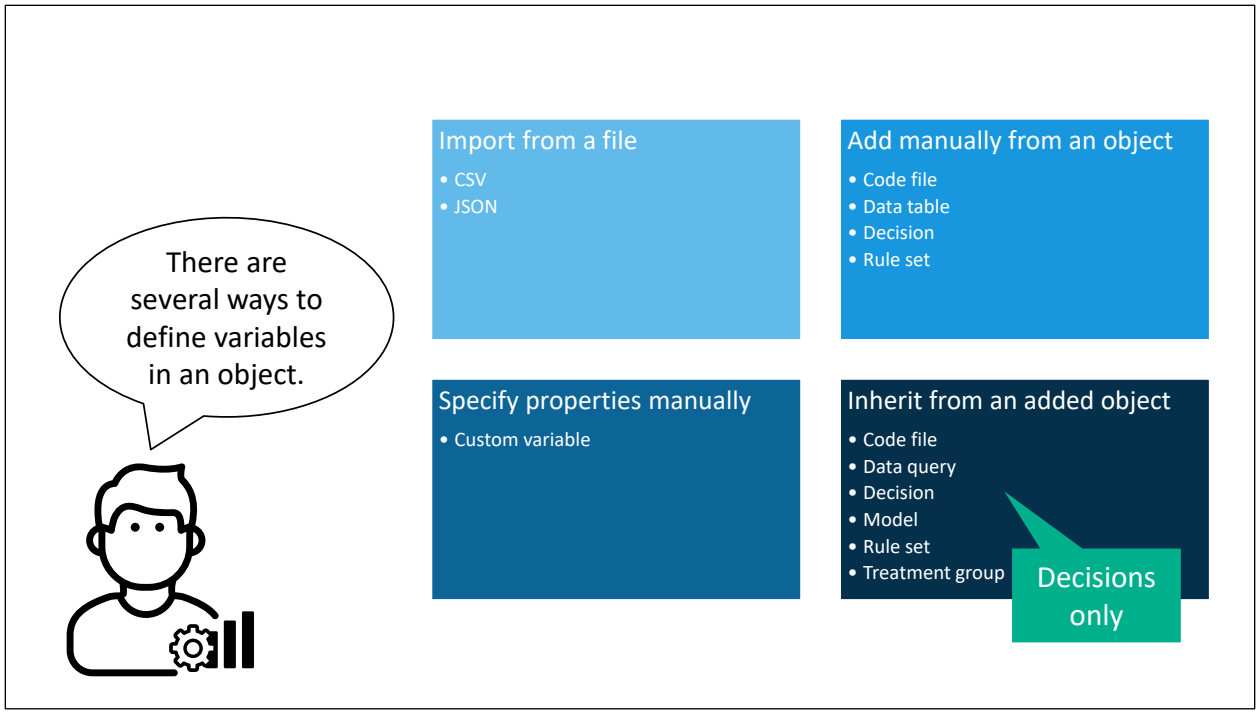


You can specify initial values only for variables that are output but not input variables, and are not data grid, binary, or varying-length binary type.

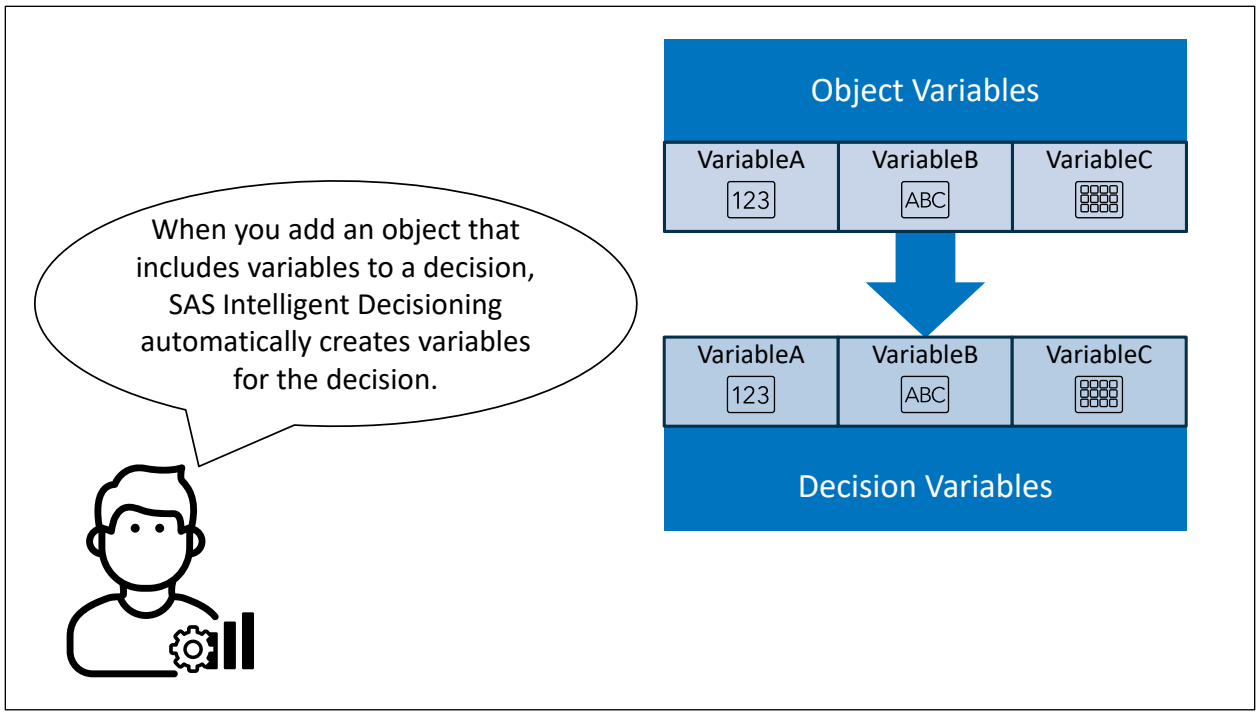


Description is optional and can be up to 256 characters long.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

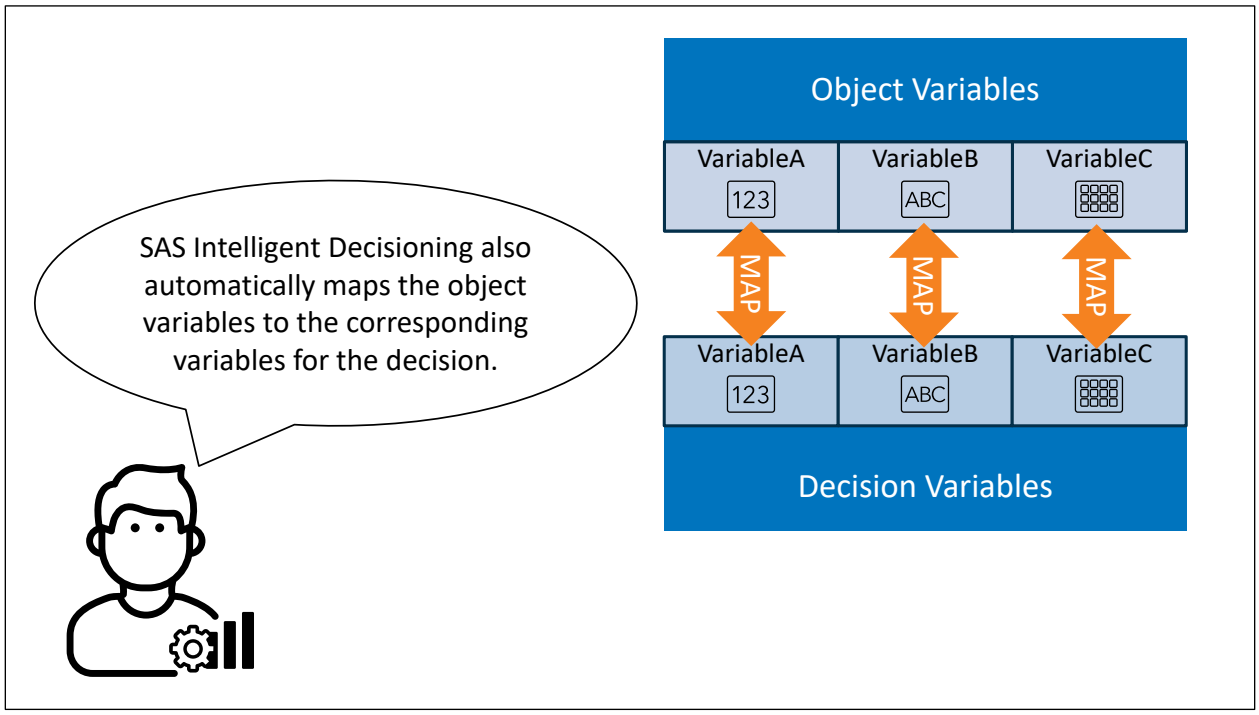


Recall that there are several ways to define variables in an object. You can define variables by importing their properties from a CSV file or JSON file. You can add variables from a code file, data table, decision, or rule set. You can define variables and their properties manually. For decisions only, any variables defined for an object (such as a rule set) are defined for the decision when you add that object to a decision.

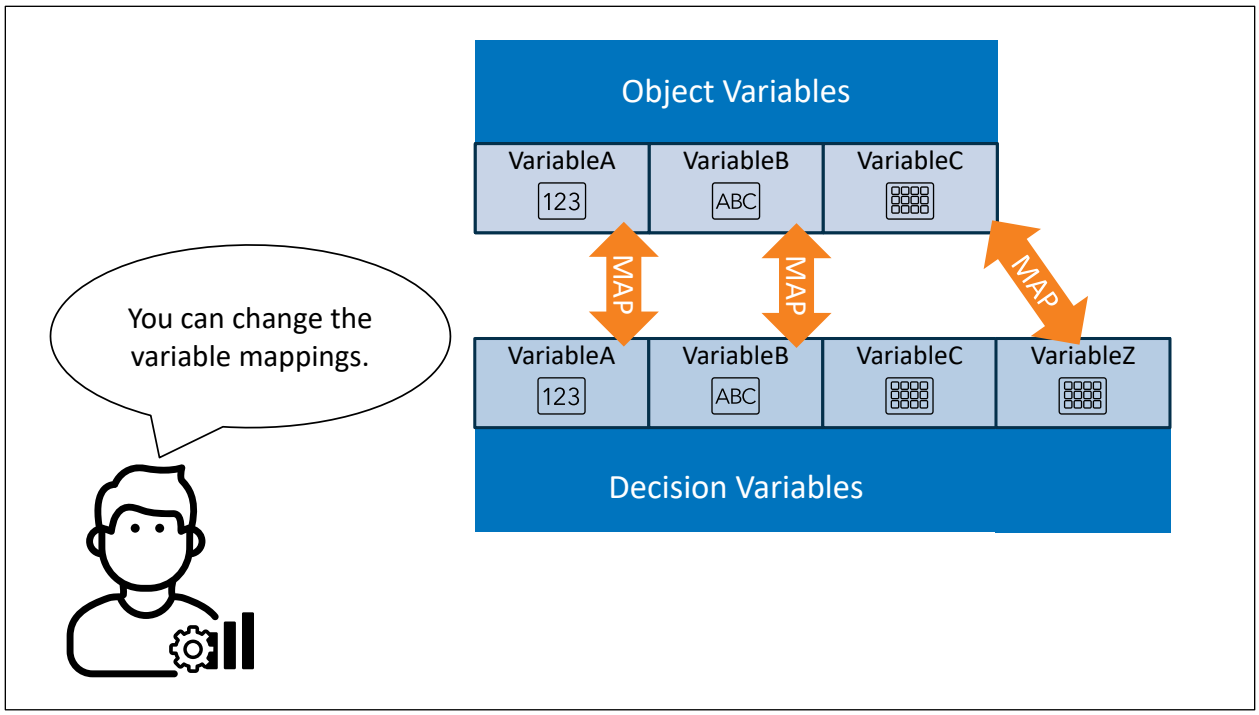


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

When you add an object that includes variables (such as a rule set or model) to a decision, SAS Intelligent Decisioning automatically creates variables with the same name and data type for the decision.

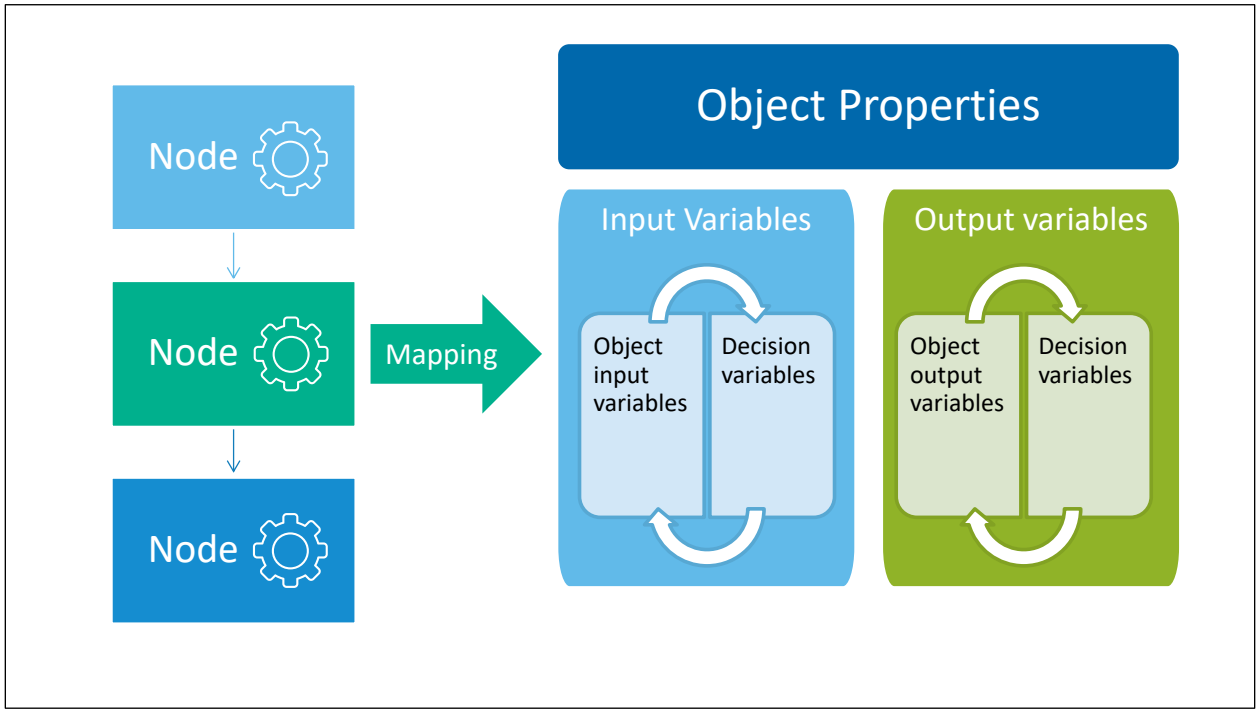


SAS Intelligent Decisioning also automatically maps the object variables to the corresponding variables for the decision.

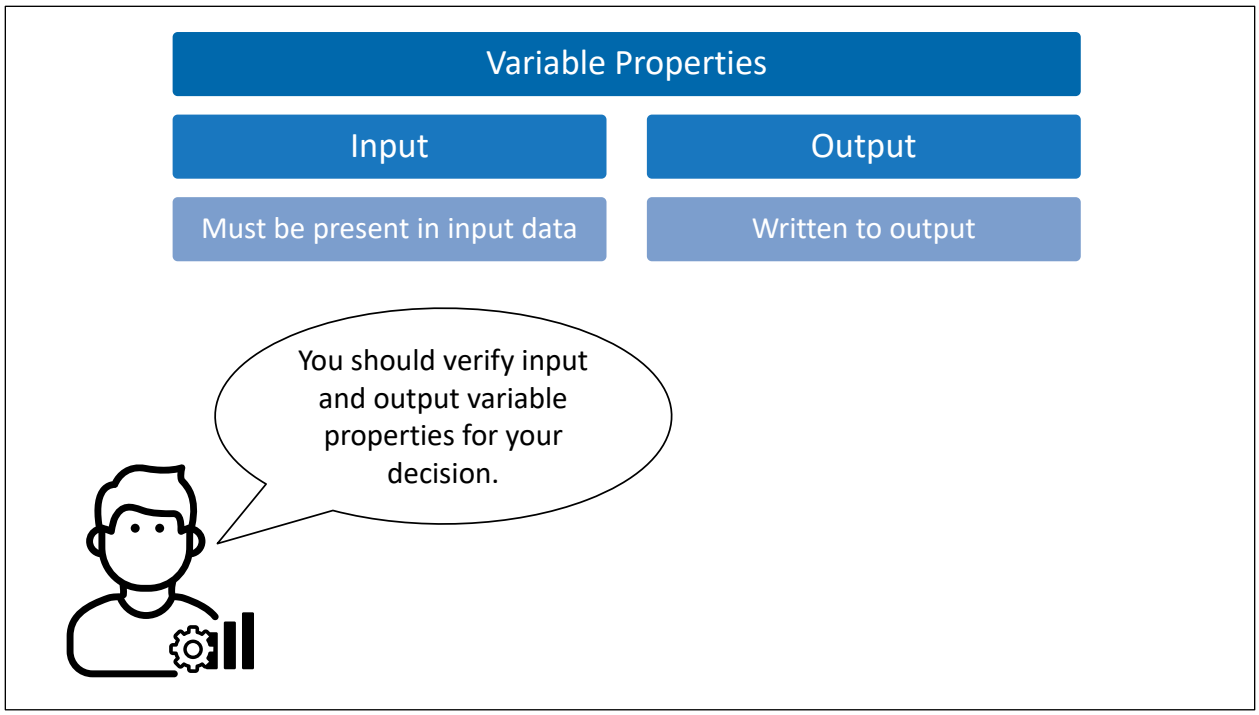


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

If needed, you can change the variable mappings on the property pane for the given object in the decision.

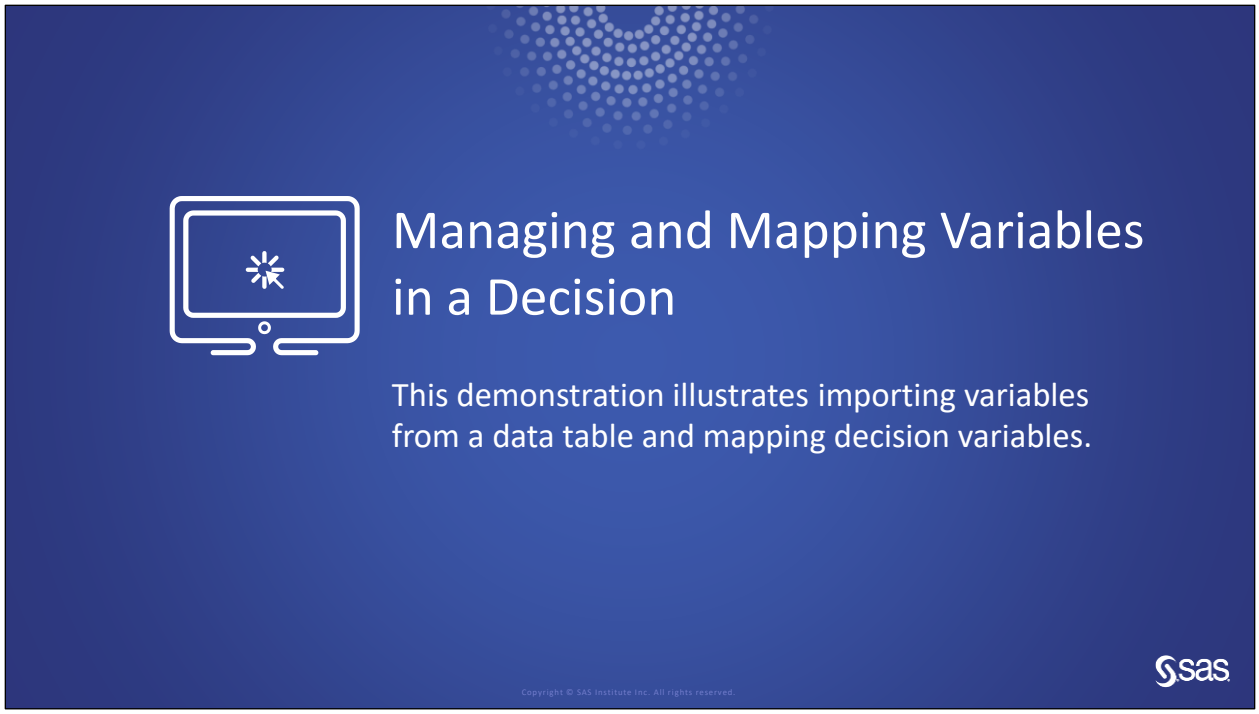


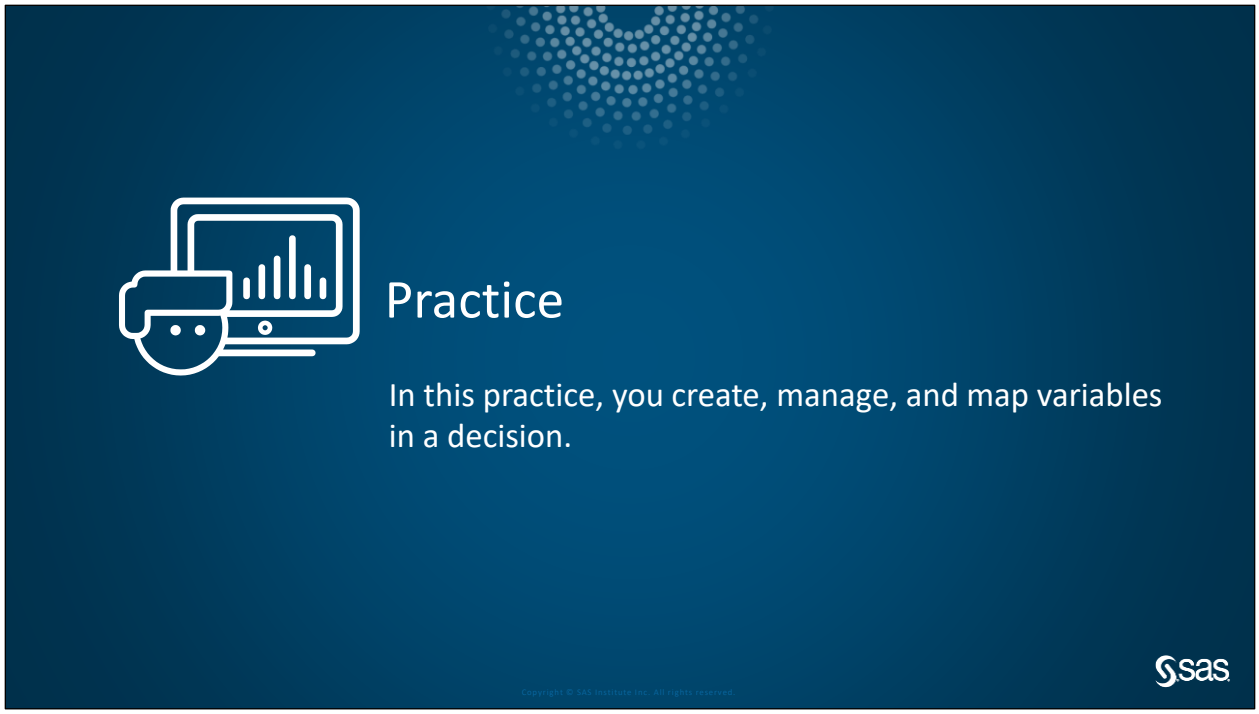
You control variable mapping using the Input Variables and Output Variables properties panes of an object. The two panes control how the input and output variables for the object are mapped to decision variables.



Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

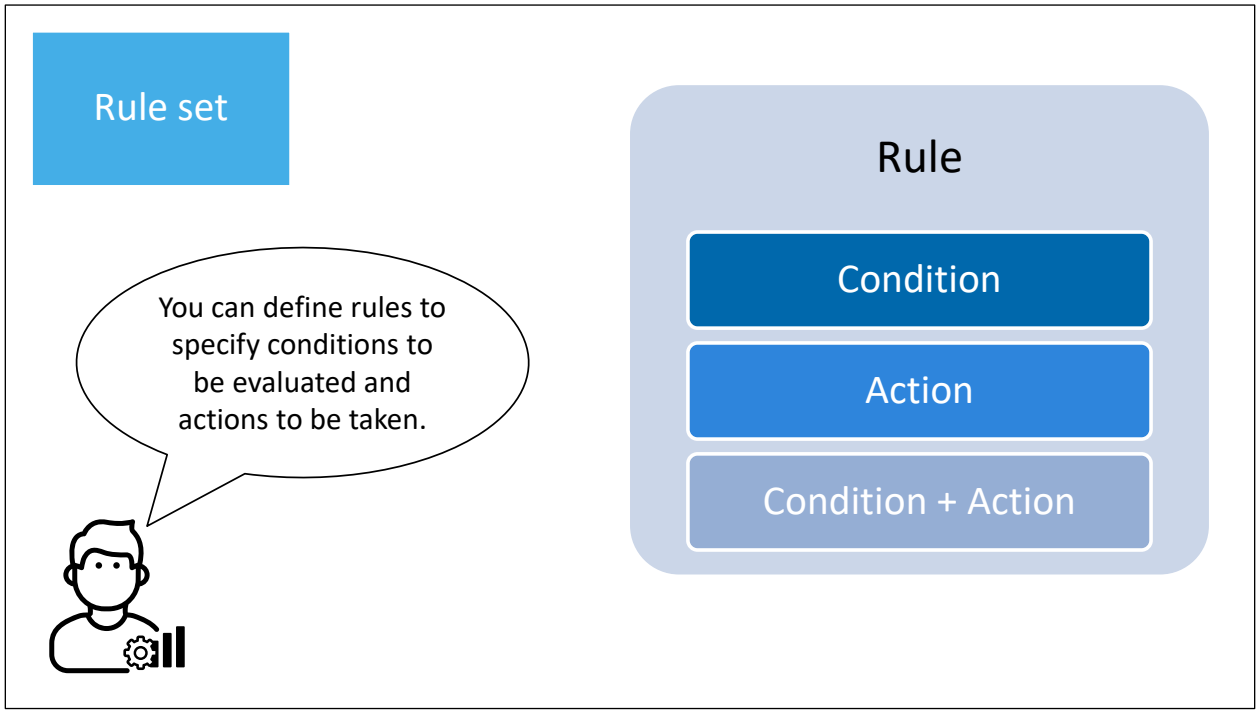
Recall that variables with the Input property must be present in the input data for a decision, and variables with the output property are written to the output that is created when a decision is run. You should verify these property settings for your decision. In particular, when you import variable properties or add an object that includes variables to your decision, you might need to change the default settings for these properties.



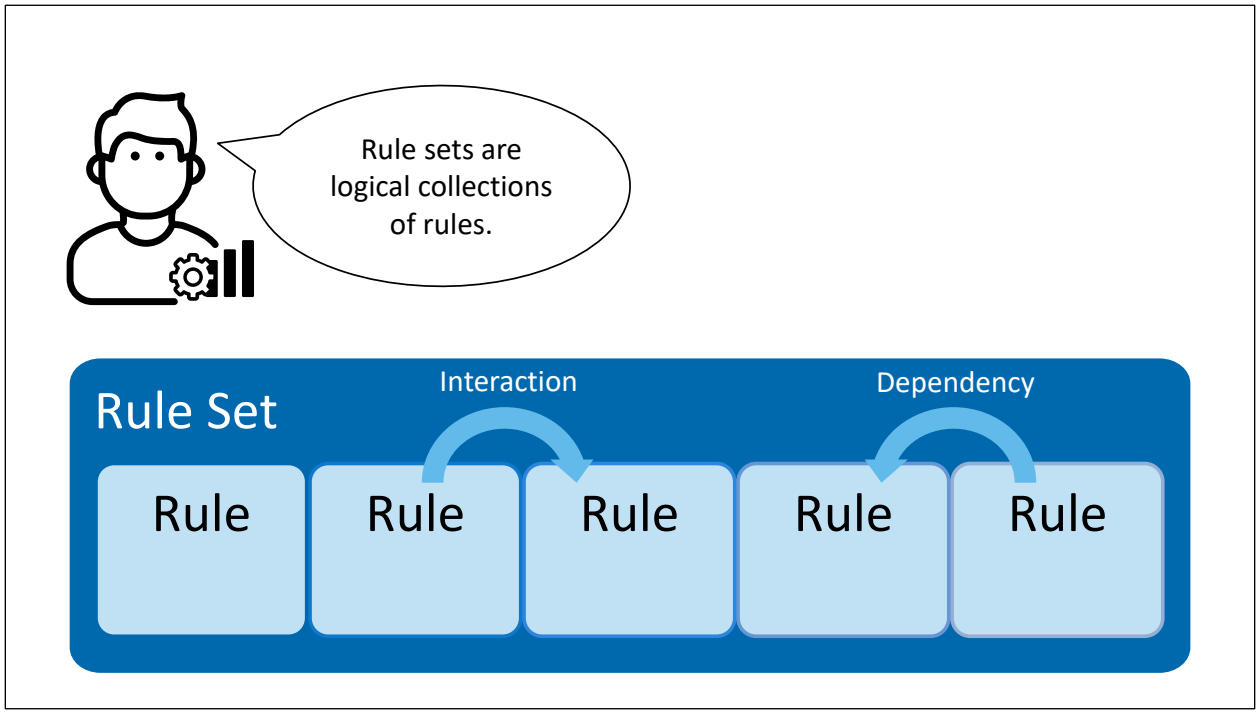


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.2 Rule Sets and Rules

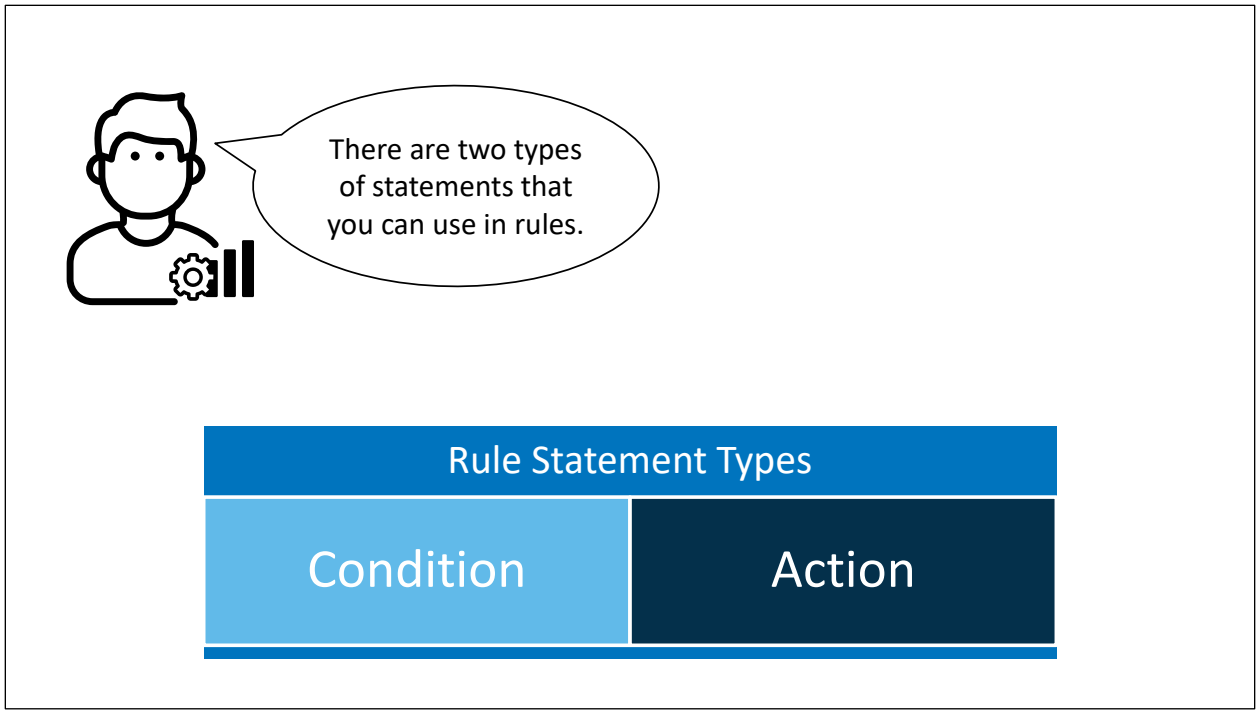


You can define rules to specify conditions to be evaluated and actions to be taken.

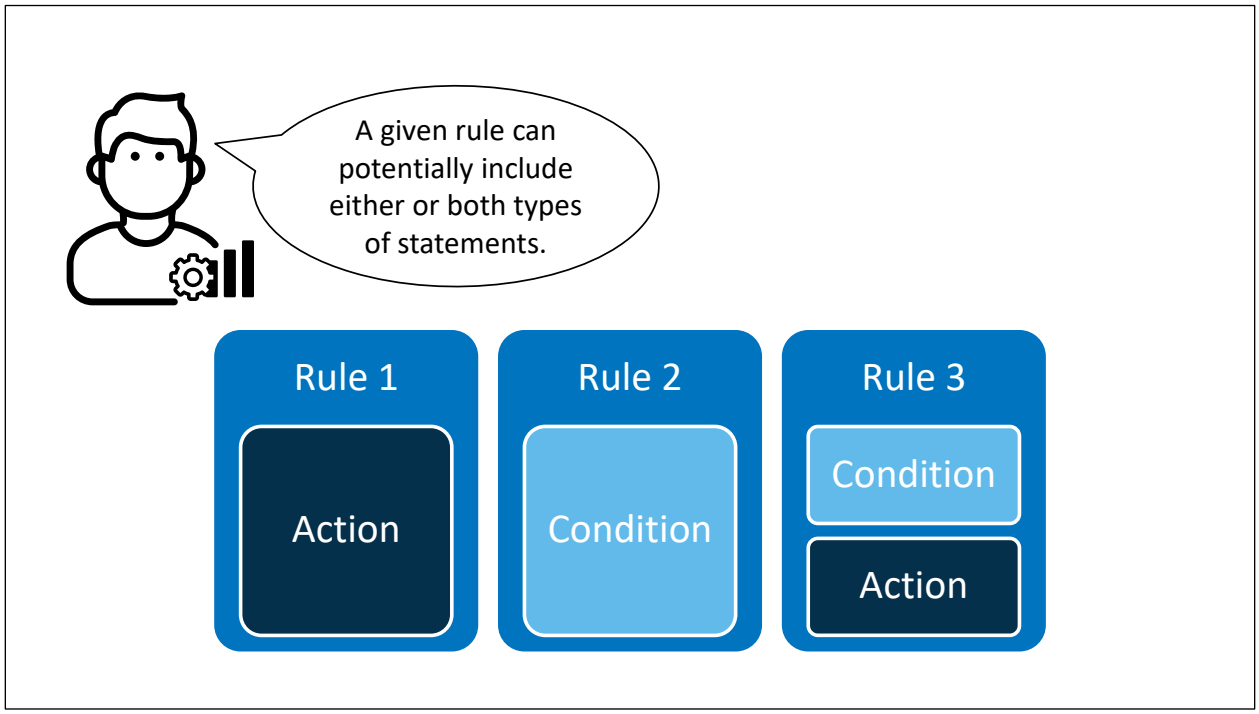


Rule sets are logical collections of rules that are processed together after they are published. Rules within a rule set can have interactions with or dependencies on others in the set.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

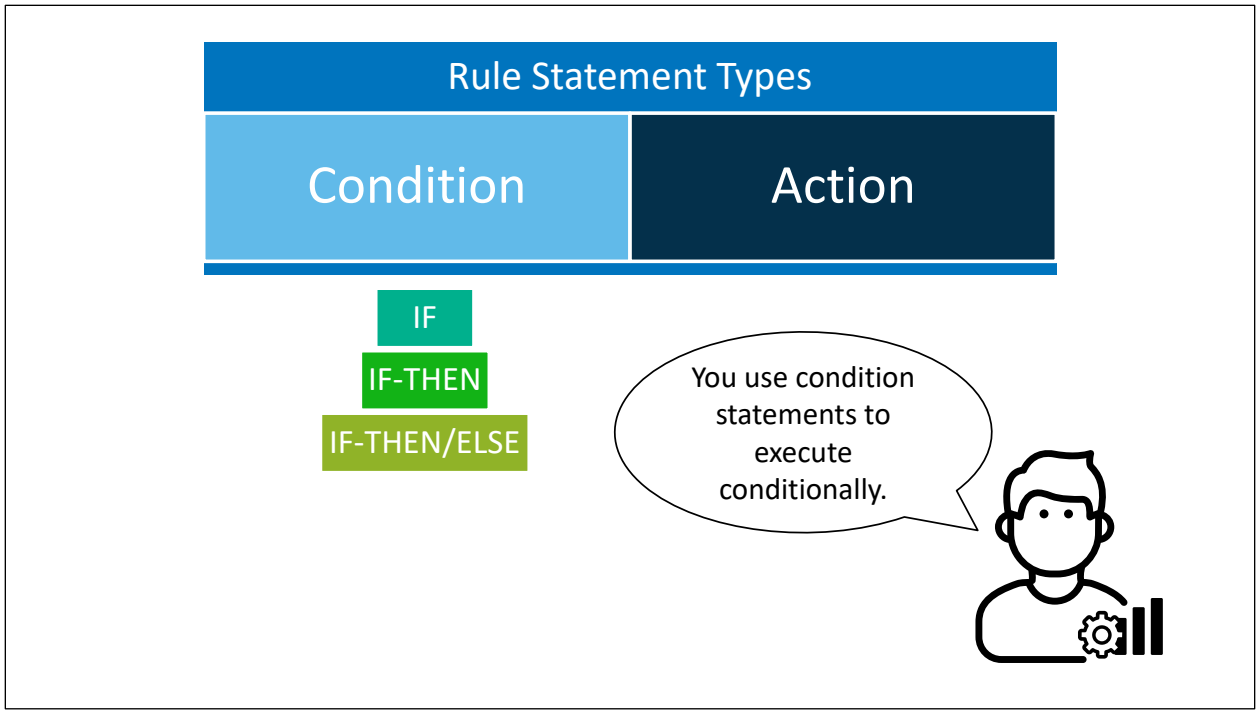


There are two types of statements that you can use in rules: conditions and actions.

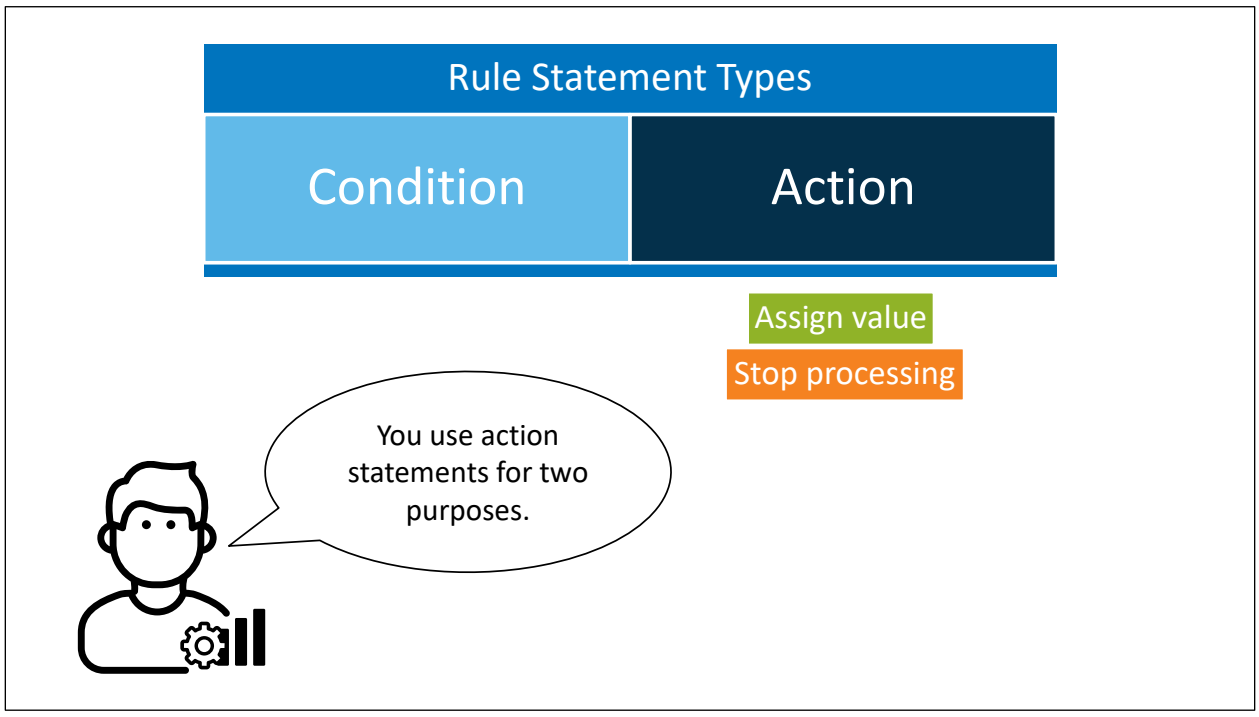


A given rule can potentially include either or both types of statements.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

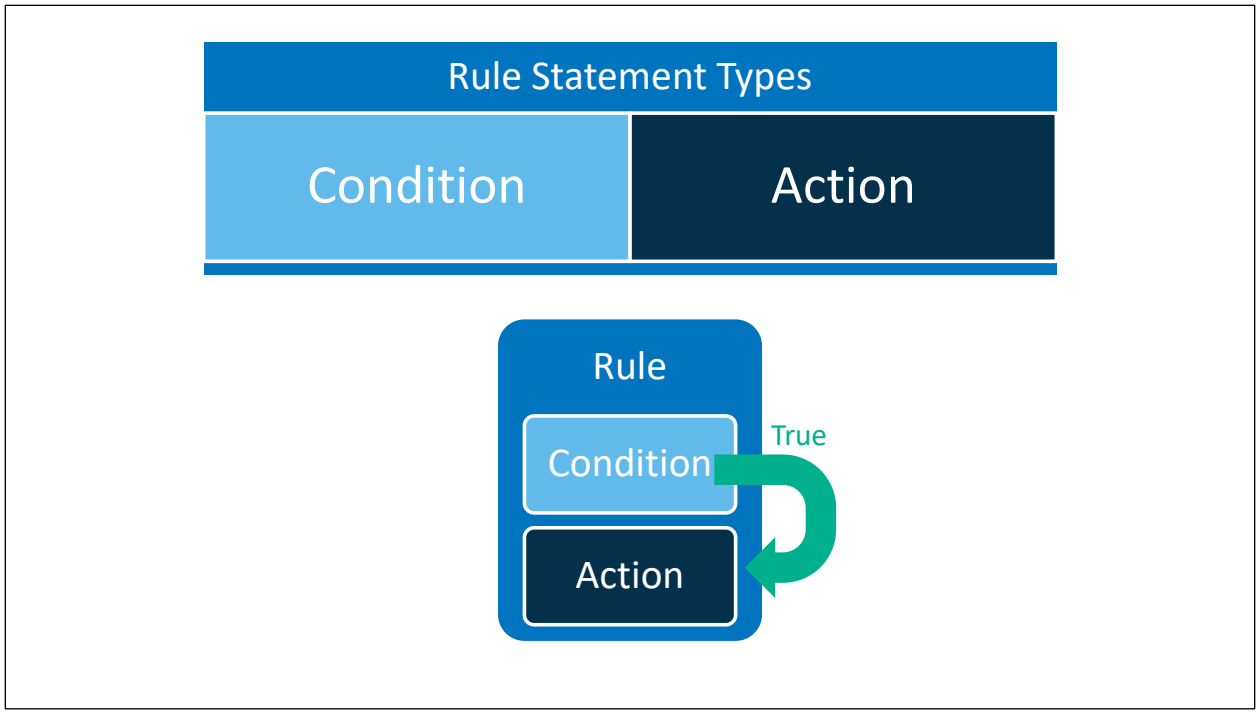


You use condition statements to execute conditionally. There are three types of conditional statements: IF, IF-THEN, and IF-THEN/ELSE.

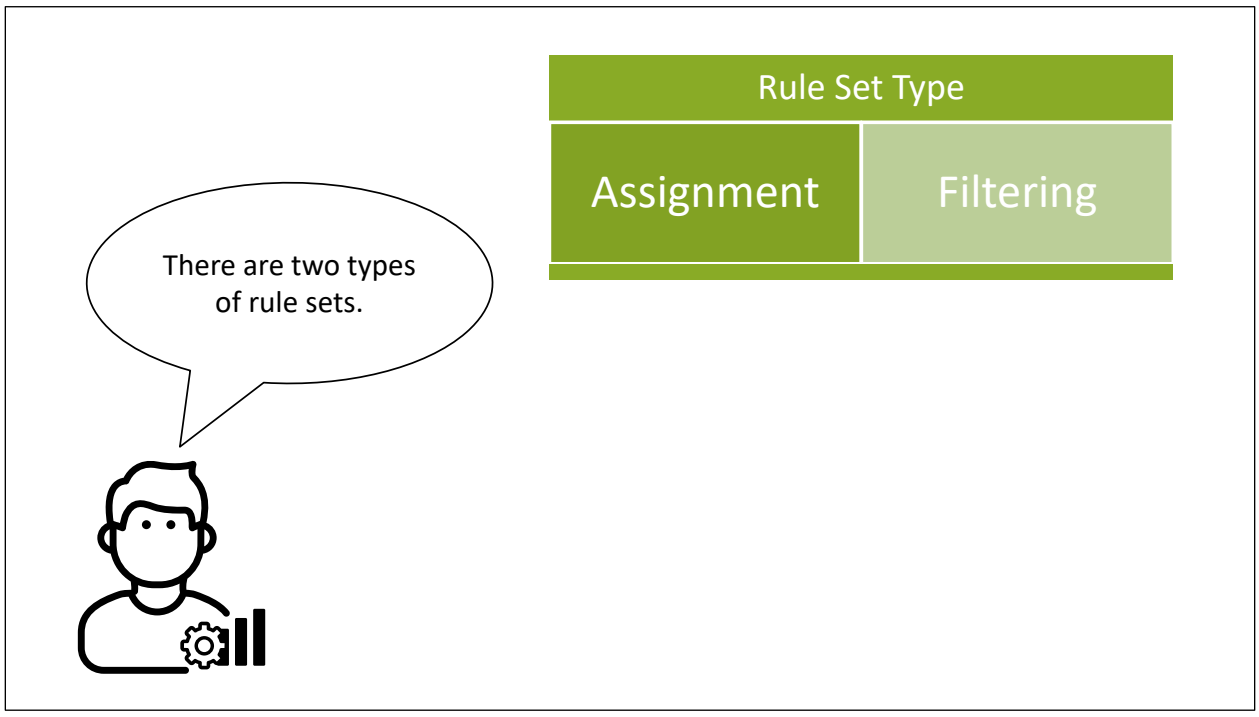


You use action statements to assign values to variables or to stop processing.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

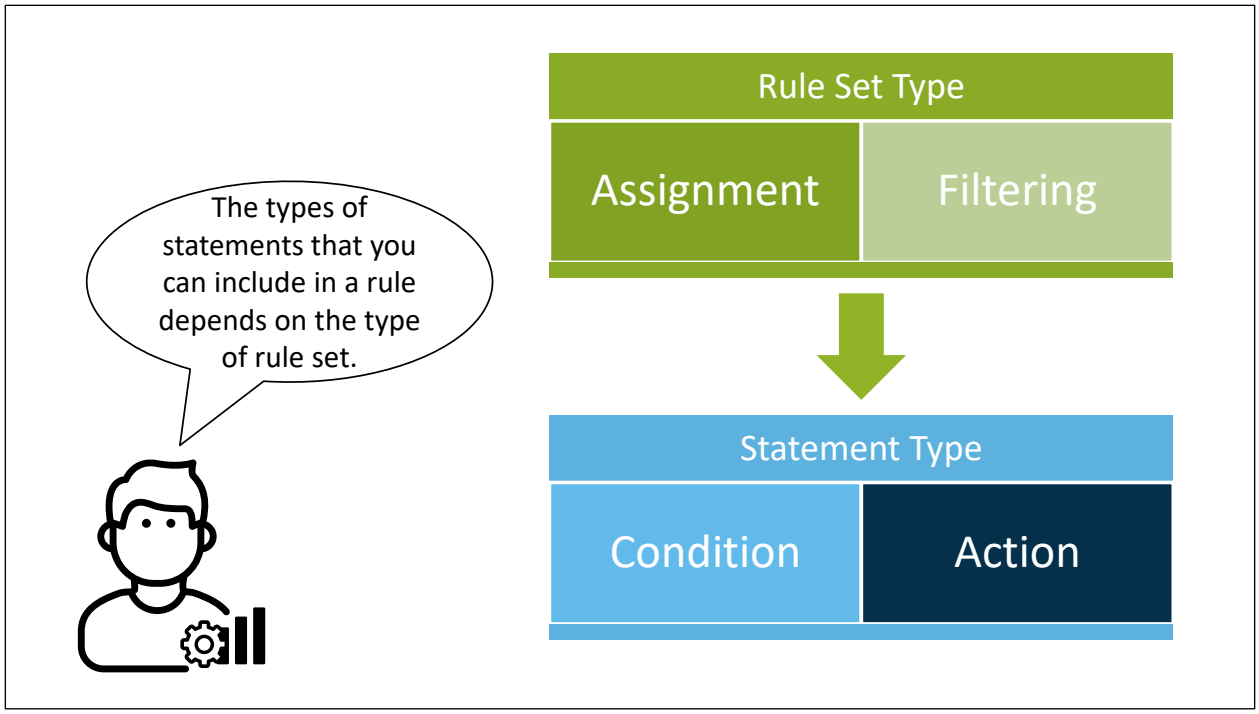


When you combine condition statements with action statements, the action is executed only when the condition is true.

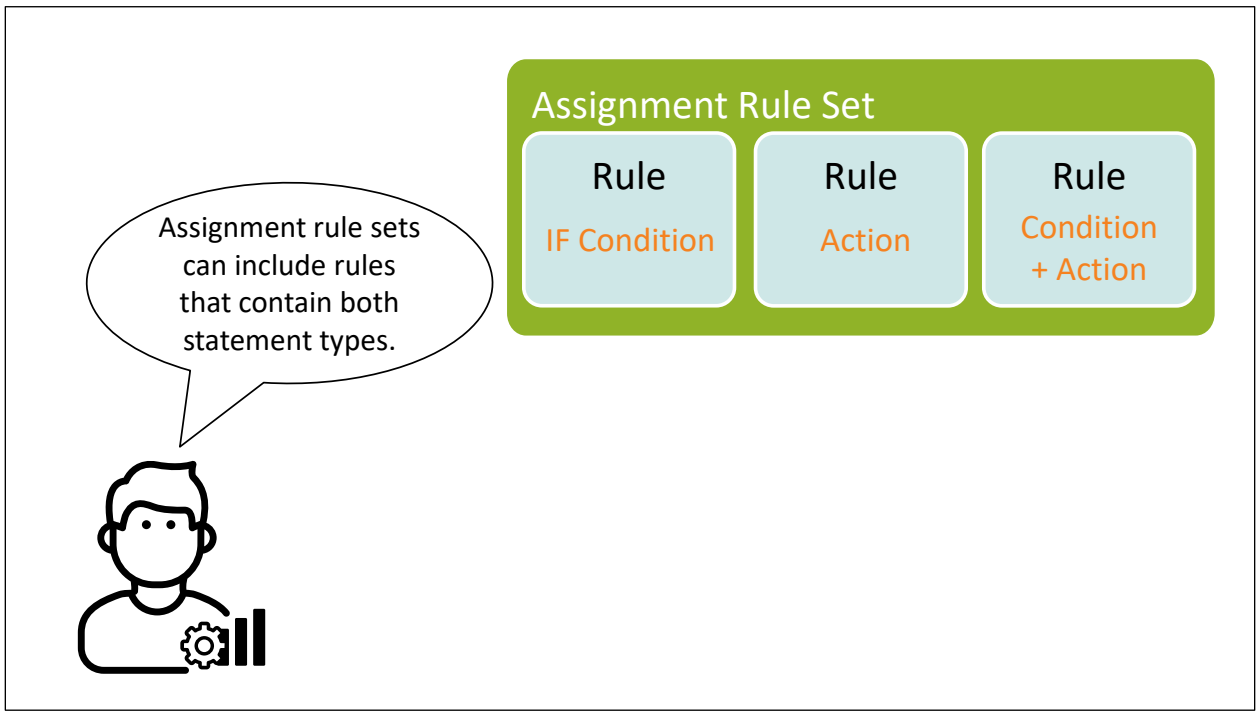


You can create two types of rule sets: assignment and filtering.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

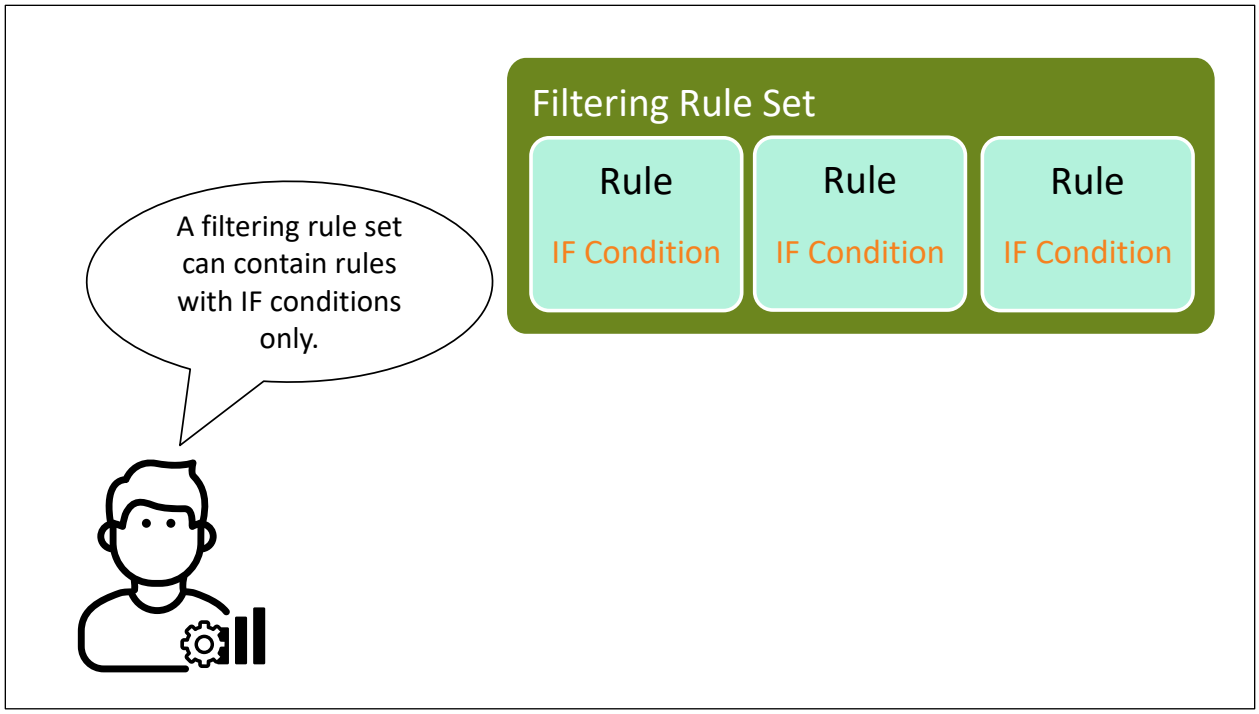


The types of statements that you can include in a rule depends on the type of rule set that you are working with.

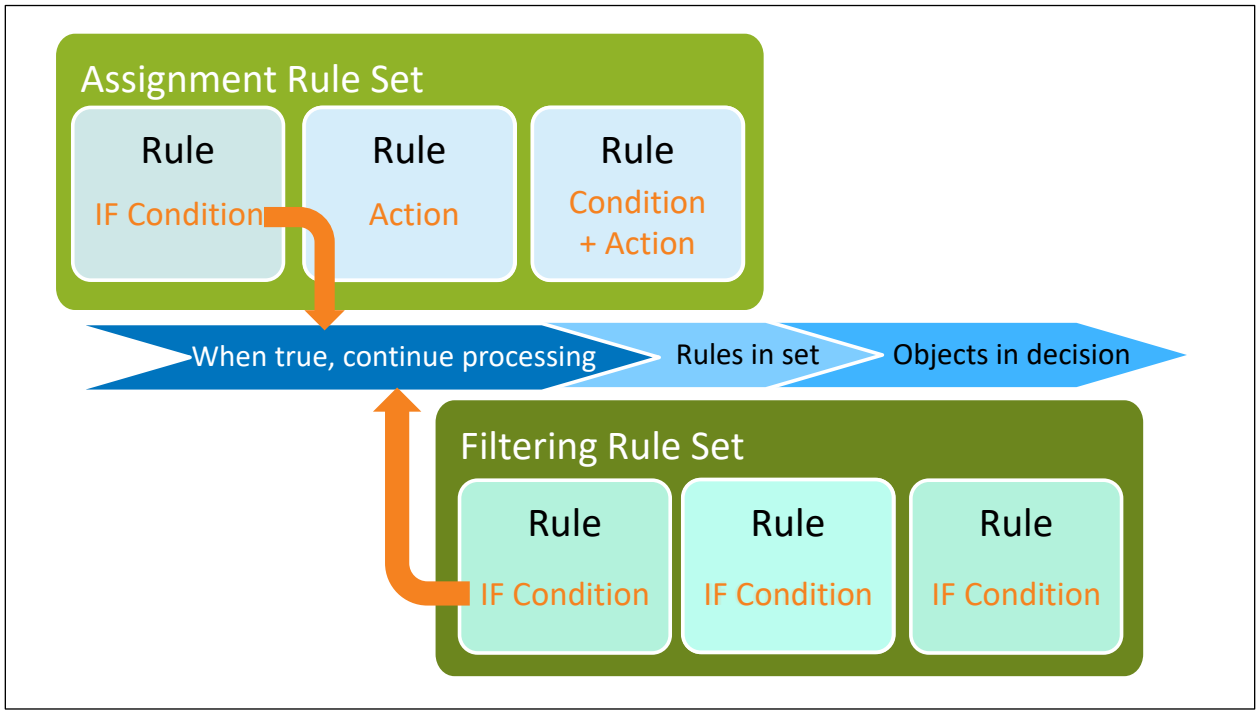


Assignment rule sets can include rules that contain condition statements and action statements, either together or on their own.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

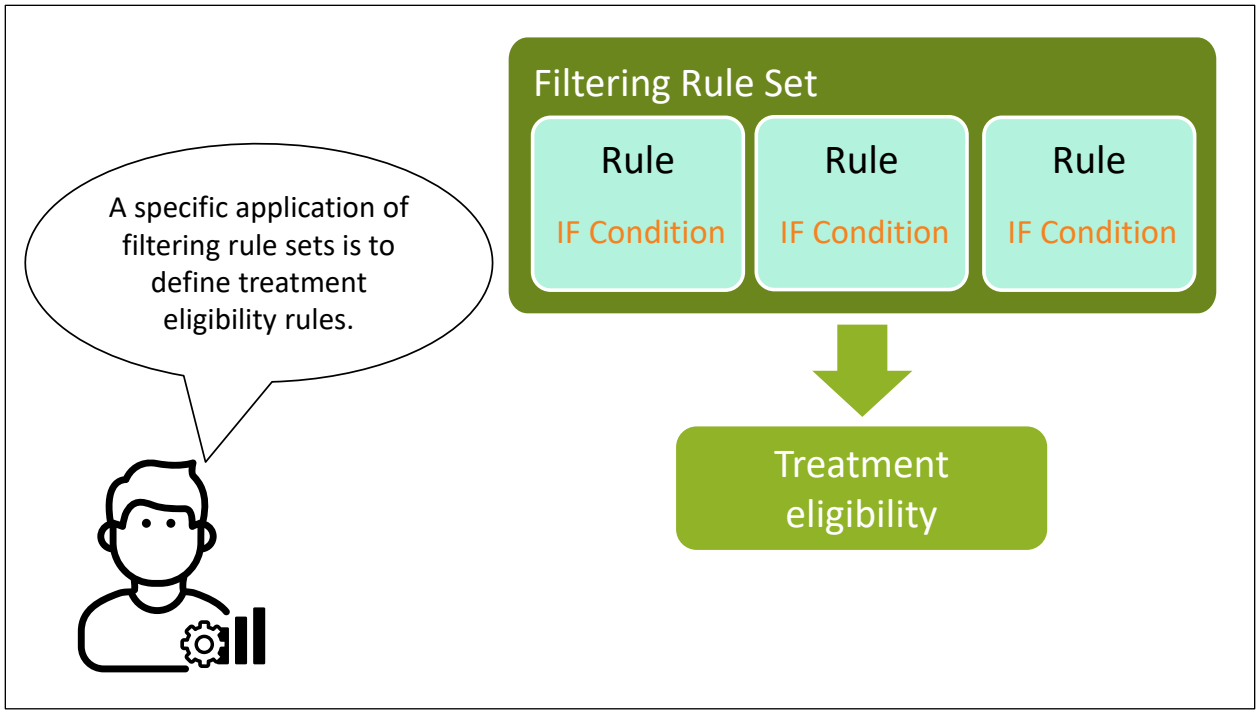


A filtering rule set can contain rules with IF condition statements only.

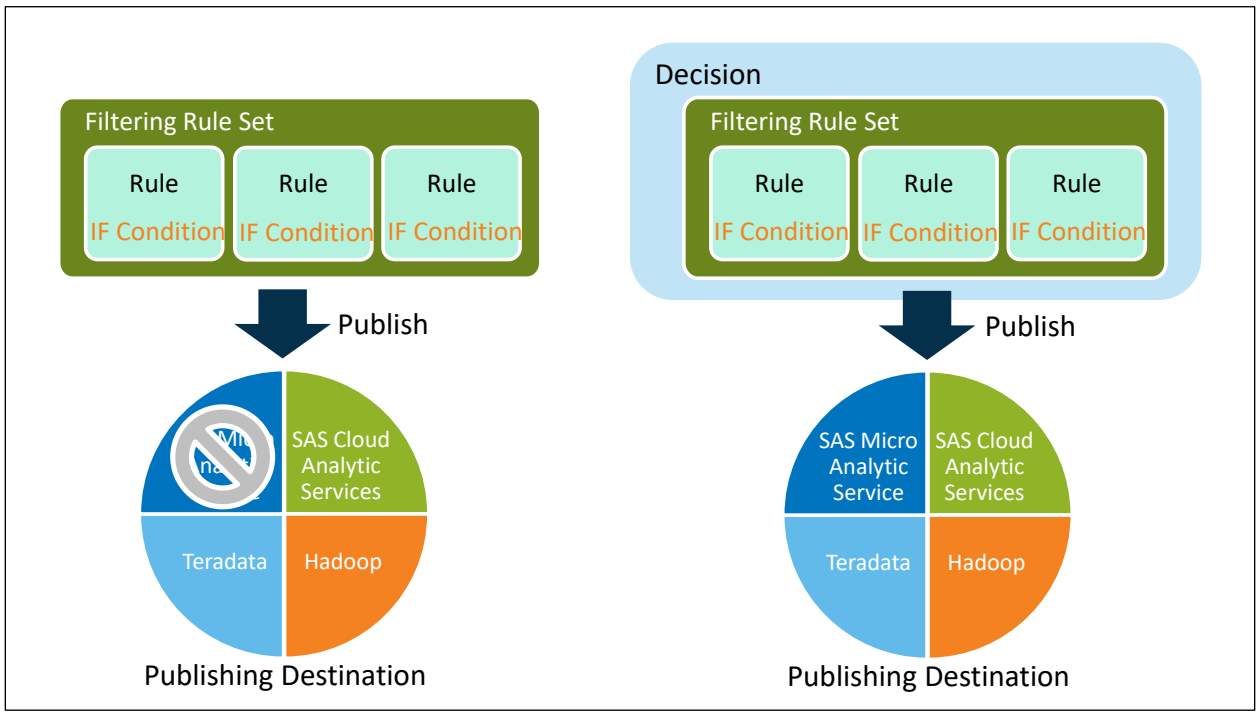


When a rule includes an IF condition only, it controls whether processing continues for the record. When the condition is true, processing of further rules in the rule set and objects in the decision continues. When false, processing stops.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

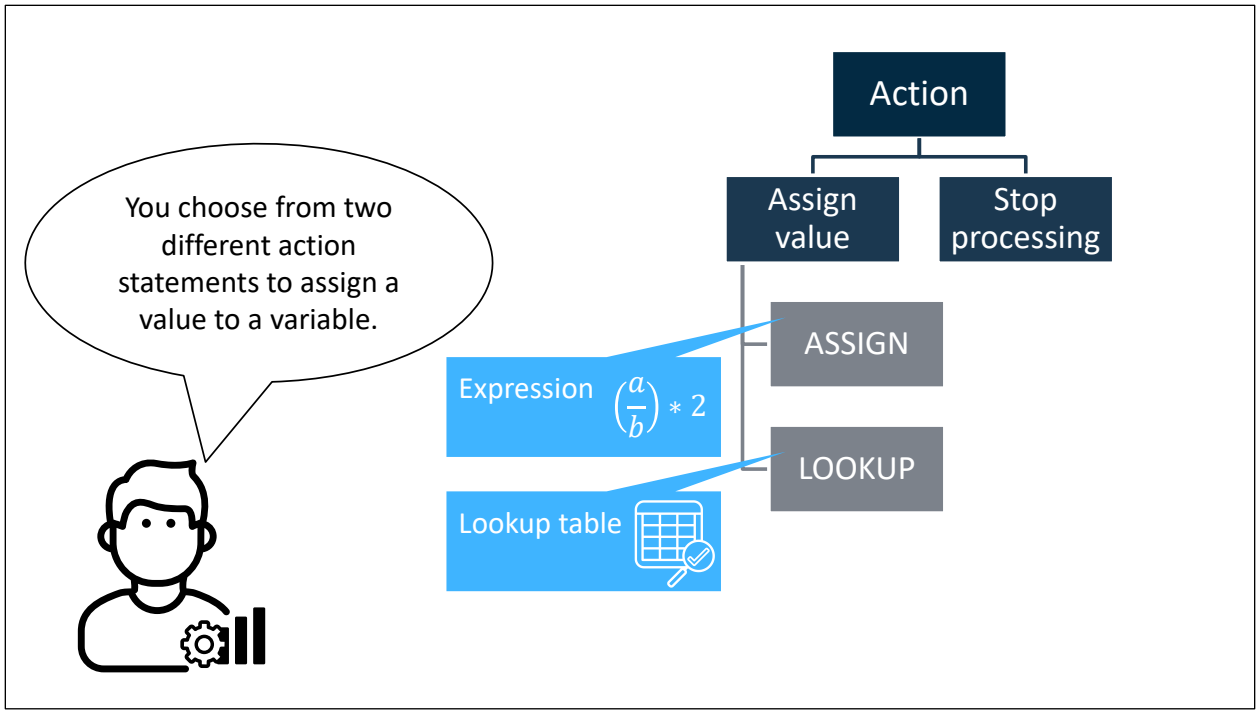


A specific application of filtering rule sets is to define eligibility rules for a treatment. Assignment rule sets cannot be used for this purpose.

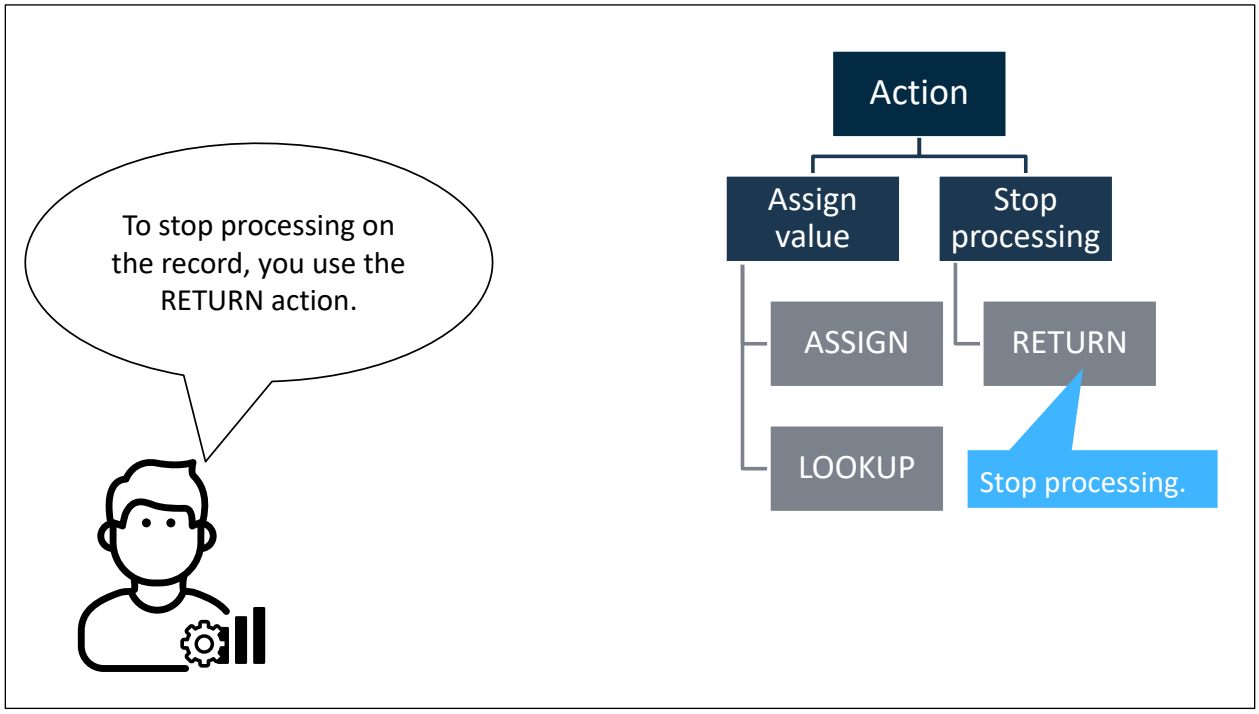


Recall that you can publish both decisions and rule sets. You cannot publish filtering rule sets to SAS Micro Analytic Service destinations. However, you can publish a decision that includes a filtering rules set to this destination.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

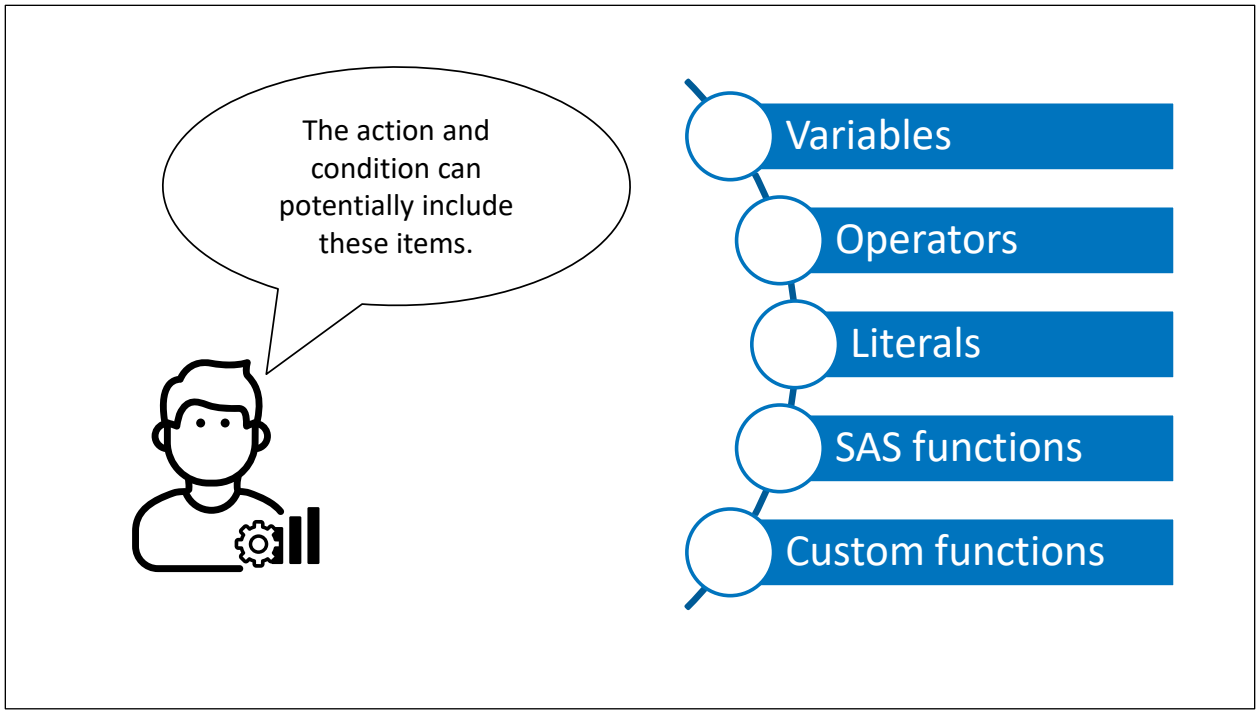


As stated earlier, there are two types of actions. You choose from two different action statements to assign a value to a variable: ASSIGN and LOOKUP. You use ASSIGN to assign a value using an expression, and LOOKUP to assign a value using a lookup table.

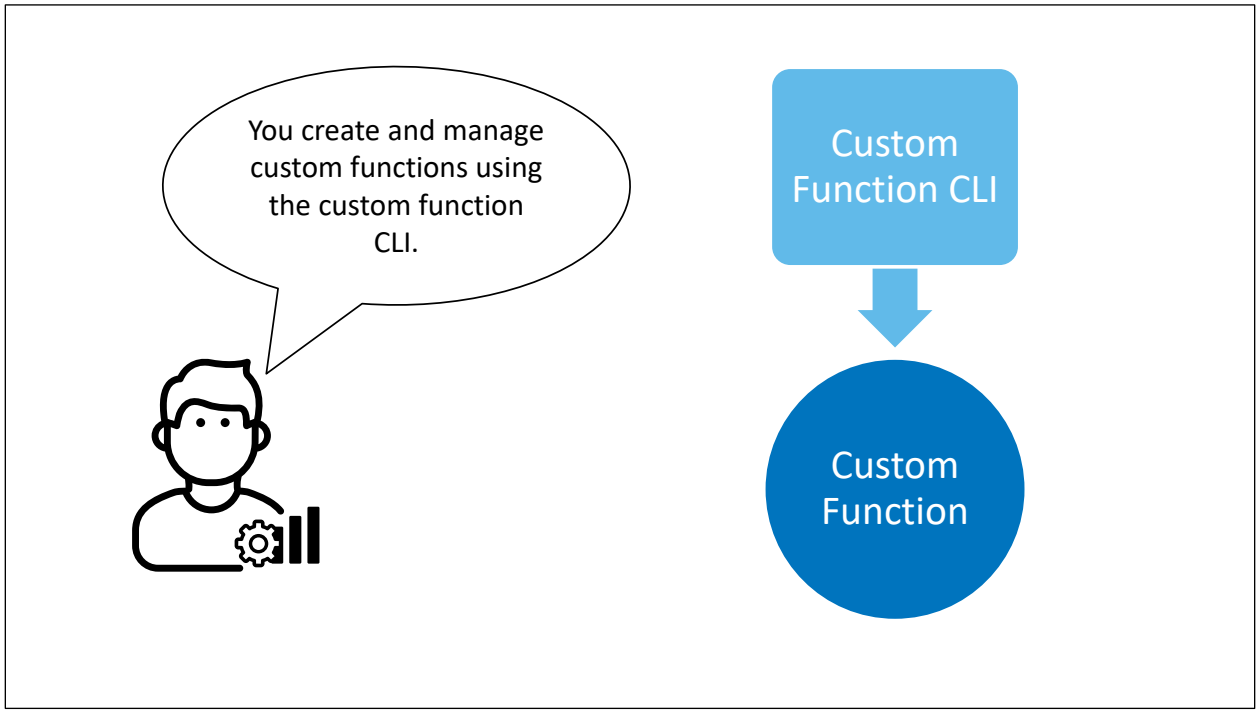


To stop processing on the record, you use the RETURN action.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

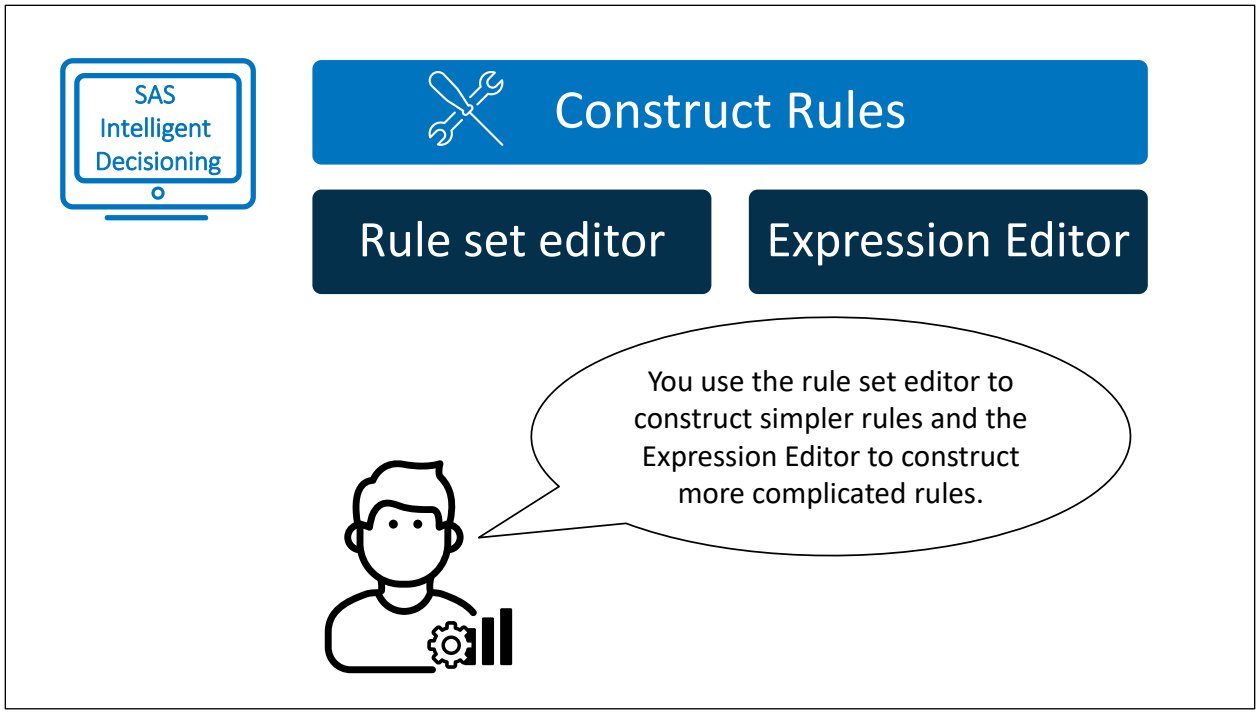


Depending on the type of rule, the action and the condition can potentially include variables, operators, literals or constants, and functions. Functions can be SAS functions that are provided with SAS Intelligent Decisioning or custom functions that you create.

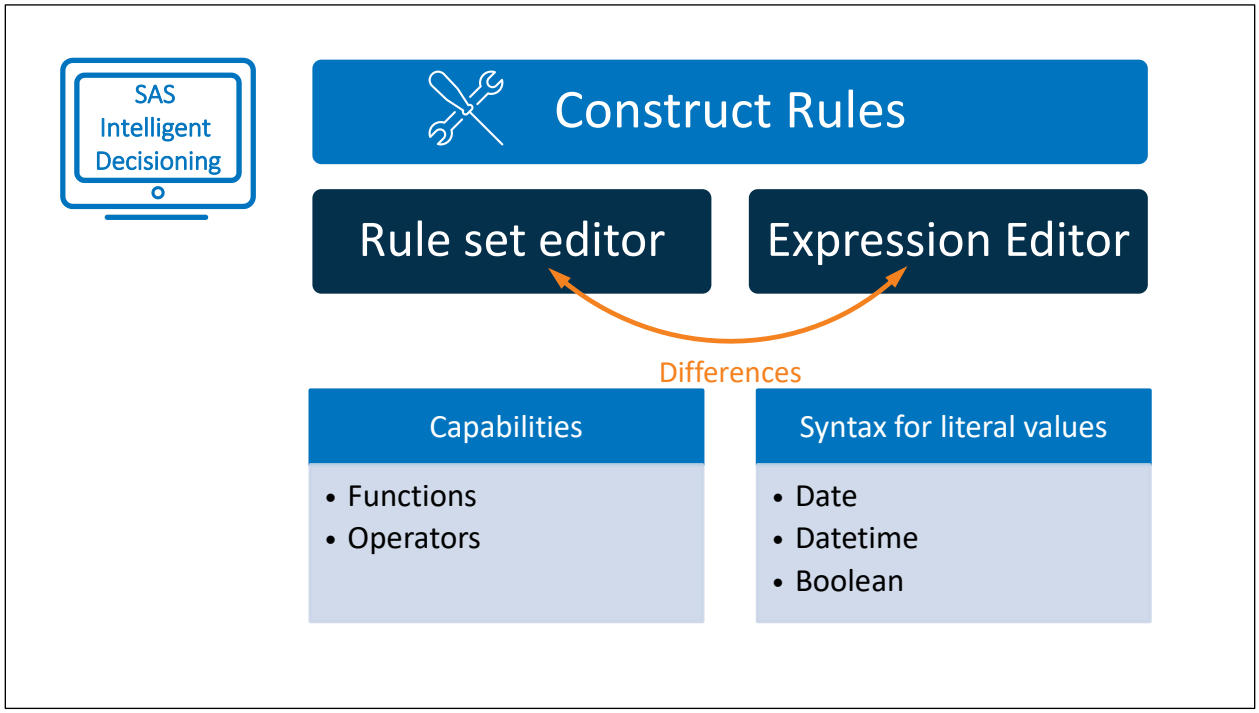


You can create custom functions to perform actions that are not available with the standard functions provided by SAS. You create and manage custom functions using the custom function CLI. For more information, refer to *SAS Intelligent Decisioning: Command-Line Interfaces*.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

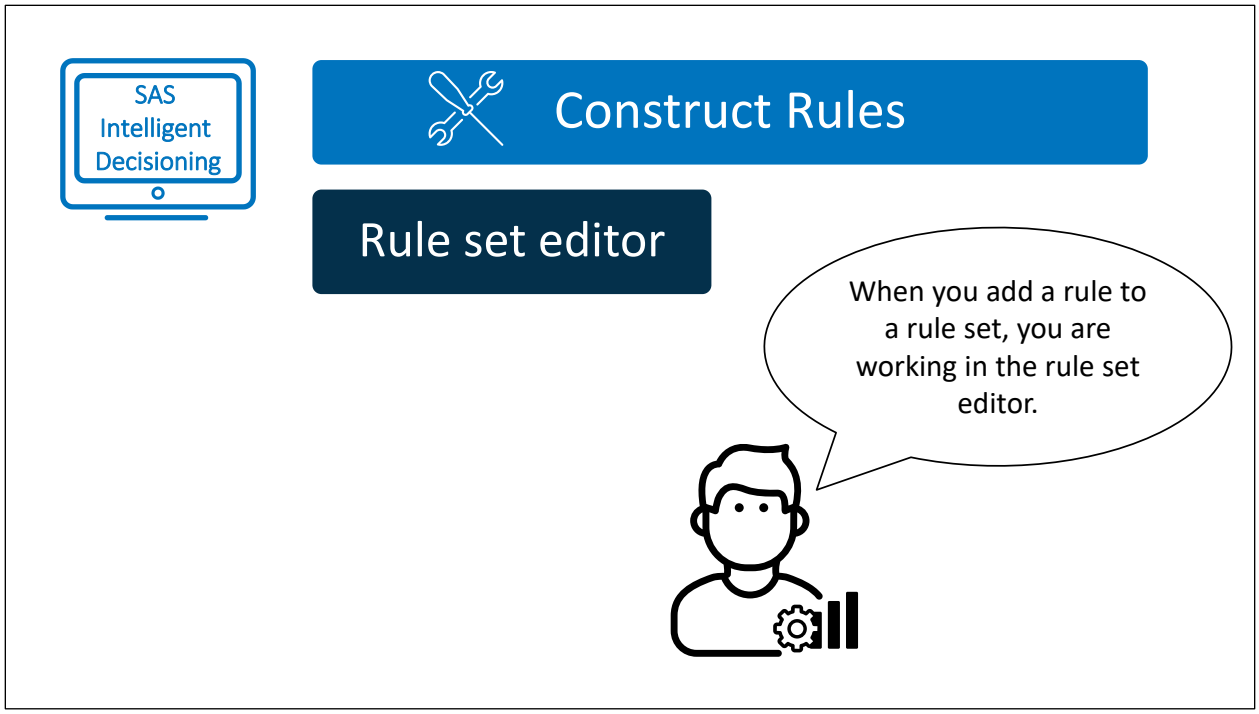


In the Intelligent Decisioning interface, there are two views that you can use to construct rules: the rule set editor and the Expression Editor. In general, you use the rule set editor to construct simpler rules and the Expression Editor to construct more complicated rules.

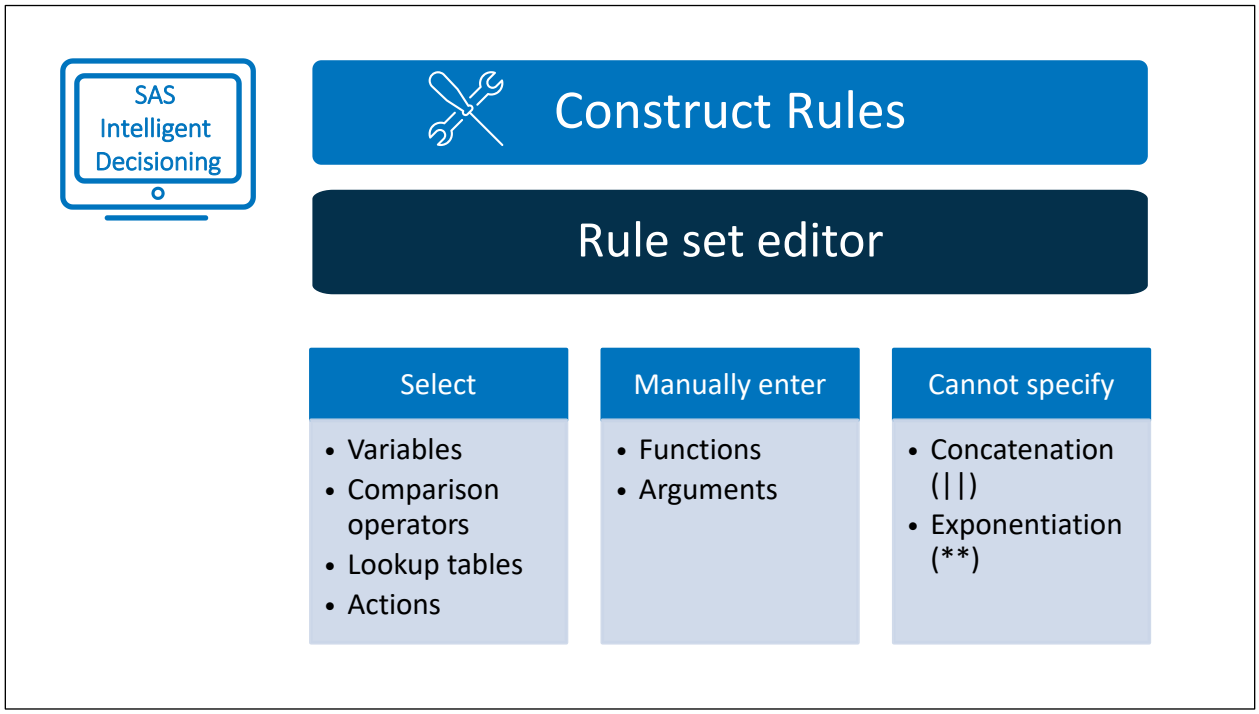


There are some important differences between these two views. You might need to choose one or the other depending on the specific capabilities that you need to use. You can use a few certain functions and operators in only one of the two. In addition, the way that you enter literal values for date, datetime, and Boolean variables is different between the two.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

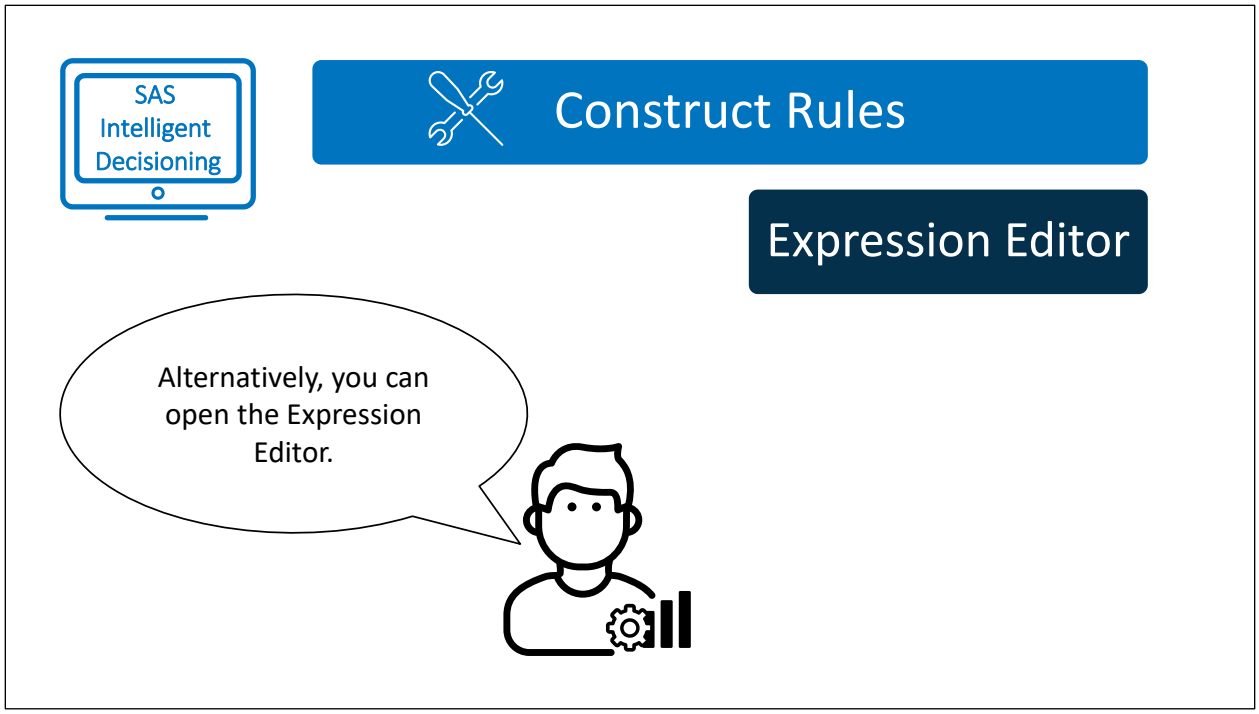


When you add a rule to a rule set, you are working in the rule set editor.

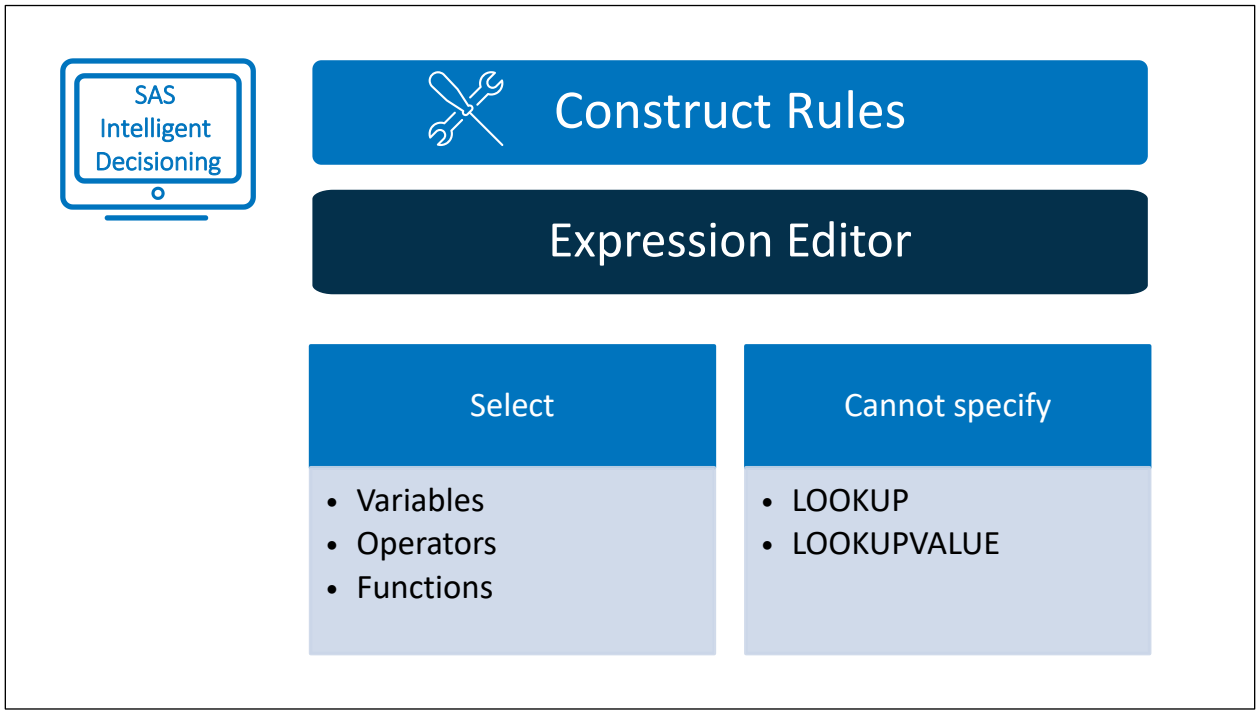


In the rule set editor, you can select items such as variables, comparison operators, lookup tables, and actions from lists. If you want to use functions, you must manually enter them along with their arguments. You cannot use this view to specify concatenation, or exponentiation operators.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

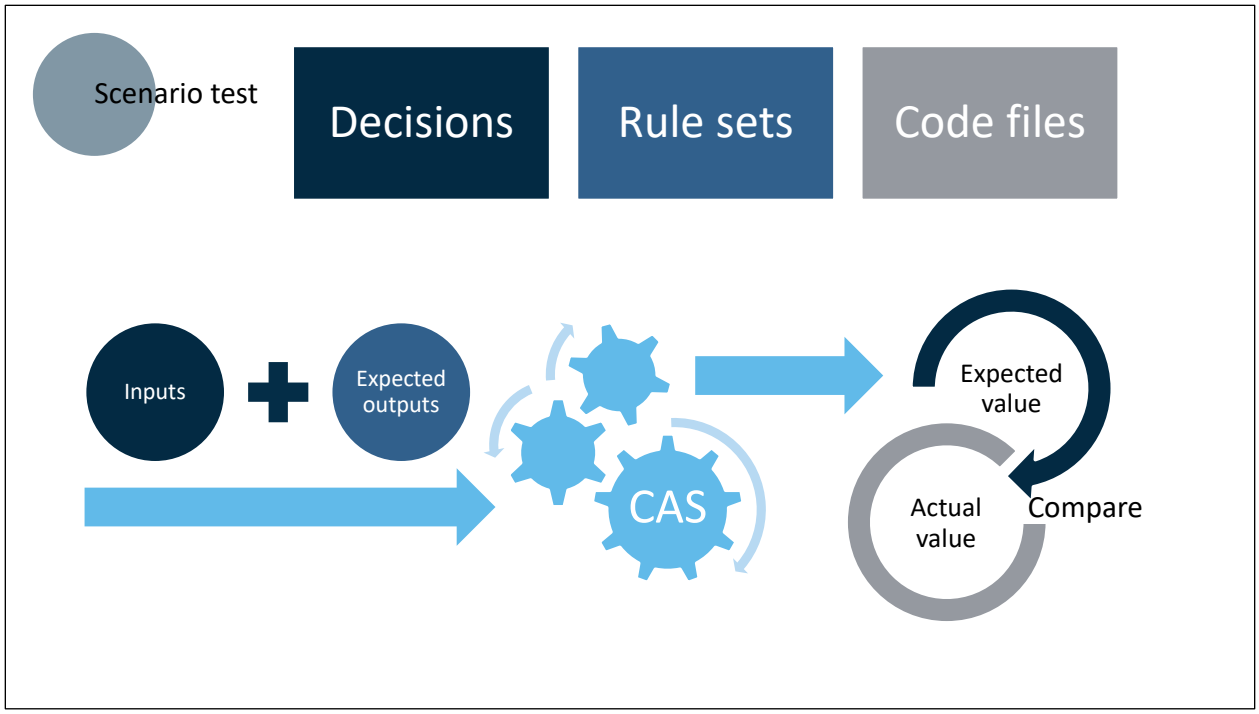


Alternatively, you can open the Expression Editor.

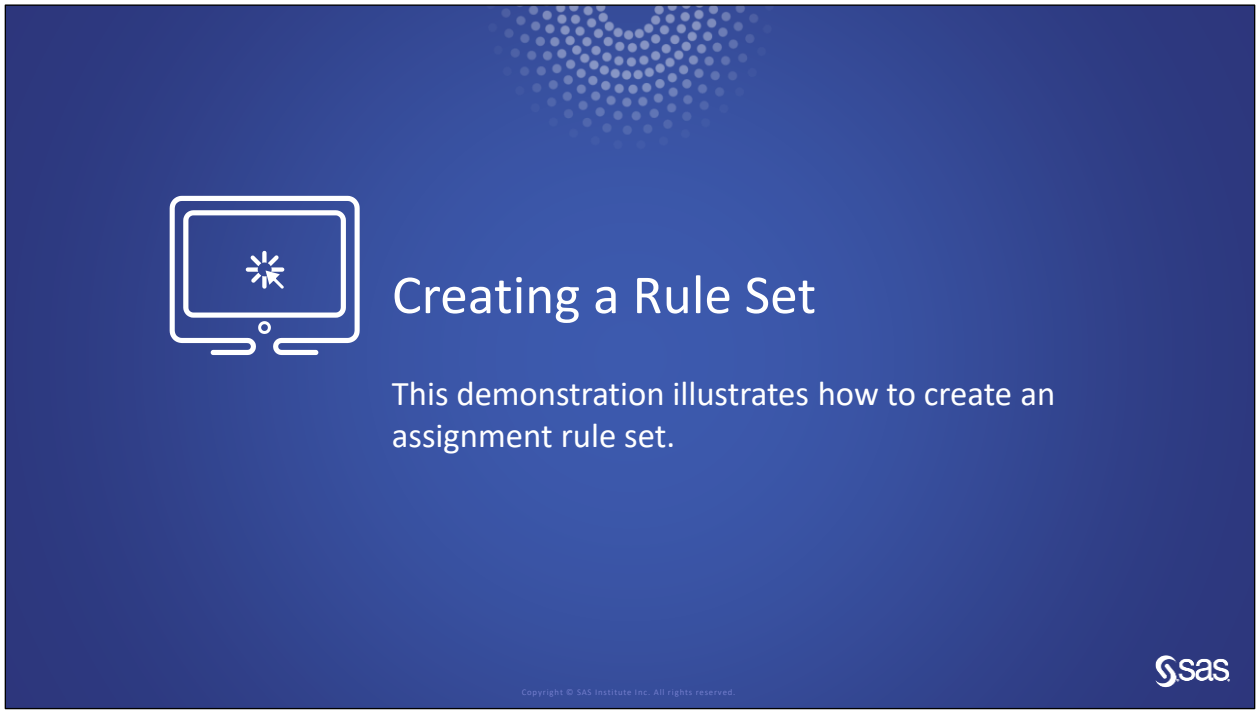


In the Expression Editor, you select variables, operators, and functions in a point-and-click interface. However, you cannot specify lookup table functions.

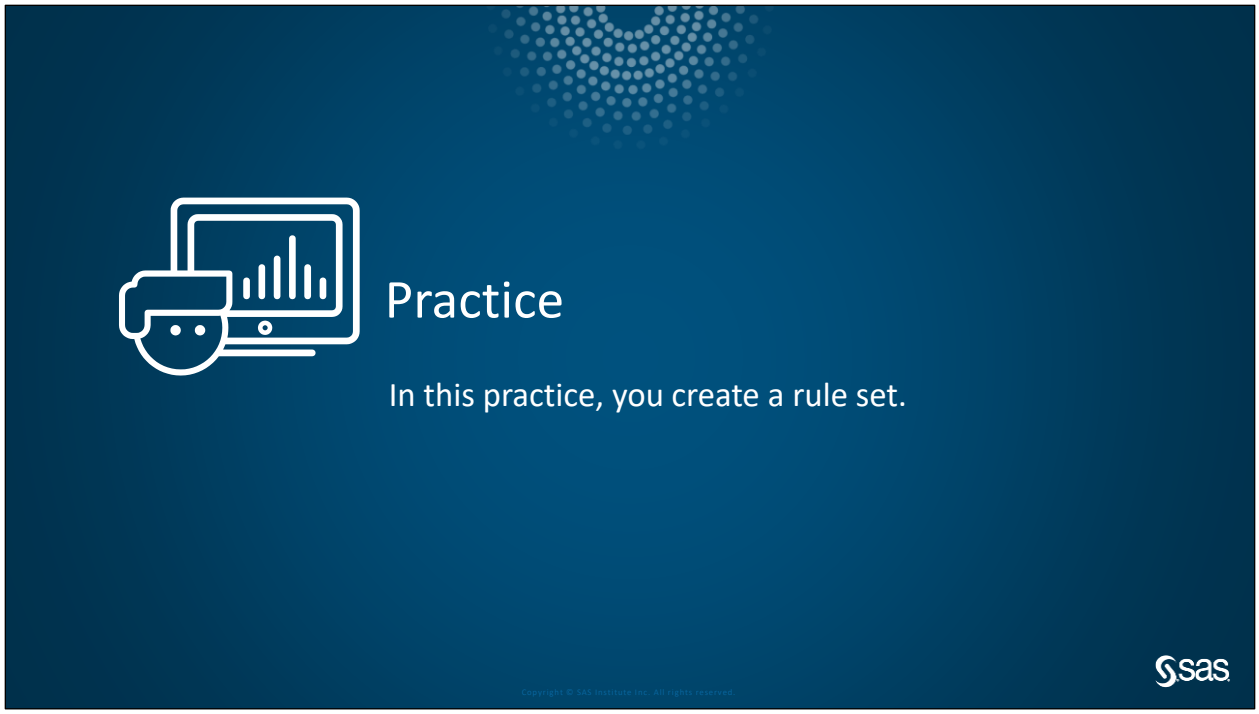
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

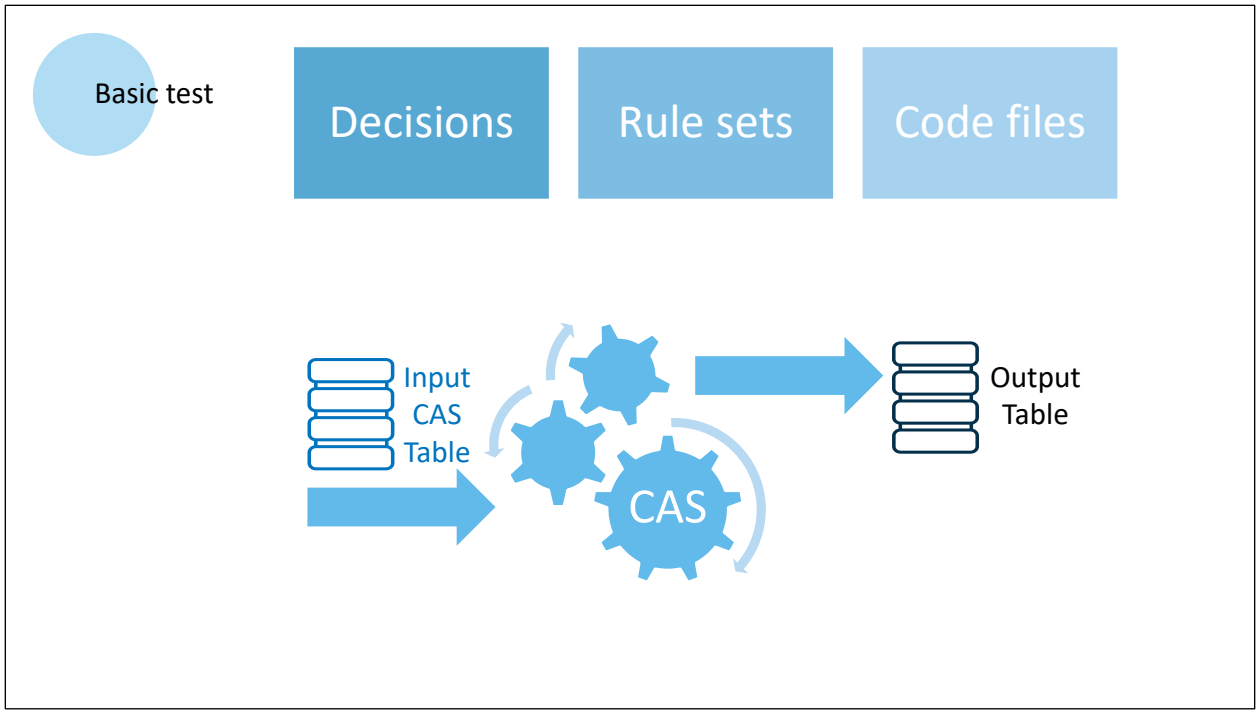


You can perform scenario testing for decisions, rule sets, and code files. When you perform a scenario test, you specify input variable values. You can also specify the corresponding output variable values that you expect the test to generate. A scenario test identifies differences between the expected and actual output values.



Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





You can perform basic testing for decisions, rule sets, and code files. When you perform basic testing, the object is executed in CAS using an input table that you specify. The test generates and displays a table containing output values.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Adding a Rule Set to a Decision

This demonstration illustrates how to add a rule set to a decision.

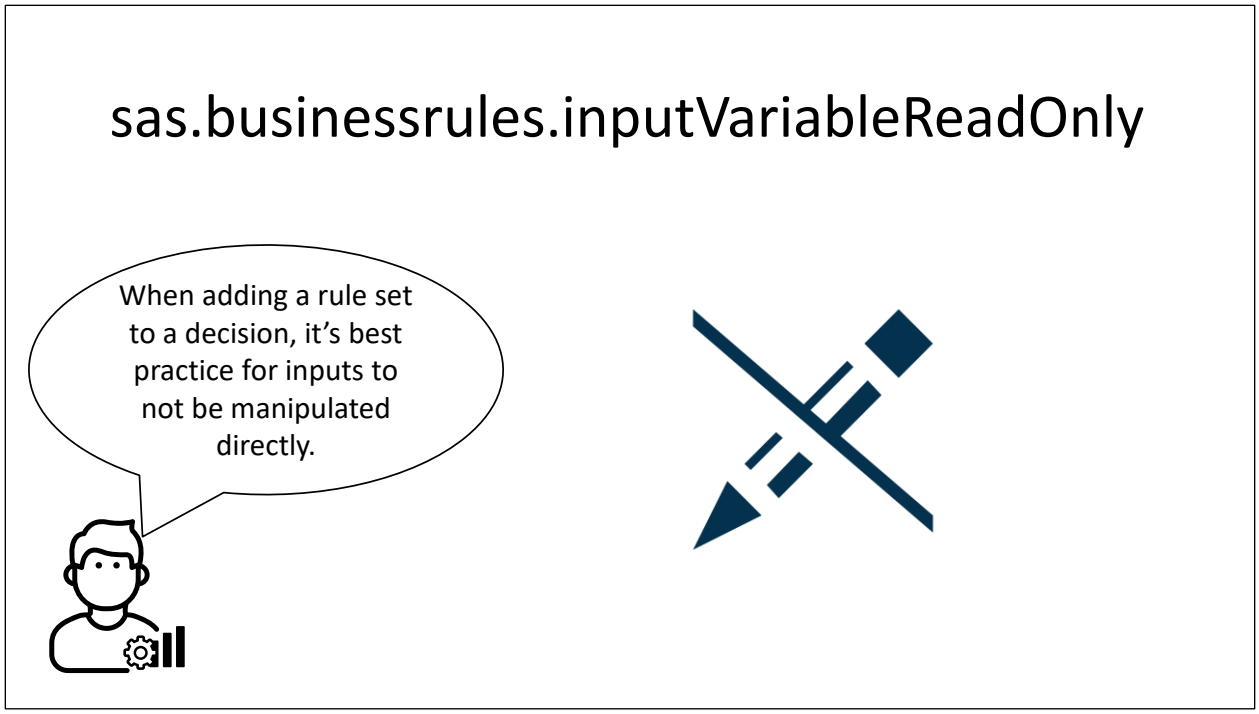
Copyright © SAS Institute Inc. All rights reserved.

Practice

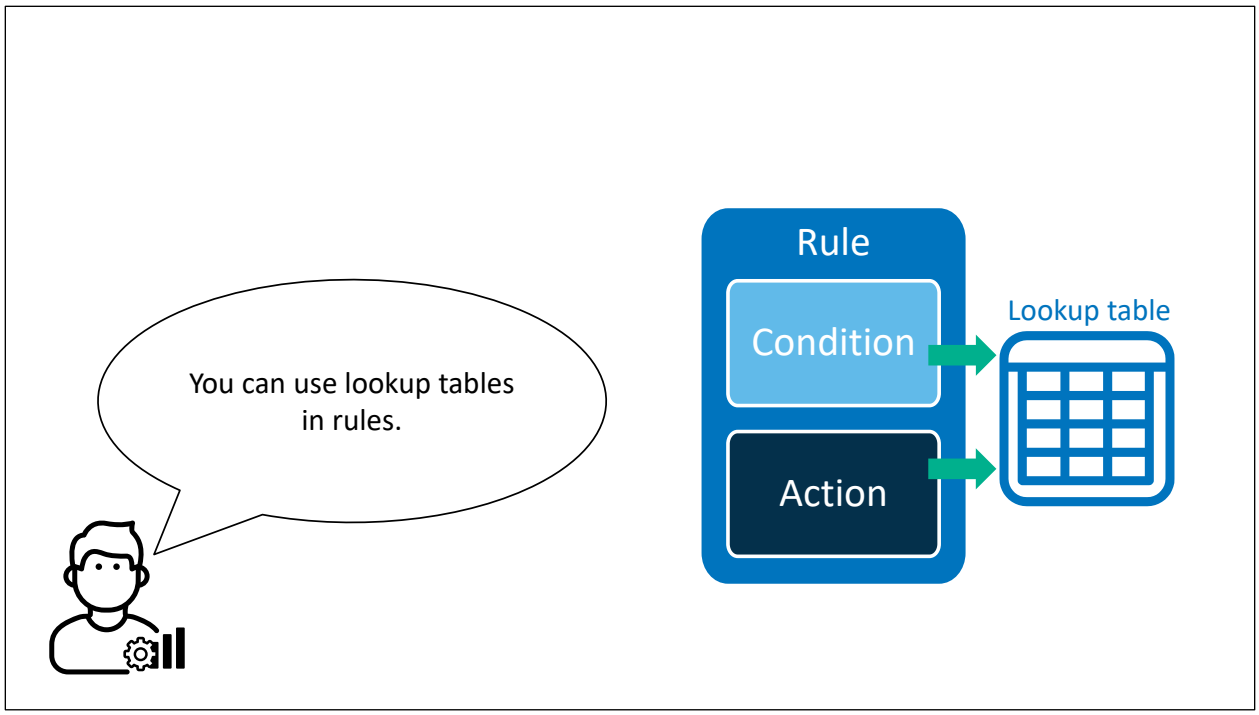
In this practice, you create a decision and add a rule set.

Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

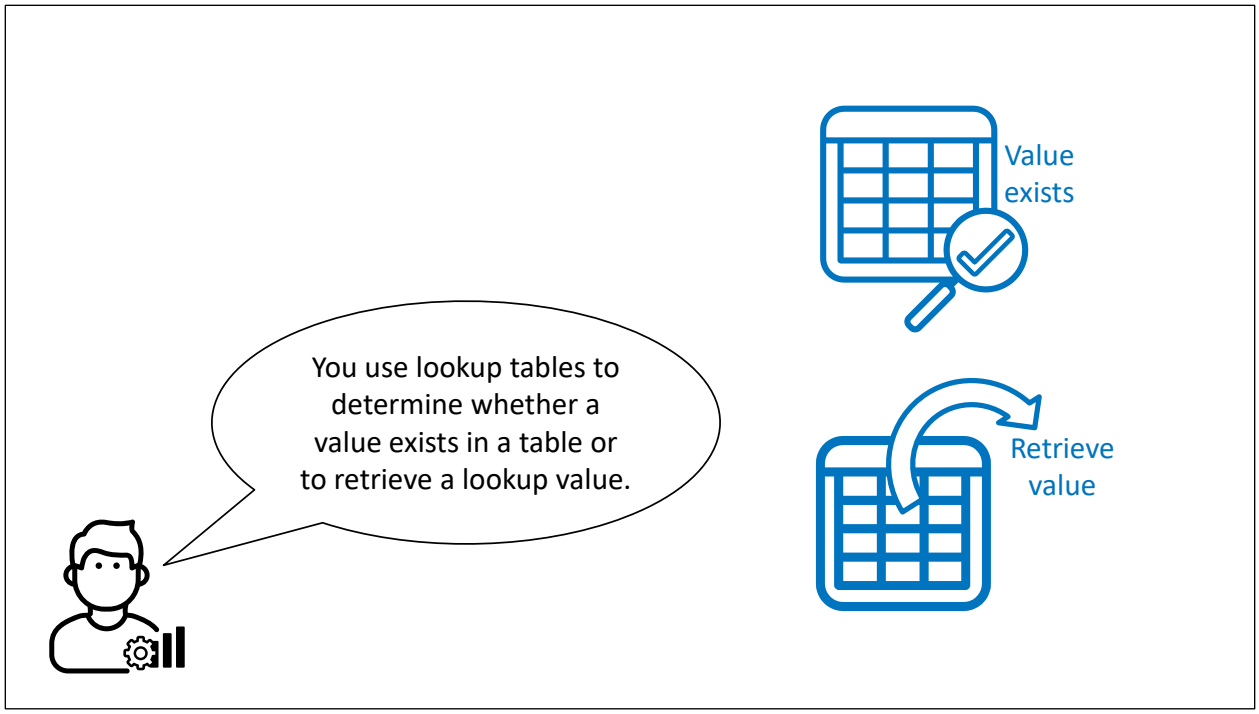


When adding a rule set to a decision, it's best practice for inputs to not be manipulated directly. By default, character variables from a rule set are passed into the decision by reference. You can pass character variables in by value by modifying the configuration option sas.businessrules.inputVariableReadOnly in the Environment Manager, which prevents modification of rule set input-only variables when the rule set is called from a decision.

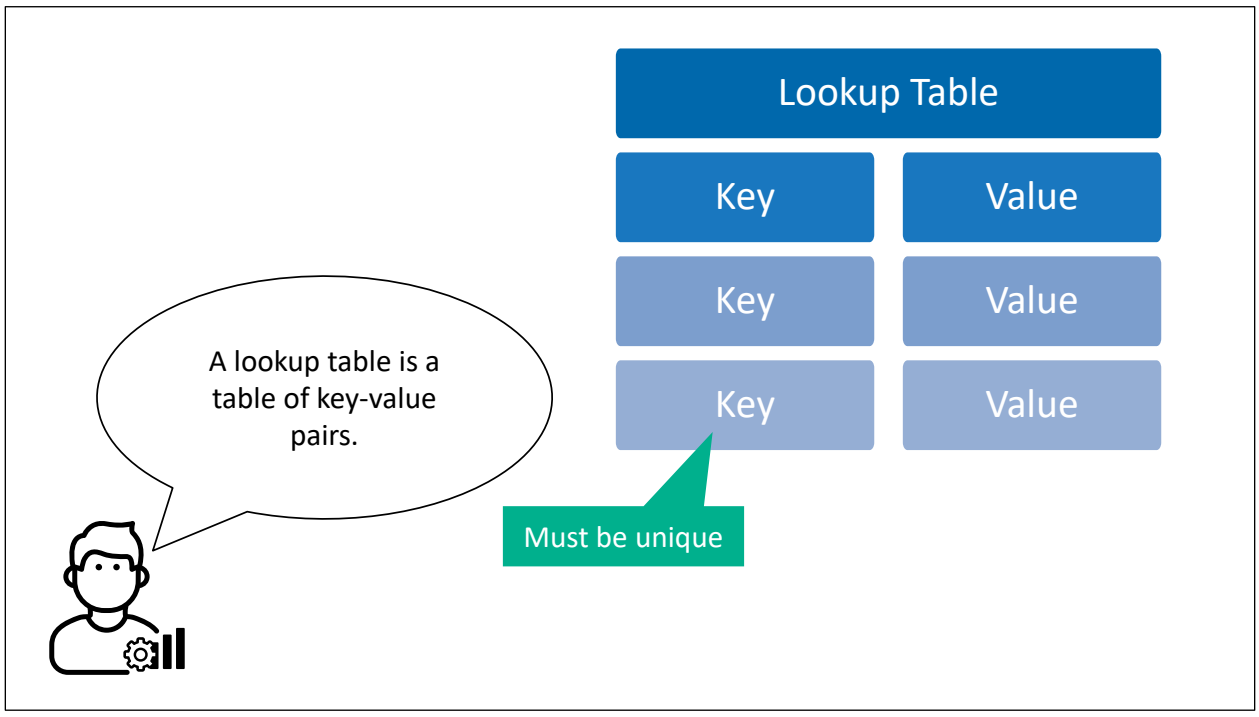


You can use lookup tables in rules.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

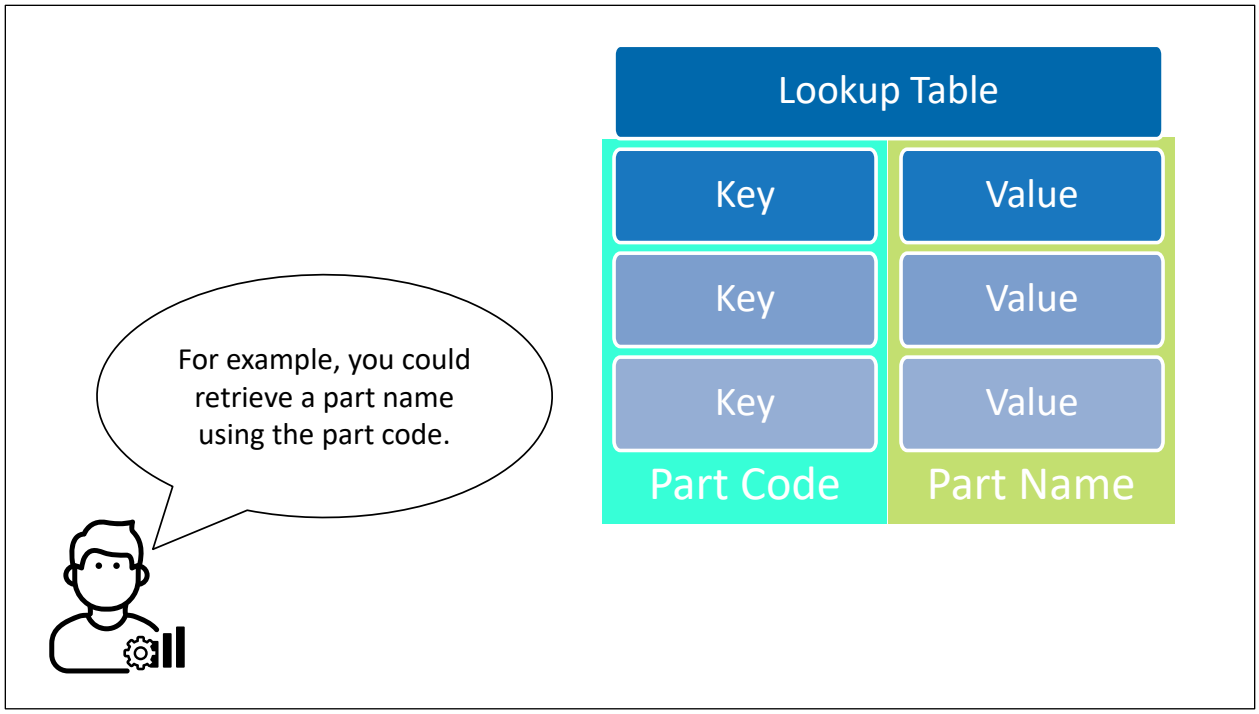


You can use lookup tables to determine whether a value exists in a table or to retrieve a lookup value.

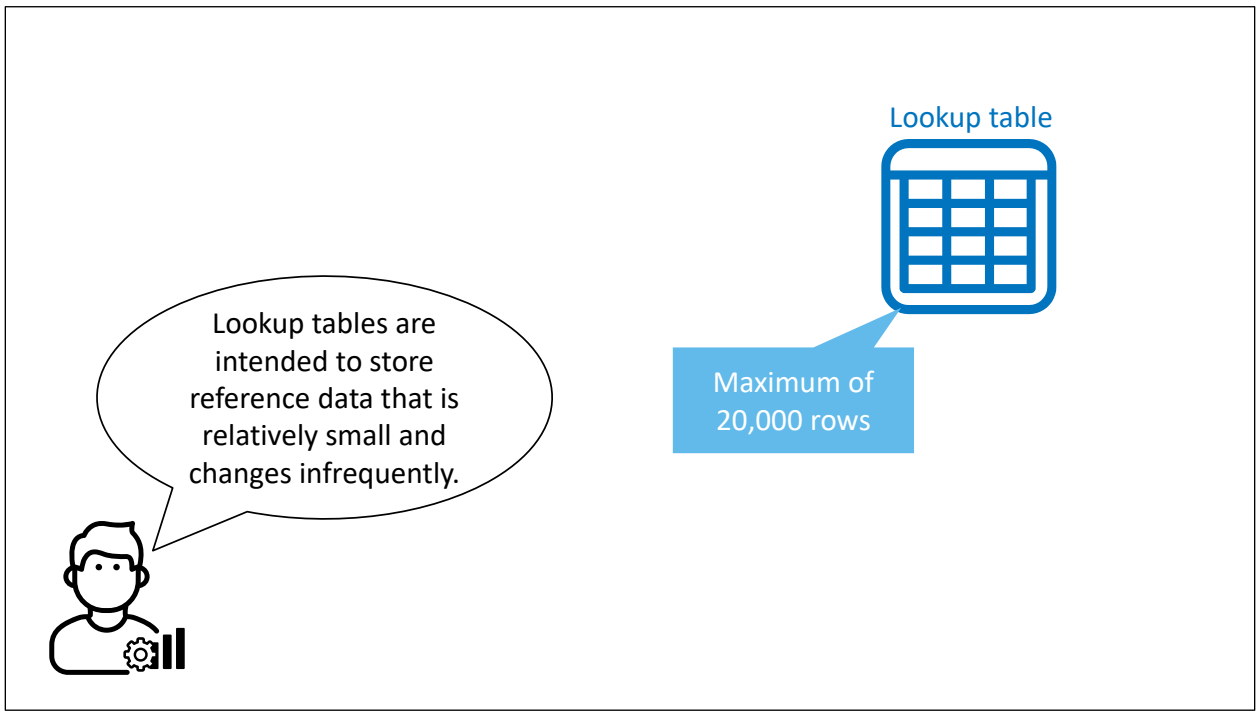


A lookup table is a table of key-value pairs. The key must be unique within the lookup table.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

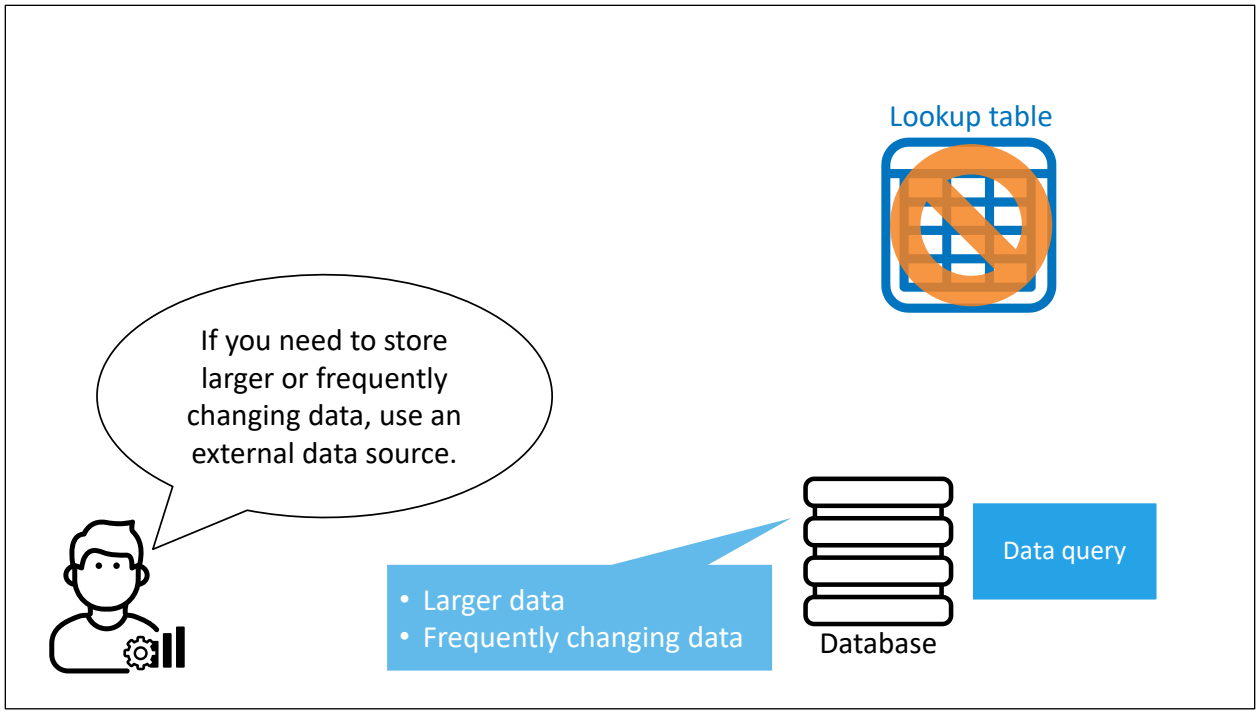


For example, you could have a lookup table where the key corresponds to a part code and the value is the part name. You could then retrieve the part name in a rule set using the part code.

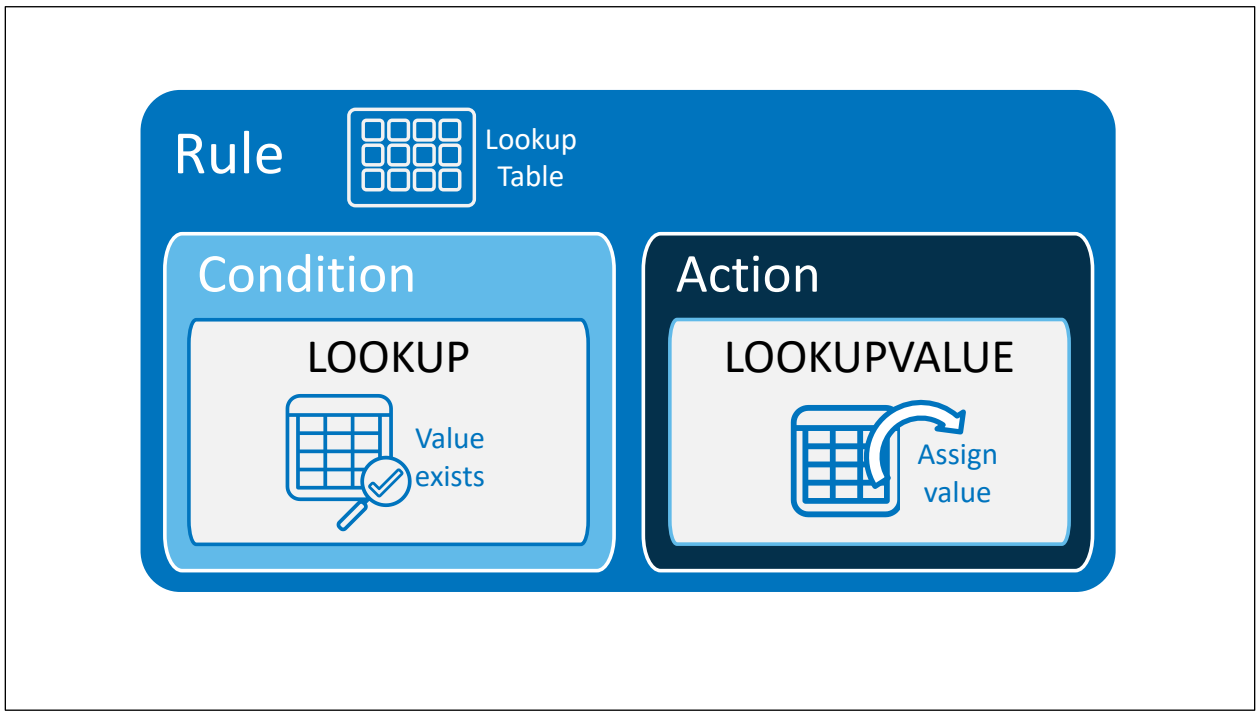


Lookup tables are intended to store reference data that is relatively small and changes infrequently. A maximum of 20,000 rows is recommended.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

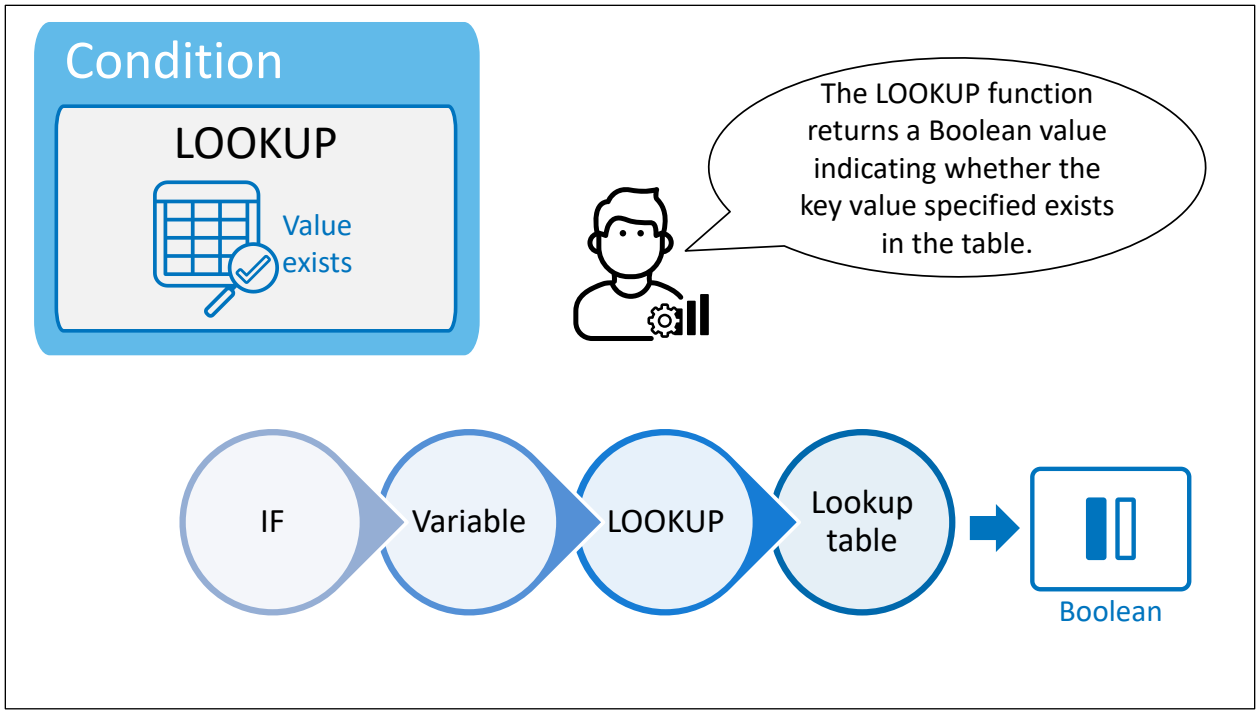


If you need to store more data or have reference data that changes frequency, you should populate the data in an external data source and access it in a decision using a data query node.

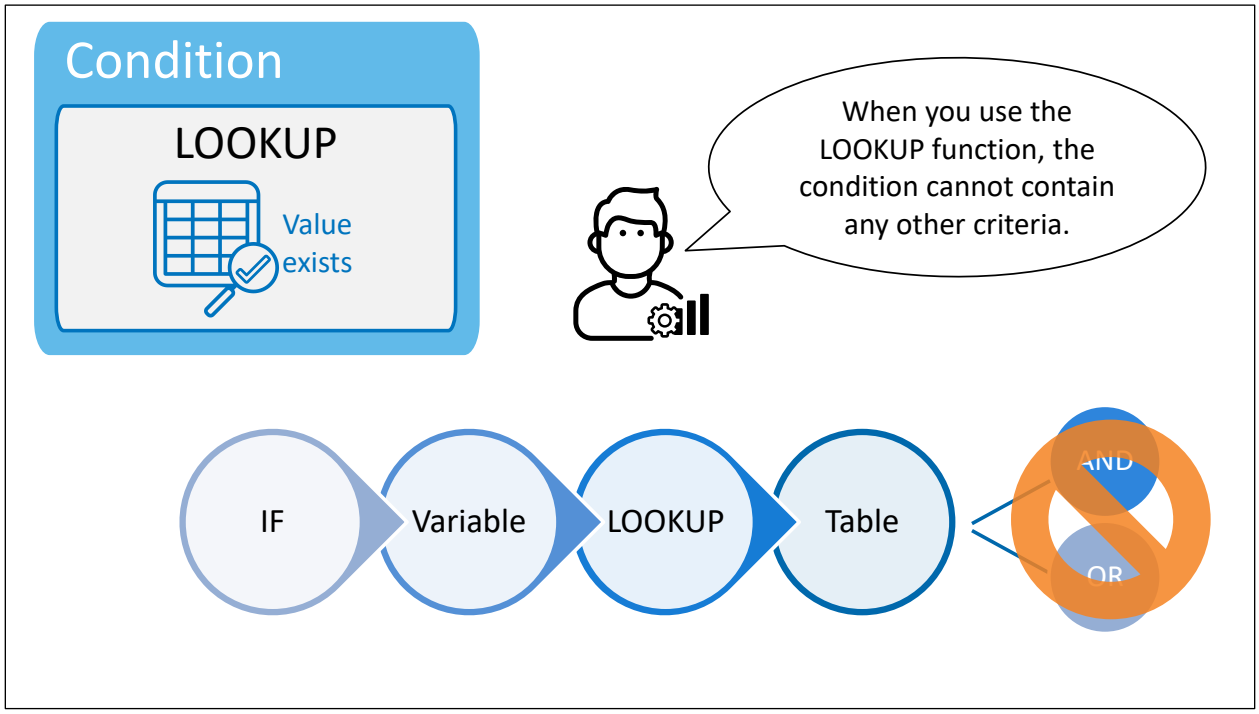


You can reference a lookup table in a rule using two functions. Recall that rules can potentially include condition and action statements. You can use the LOOKUP function in a condition statement to test whether a key value exists in a lookup table. You can use the LOOKUPVALUE function in an action statement to assign a value to a variable using a lookup table.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

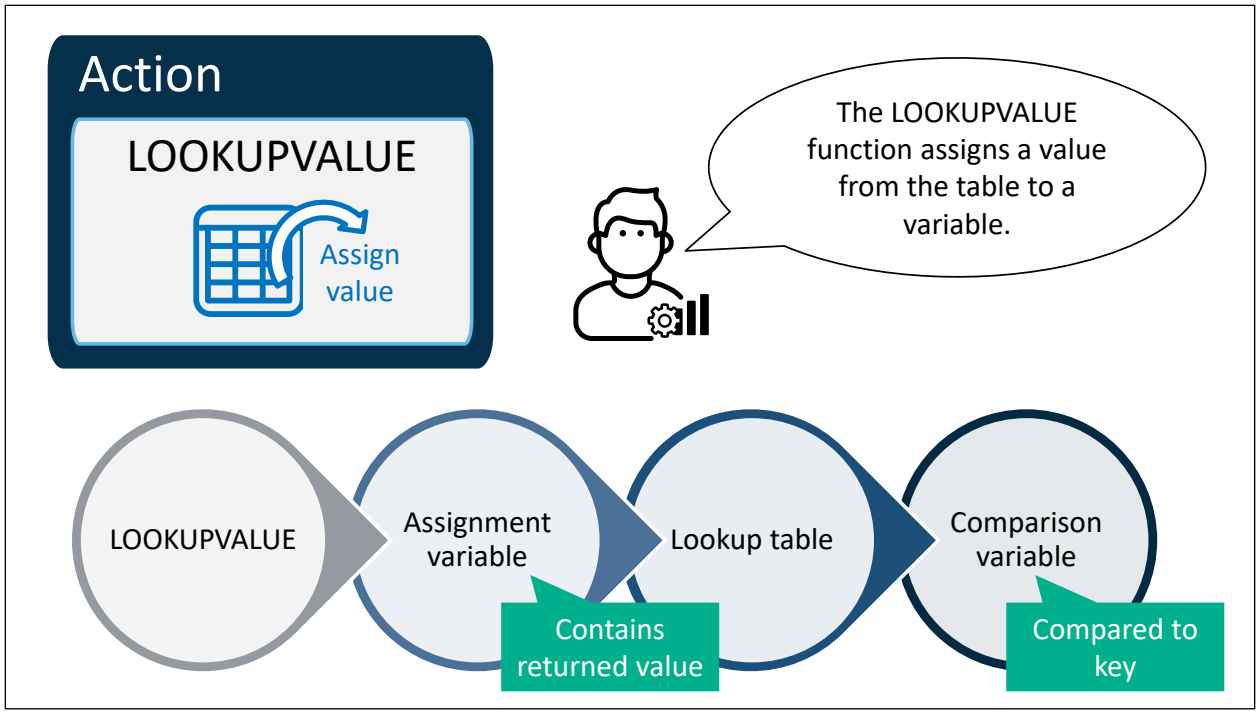


You use the LOOKUP function in a condition. You select a variable to compare to the key in a lookup table. You select the LOOKUP function as the operator for the condition, and then select the lookup table. The LOOKUP function determines whether the value specified exists as a key value in the lookup table and returns a Boolean value.

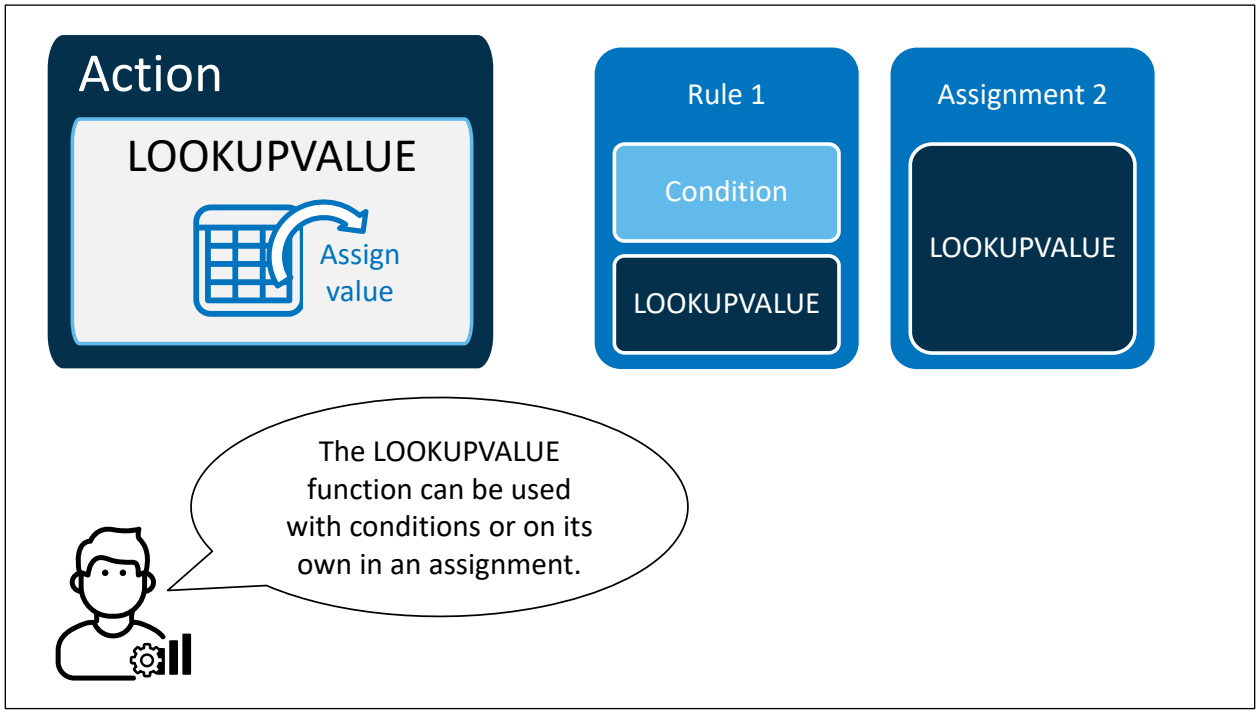


When you use the LOOKUP function in a condition, the condition cannot contain any other criteria.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

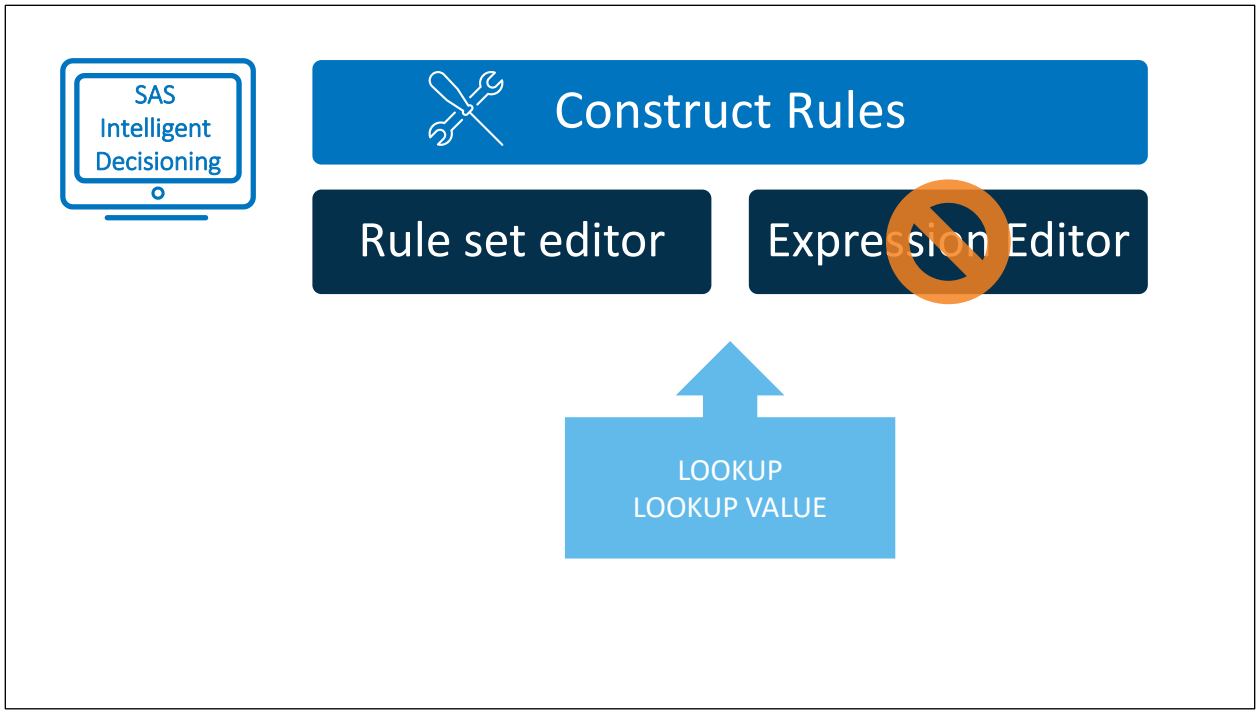


LOOKUPVALUE is one of the three actions that you can use in a rule. You use it to assign a value from a lookup table to a variable. You select LOOKUPVALUE as the action, select a variable to contain the value returned from the lookup table, specify the lookup table, and specify the variable that is compared to the key value in the lookup table.

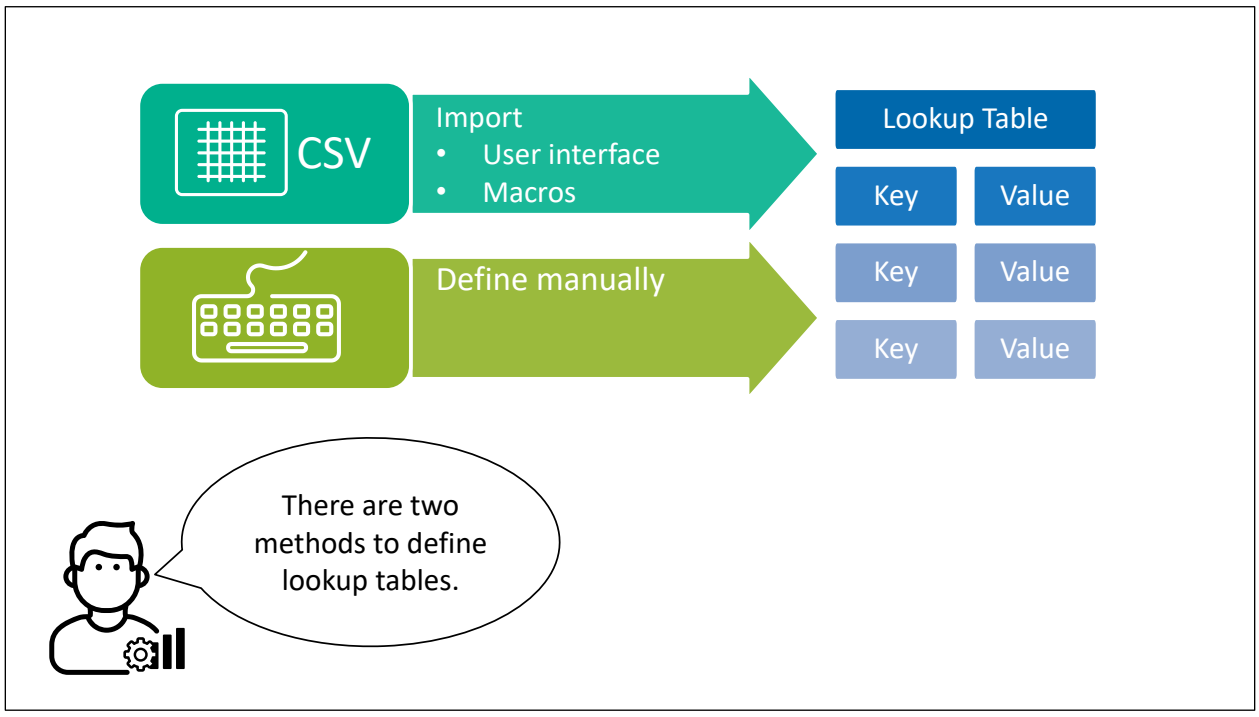


You can use the LOOKUPVALUE function in rules that include conditions as well as those that don't.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

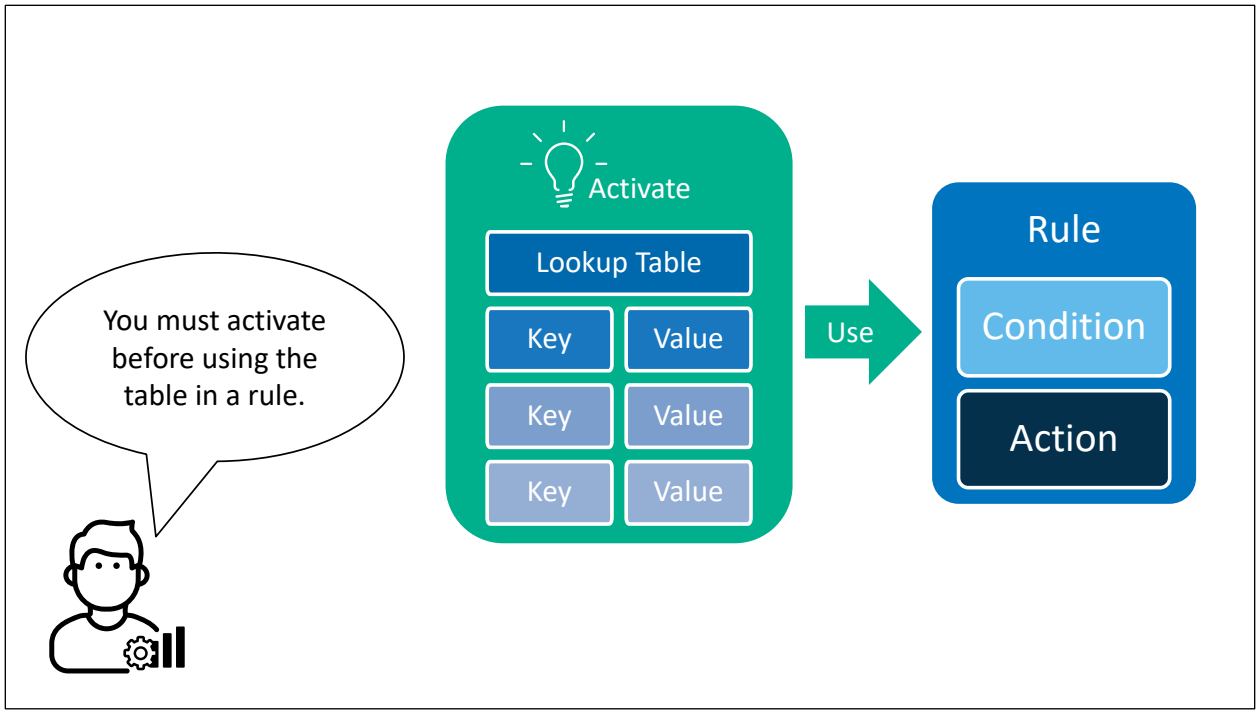


You cannot define rules using lookup table functions in the Expression Editor. You must use the rule set editor.

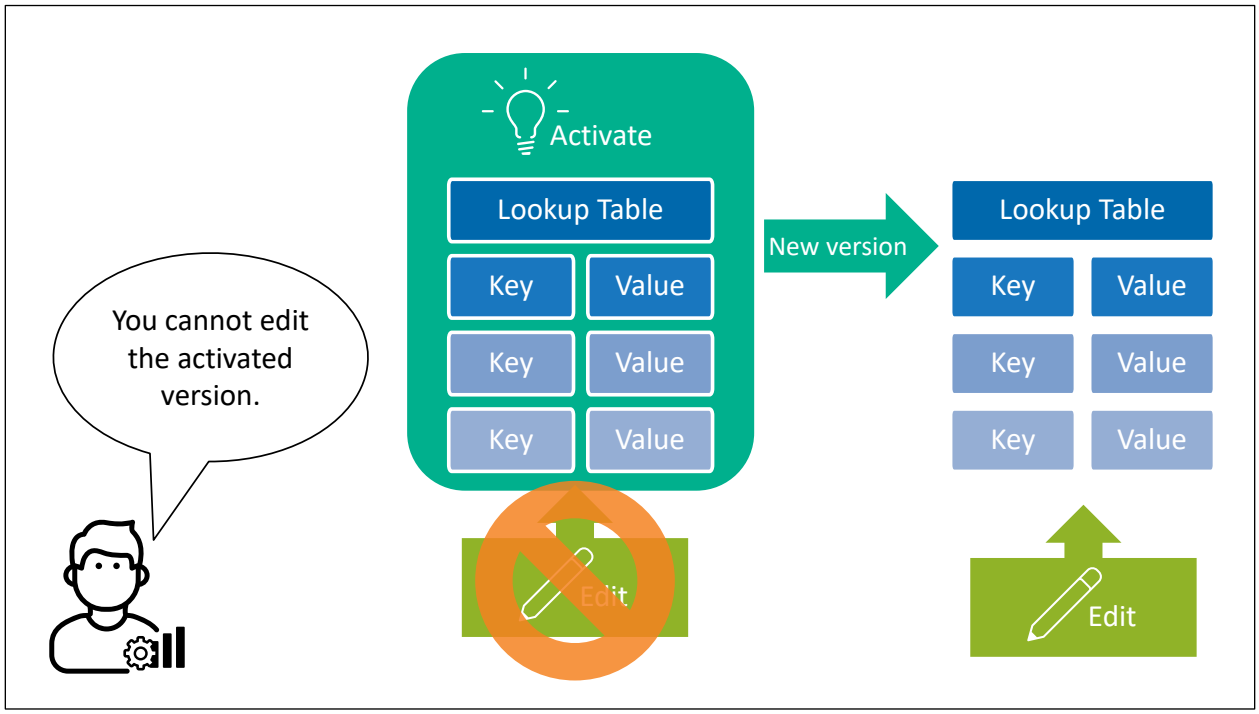


You can create a lookup table byimporting a CSV file or bydefining the key-value pairs manually. If you import from CSV, you can do so using the user interface or one of the SAS Intelligent Decisioning macros. The CSV file should contain the name-value pairs only without headers.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

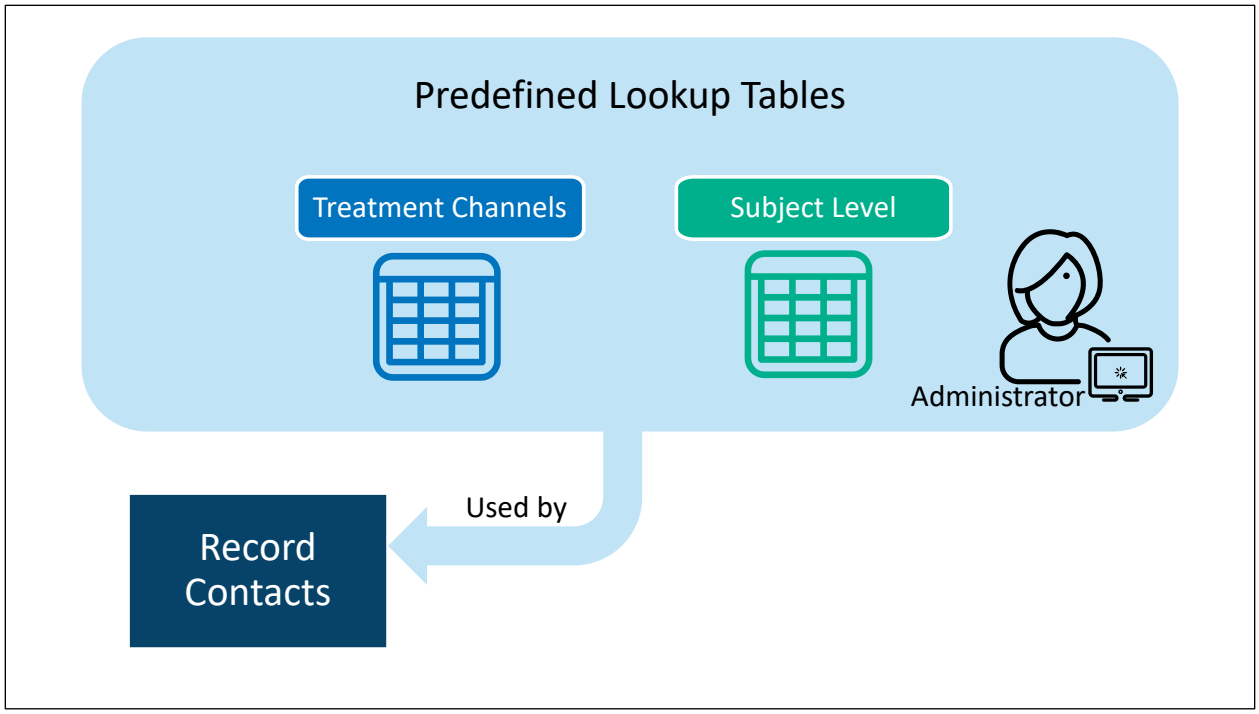


You must activate the lookup table in order to use it in a rule.

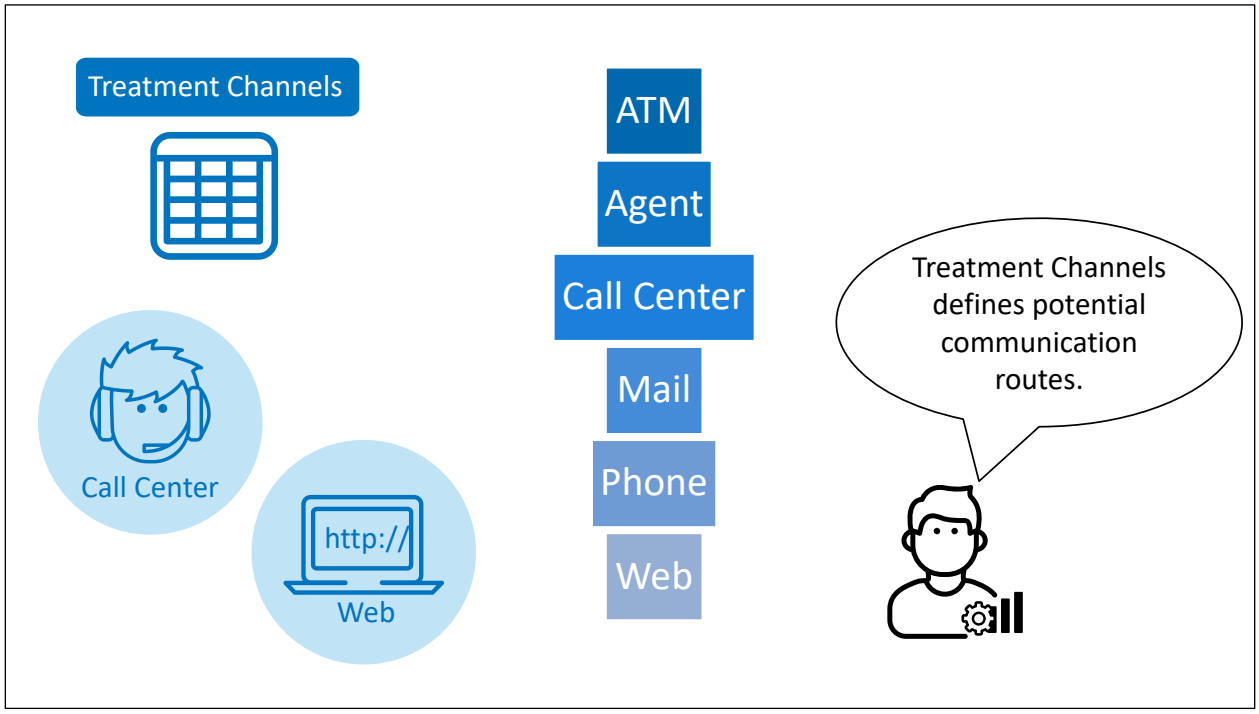


You cannot edit the activated version of a lookup table. However, you can create a new version that can be edited.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

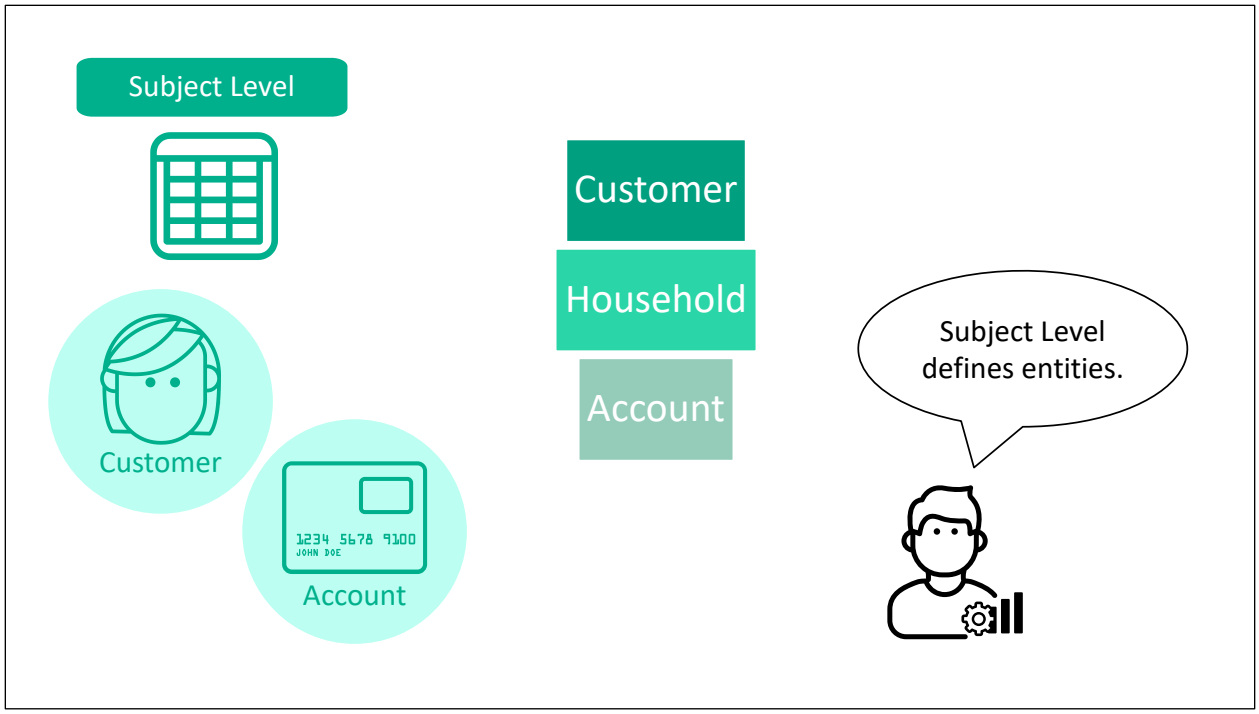


There are also two lookup tables that are predefined for Intelligent Decisioning: **Treatment Channels** and **Subject Level**. Both are used with the record contacts node, but you could also use them in a decision. They have default values but can be modified by an administrator.

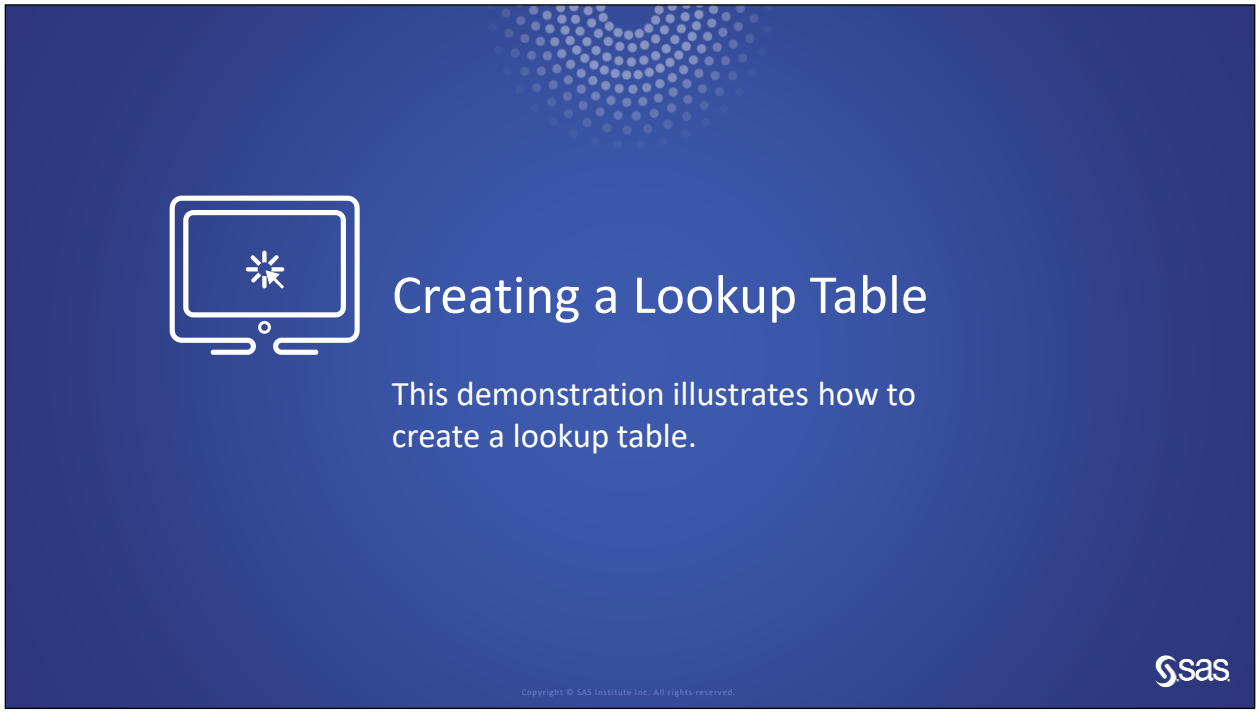


The Treatment Channels table defines channels, which are the potential communication routes for a decision, such as call center or web. The default key values in the table are shown here.

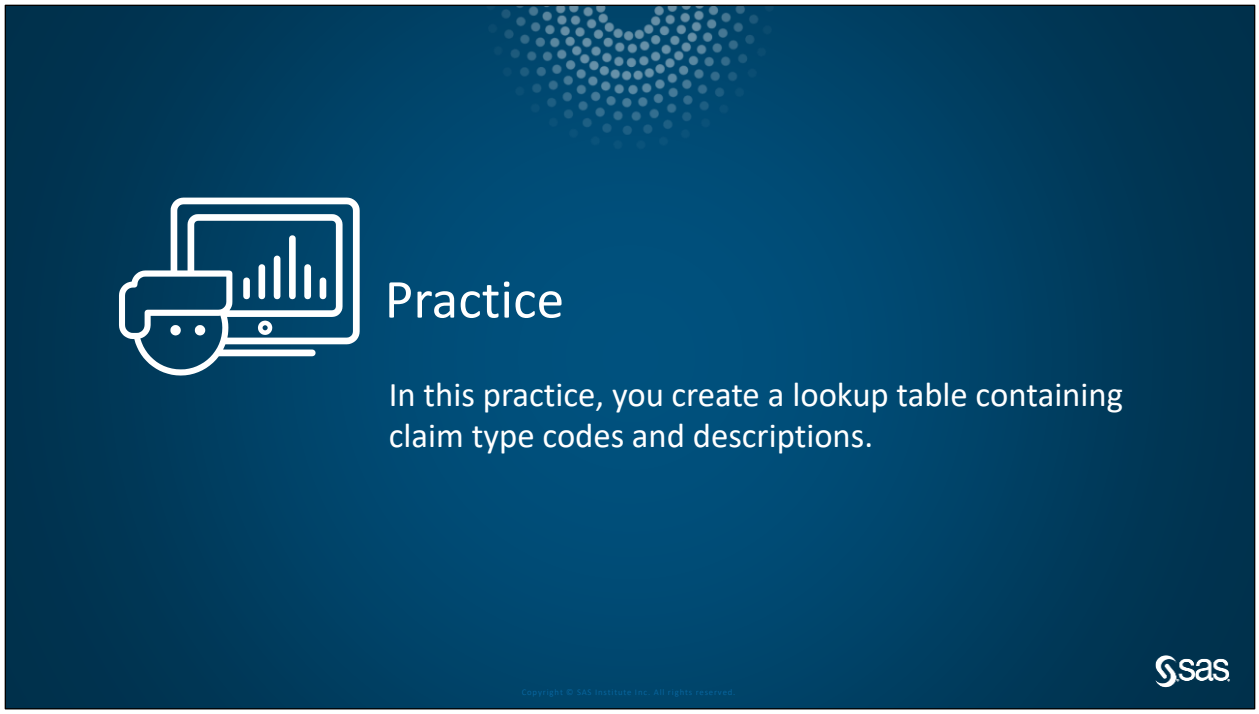
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

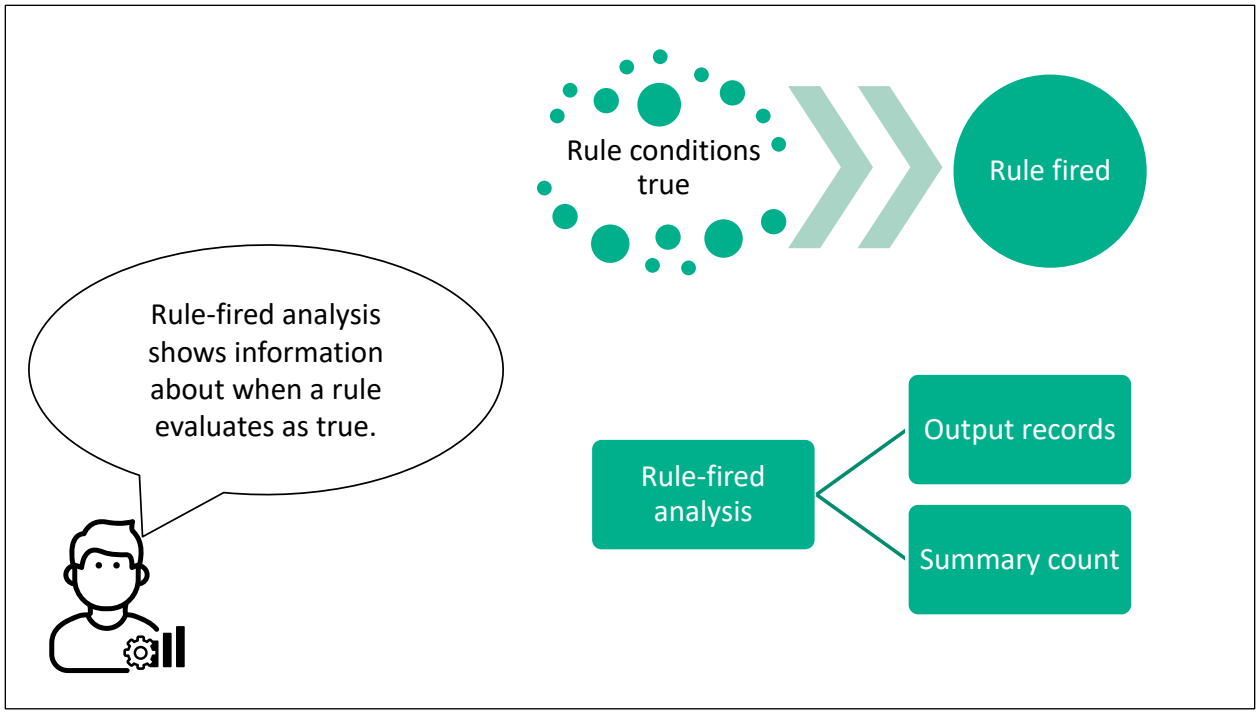


The Subject Level table describes the type of entity that the decision corresponds to. For example, subject levels might correspond to individual customers or accounts. The default key values in the table are shown here.



Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





If a rule's conditions evaluate to true, then the rule is said to have *fired*. Rule-fired analysis results include output records with details for each rule that evaluates to true and a summary of the count of how many times each rule fired.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Using Lookup Table Functions in a Rule Set

This demonstration illustrates using the LOOKUP and LOOKUPVALUE functions in a rule set.

Copyright © SAS Institute Inc. All rights reserved.

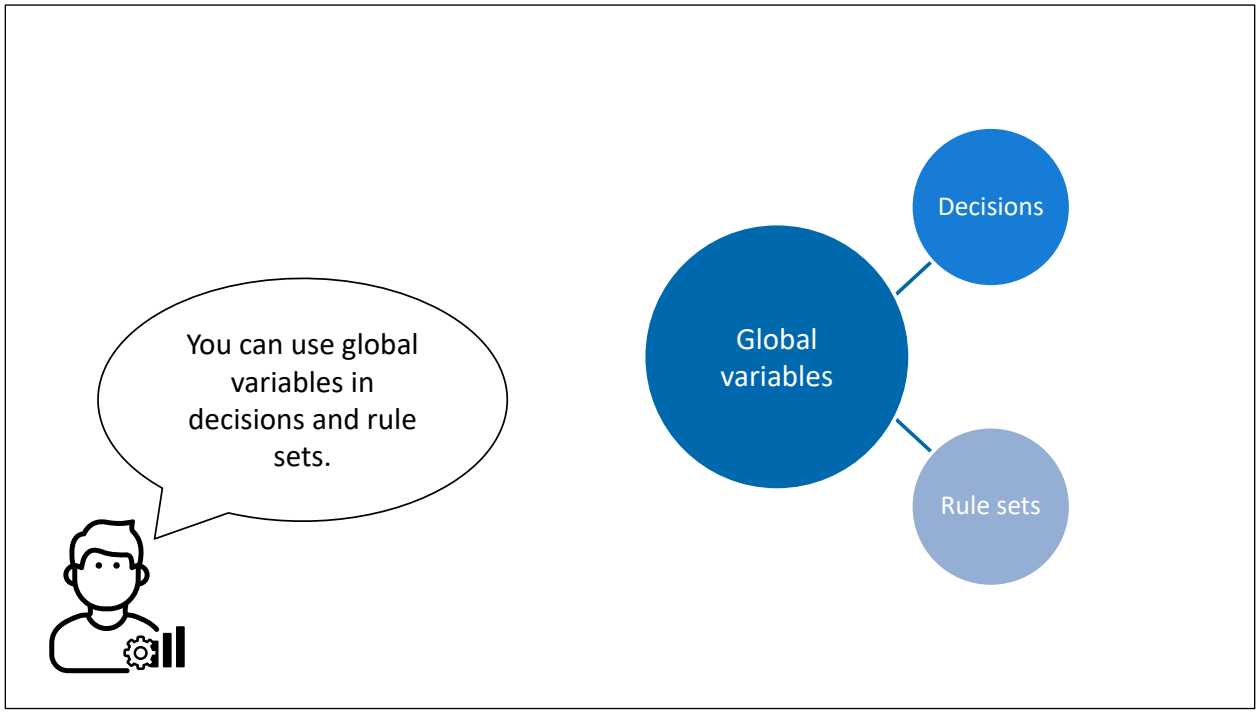
Practice

In this practice, you create a new version of a rule set and add an assignment using the LOOKUPVALUE lookup table function.

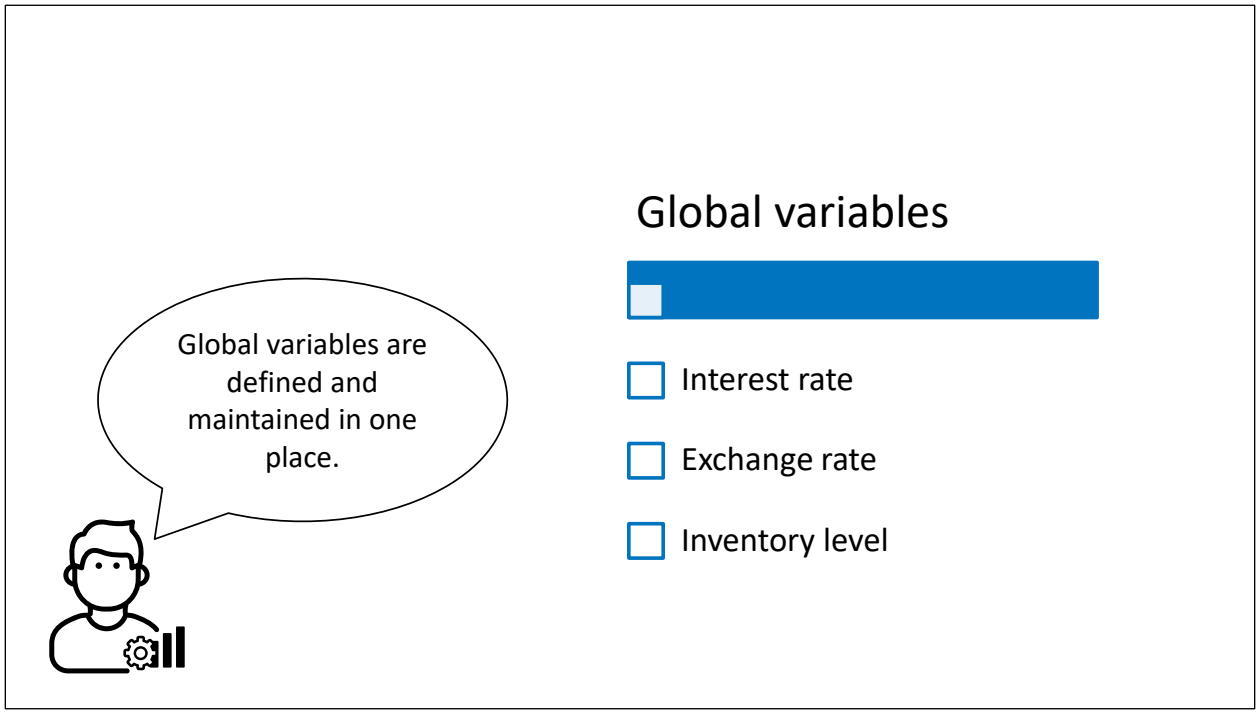
Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.3 Global Variables

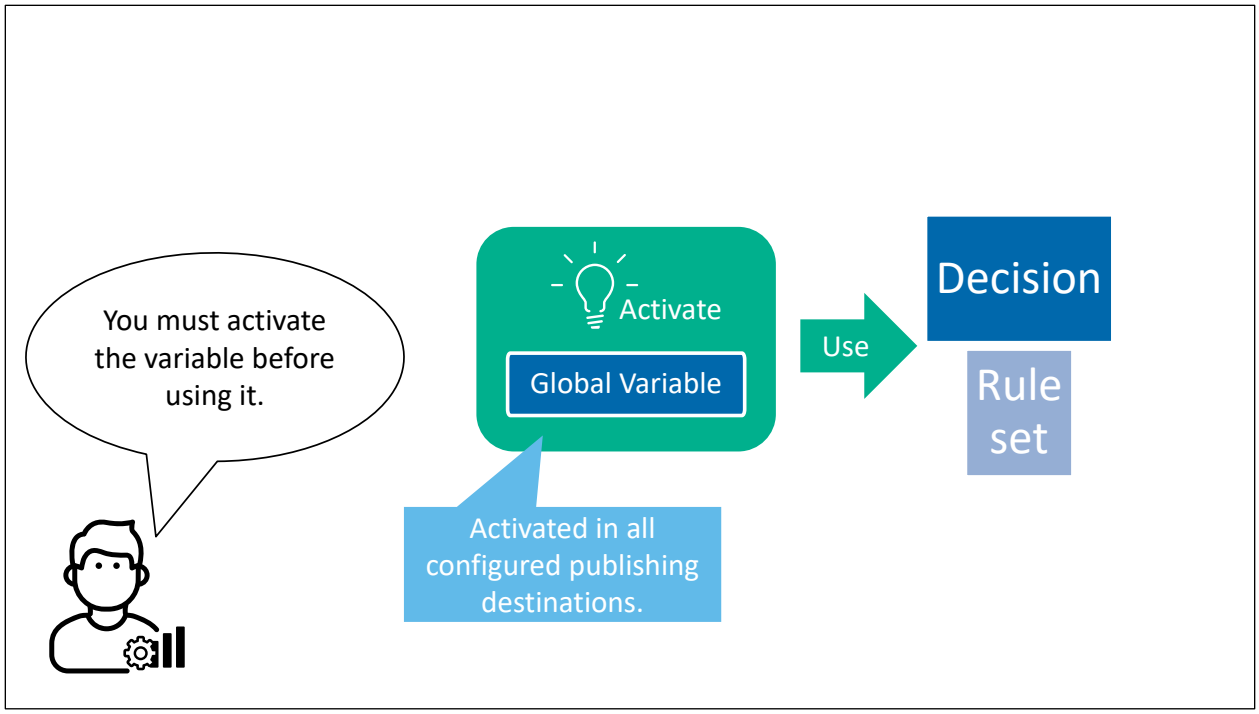


You can use global variables to define variables for use in multiple decisions and rule sets.

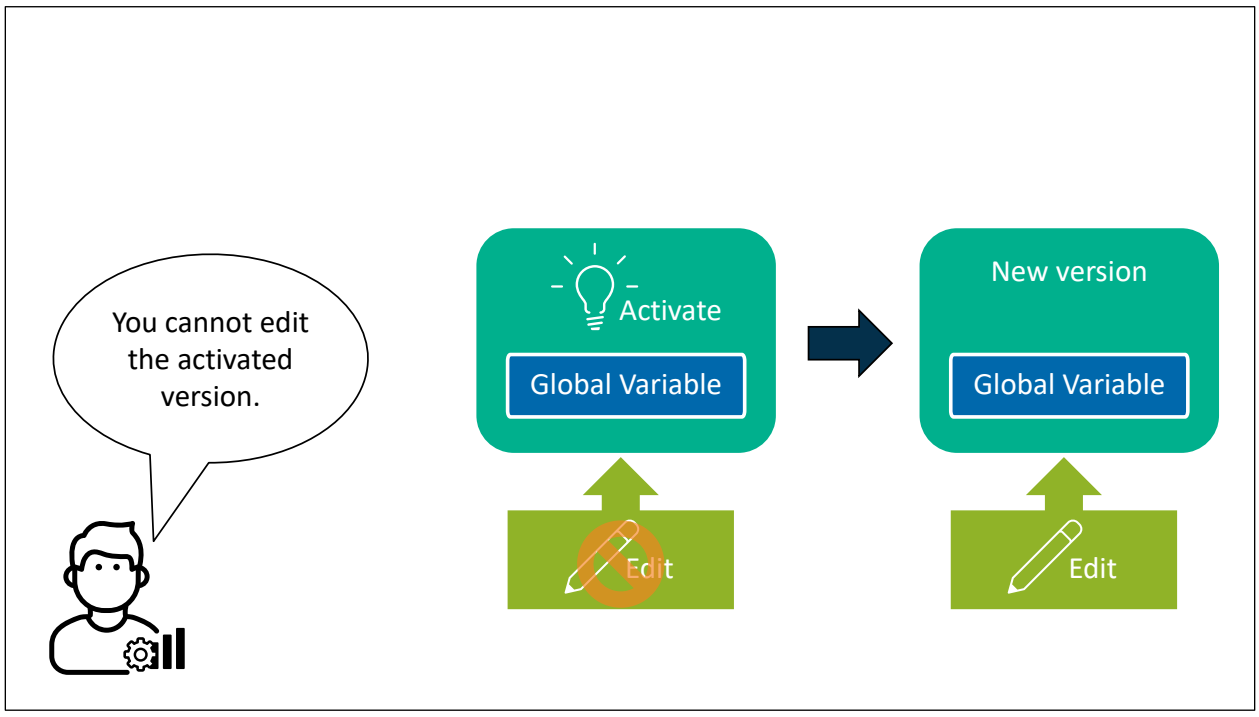


For example, you could define global variables that correspond to interest rates, exchange rates, or inventory levels. Global variables are defined and maintained in one place.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

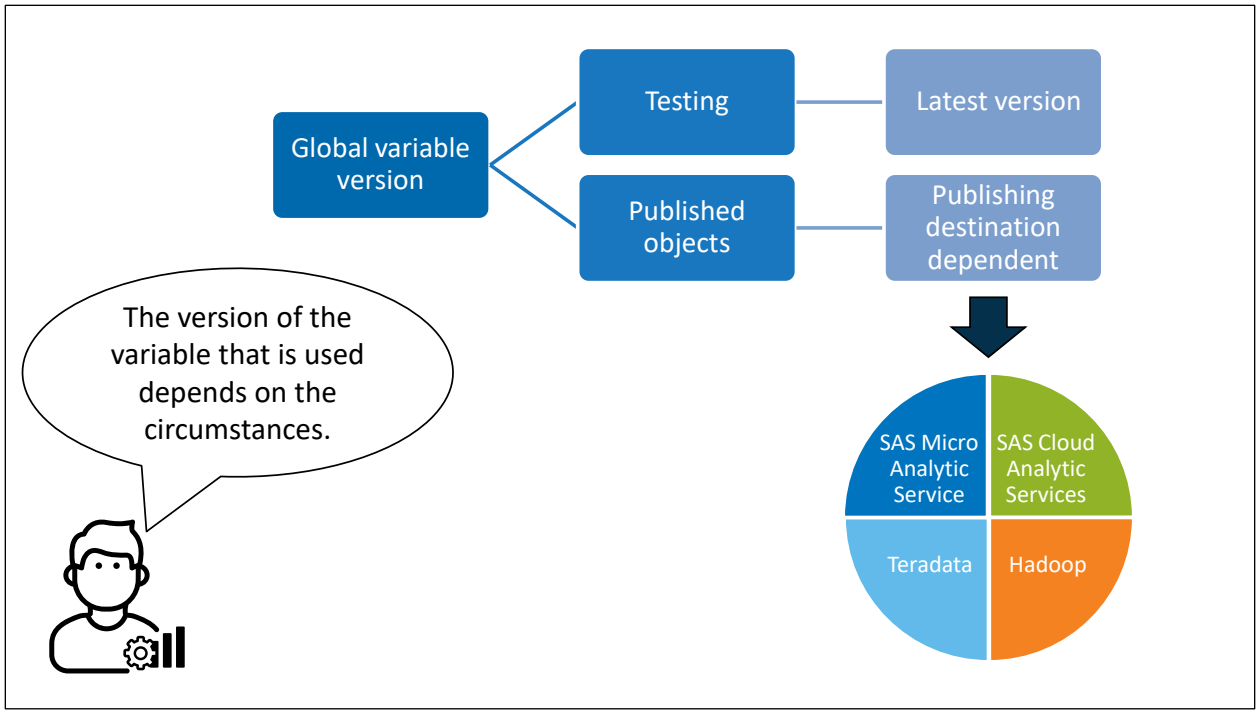


You must activate a global variable before you can use it. When you activate a global variable, it is activated in all of the publishing destinations that are configured at your site.

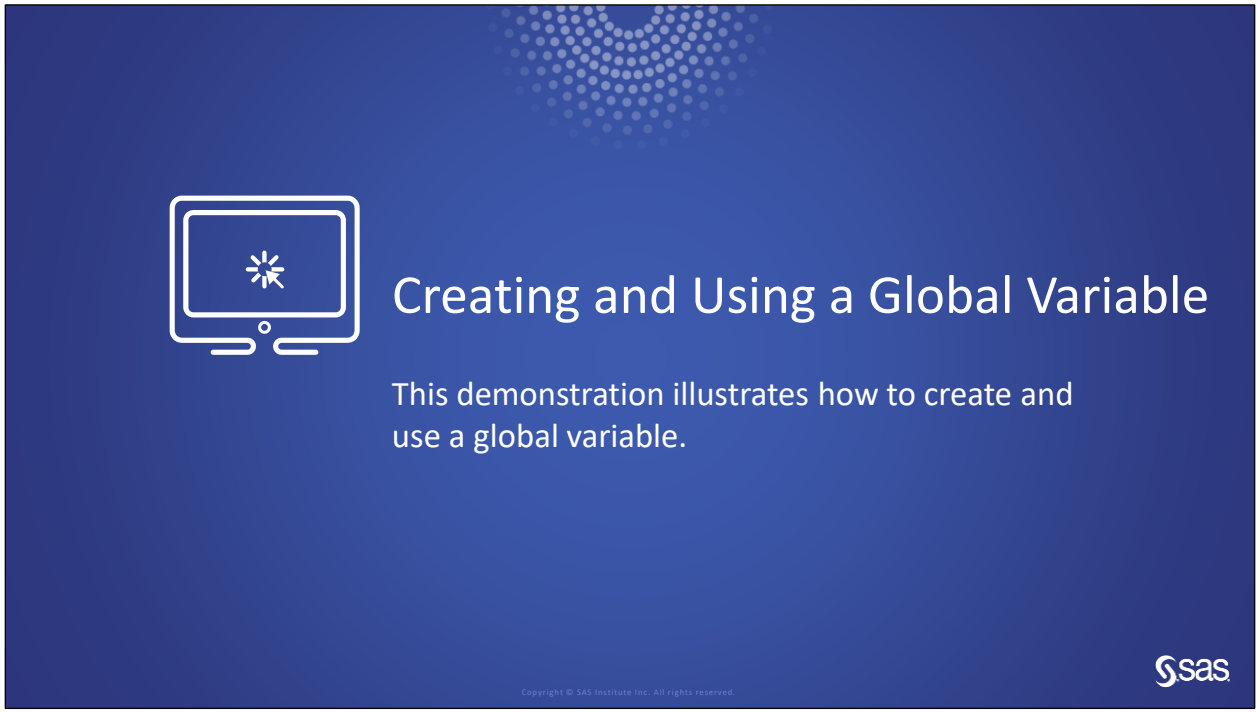


You cannot edit the activated version of a global variable. However, you can create a new version that can be edited.

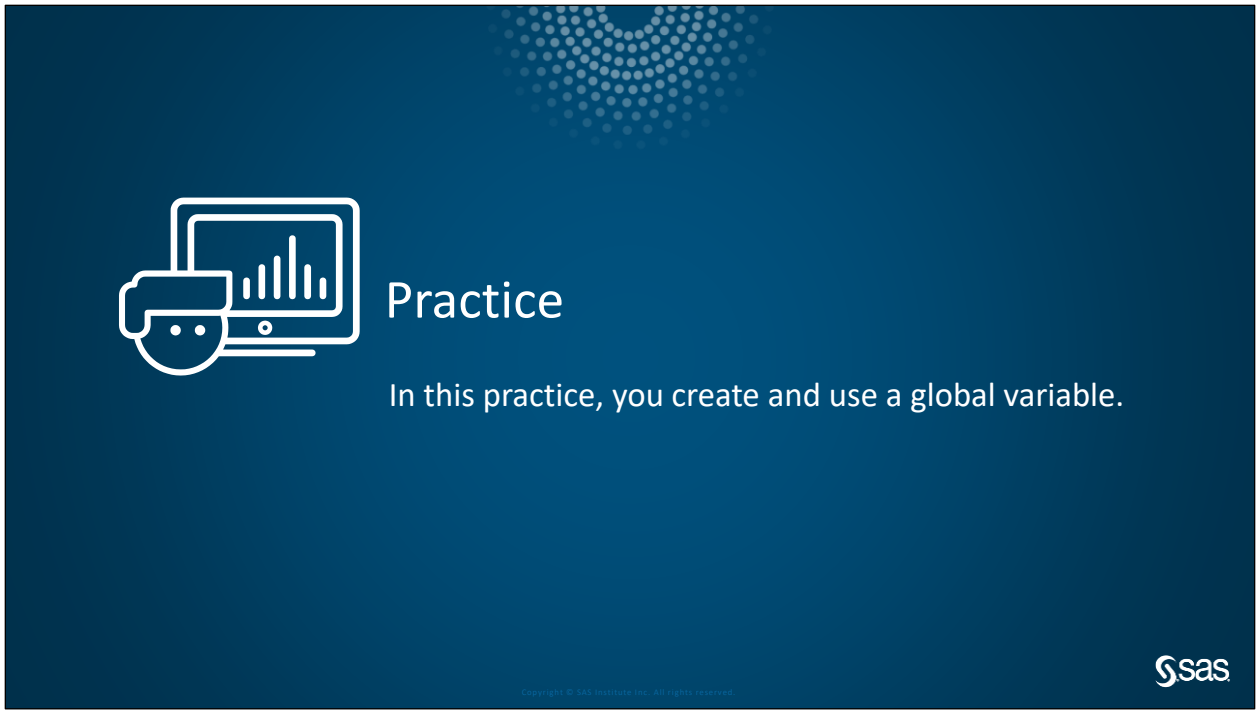
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



When you run a test for a rule set or decision that uses a global variable, the latest version (not necessarily the activated version) is used. The version that is used by published rule sets and decisions depends on the publishing destination.

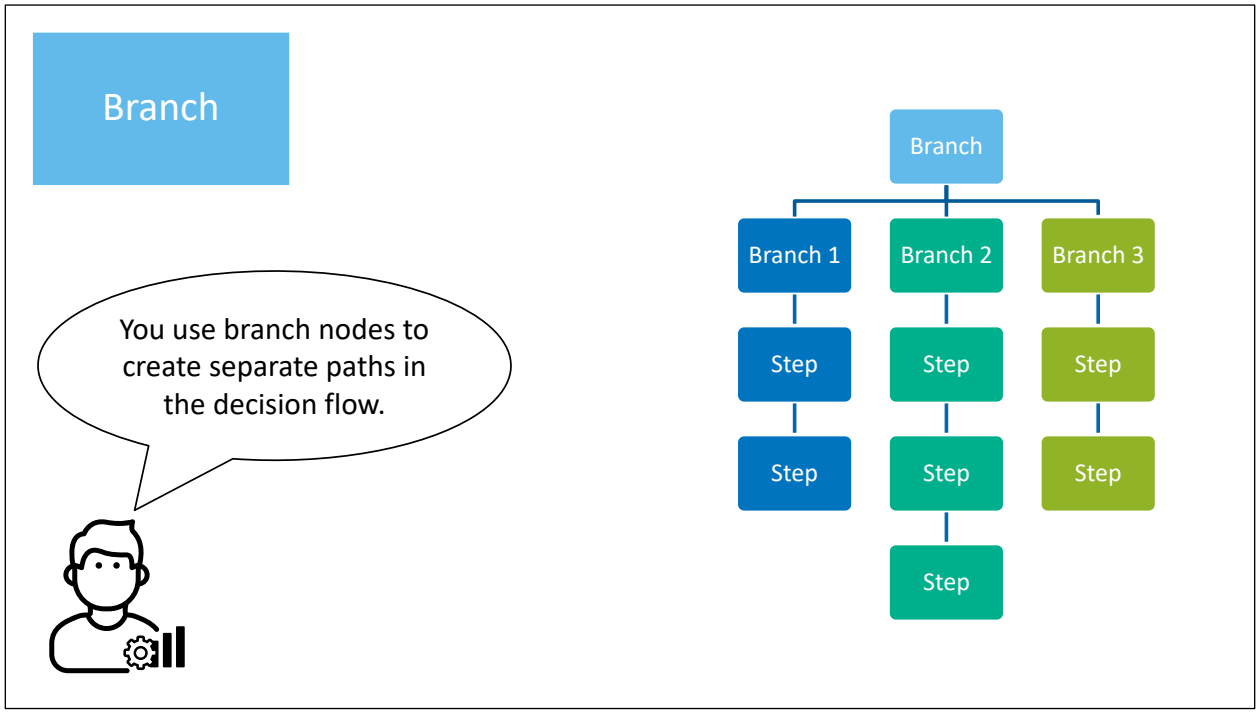


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

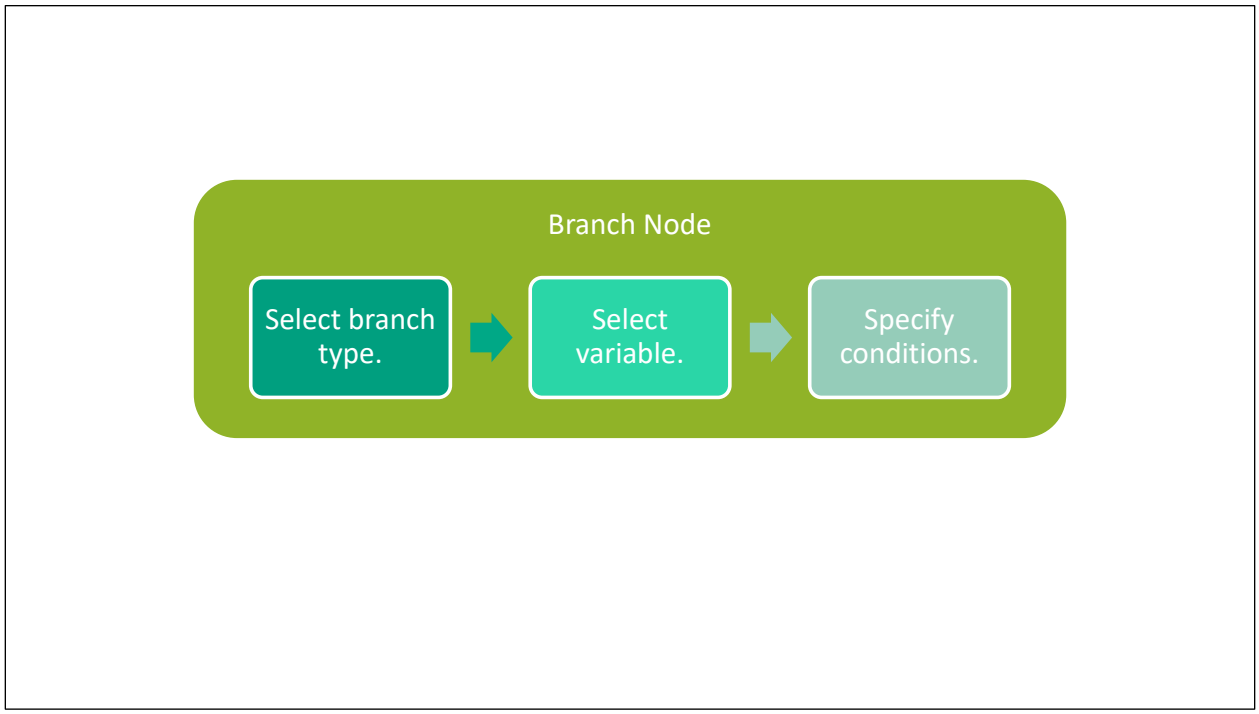


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.4 Branching and Cross-Branch Linking

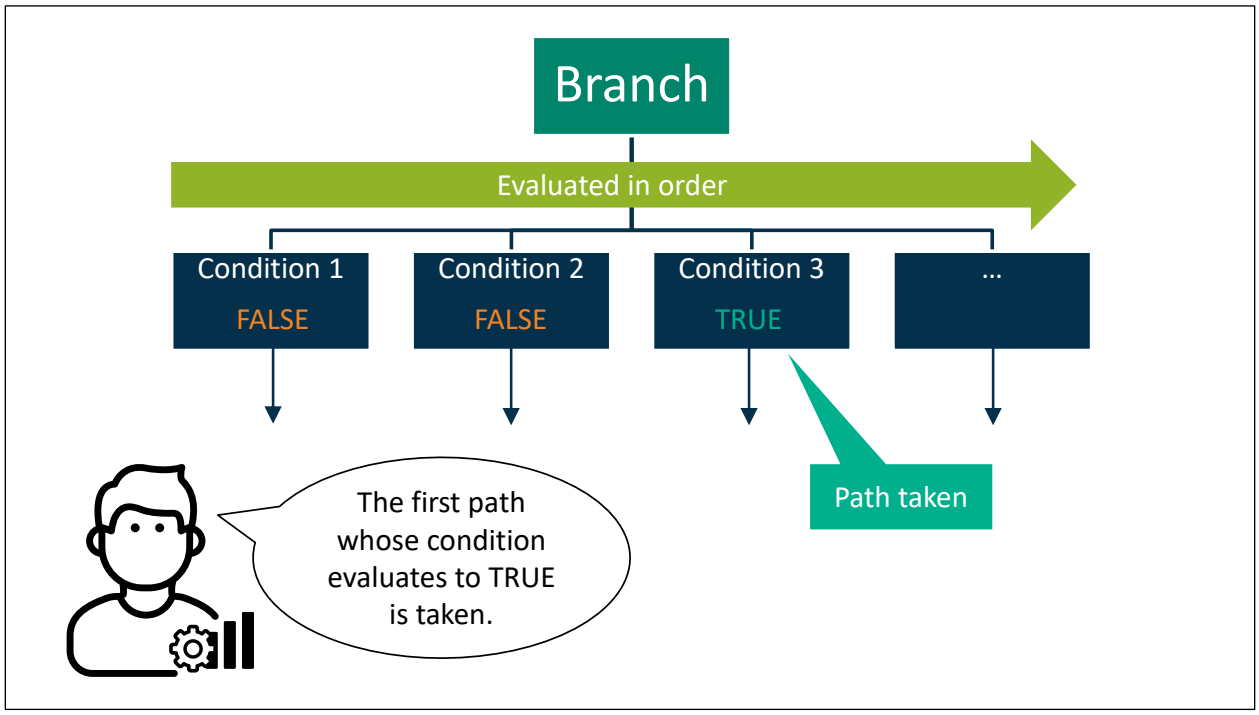


You can add a branch to a decision. Branches enable you to add conditional logic to a decision. A branch can potentially include multiple conditions and therefore have multiple outgoing paths. You can then define different downstream processing for each path.

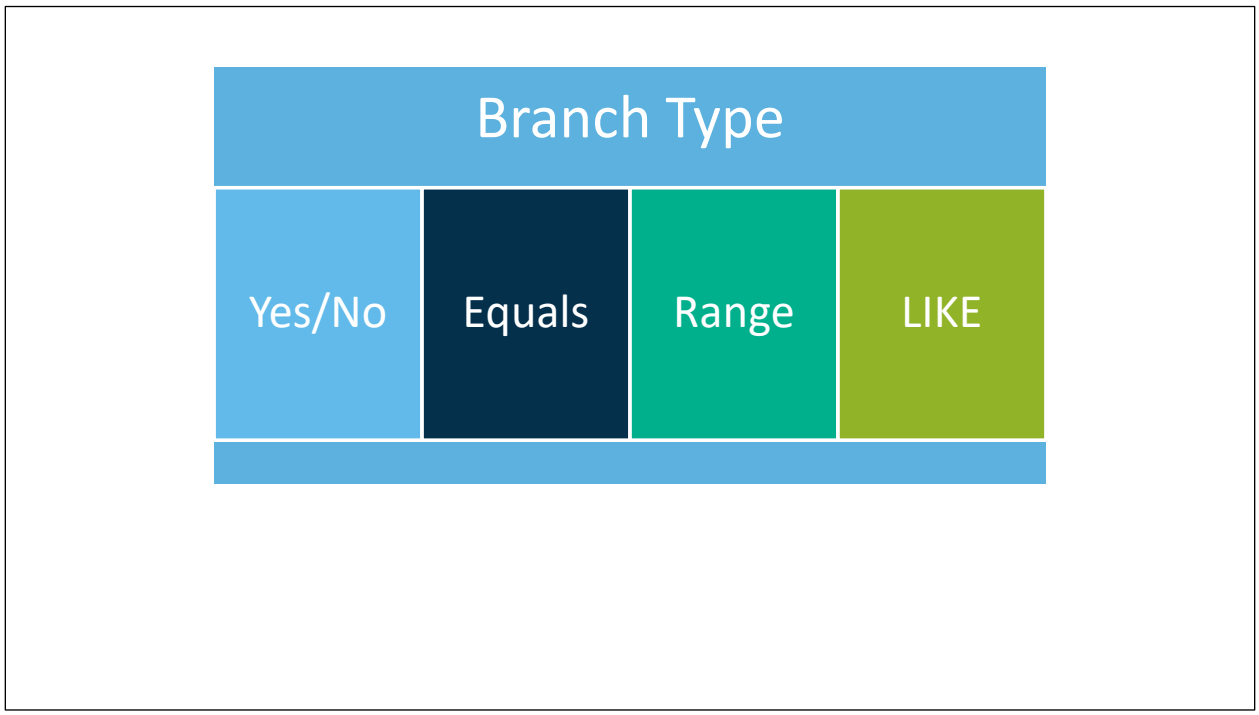


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

When you configure a branch node, you select a branch type, select a decision variable, and specify one or more conditions. The branch type determines the decision variables that you can select and the type of conditions that you can specify.

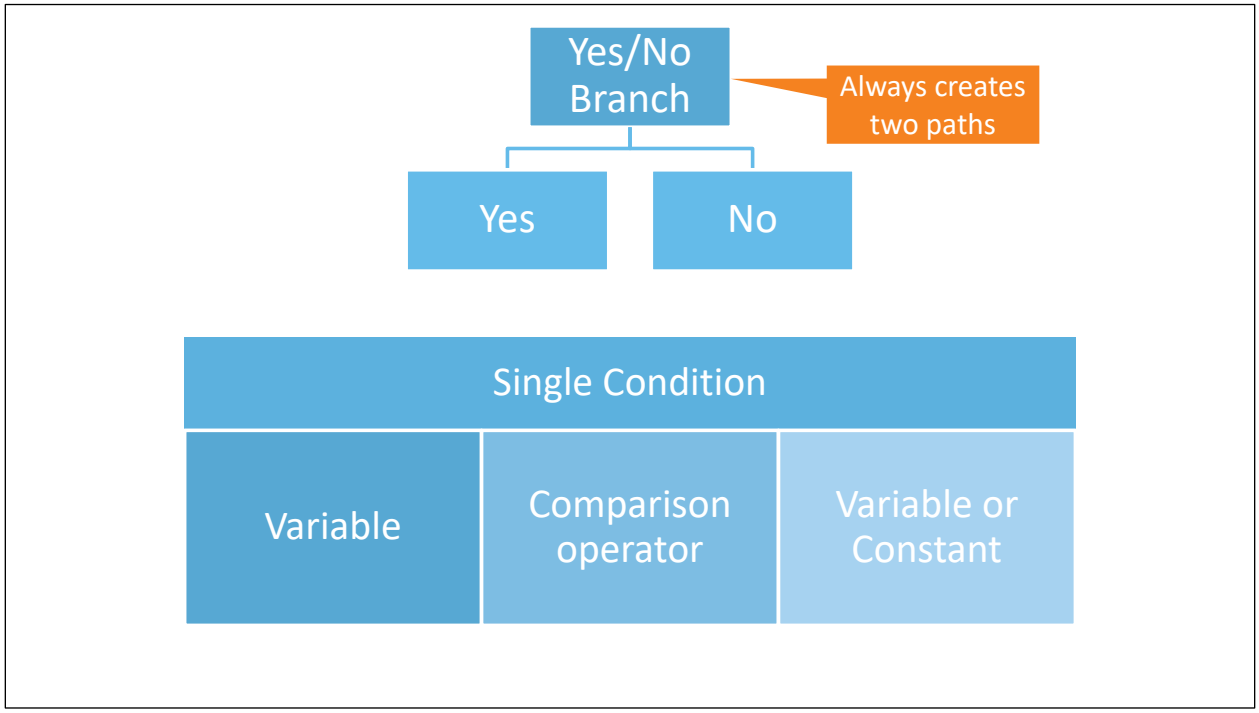


The branch node creates one output path for each condition. The conditions for the branch paths are evaluated in the order in which you specify them. When a decision executes, the first path whose condition evaluates to TRUE is taken.

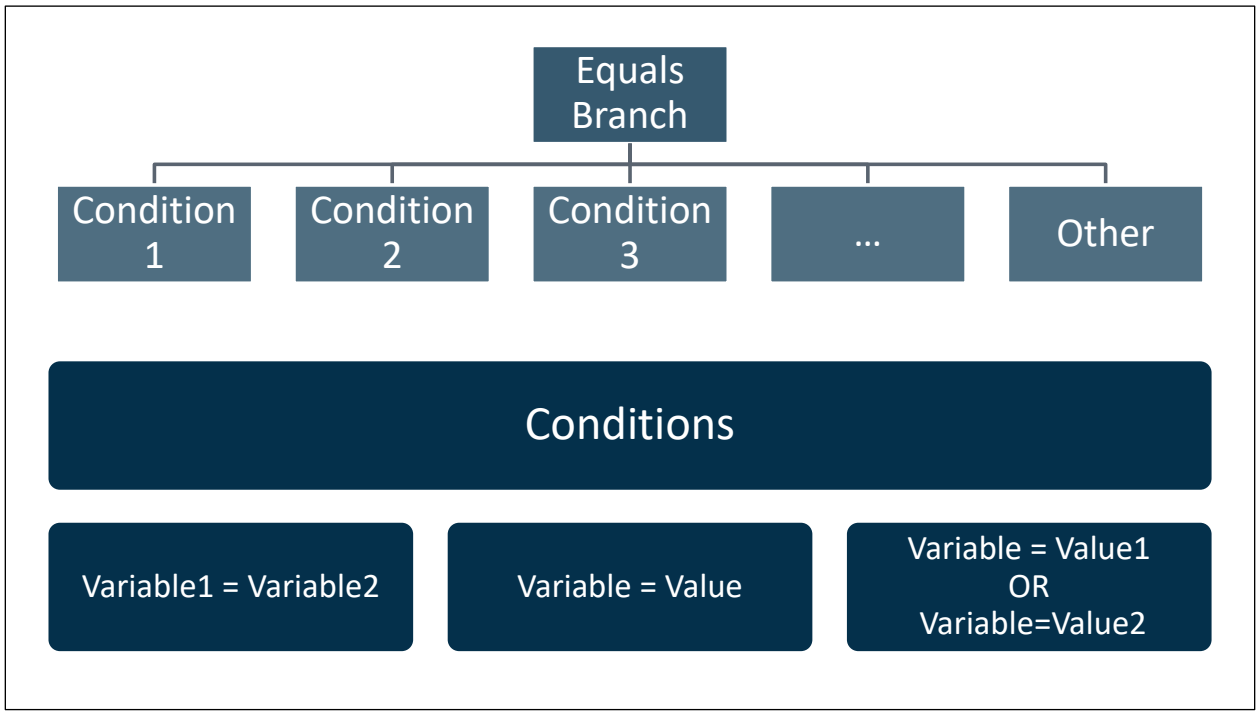


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

There are four branch types: Yes/No, Equals, Range, and LIKE.

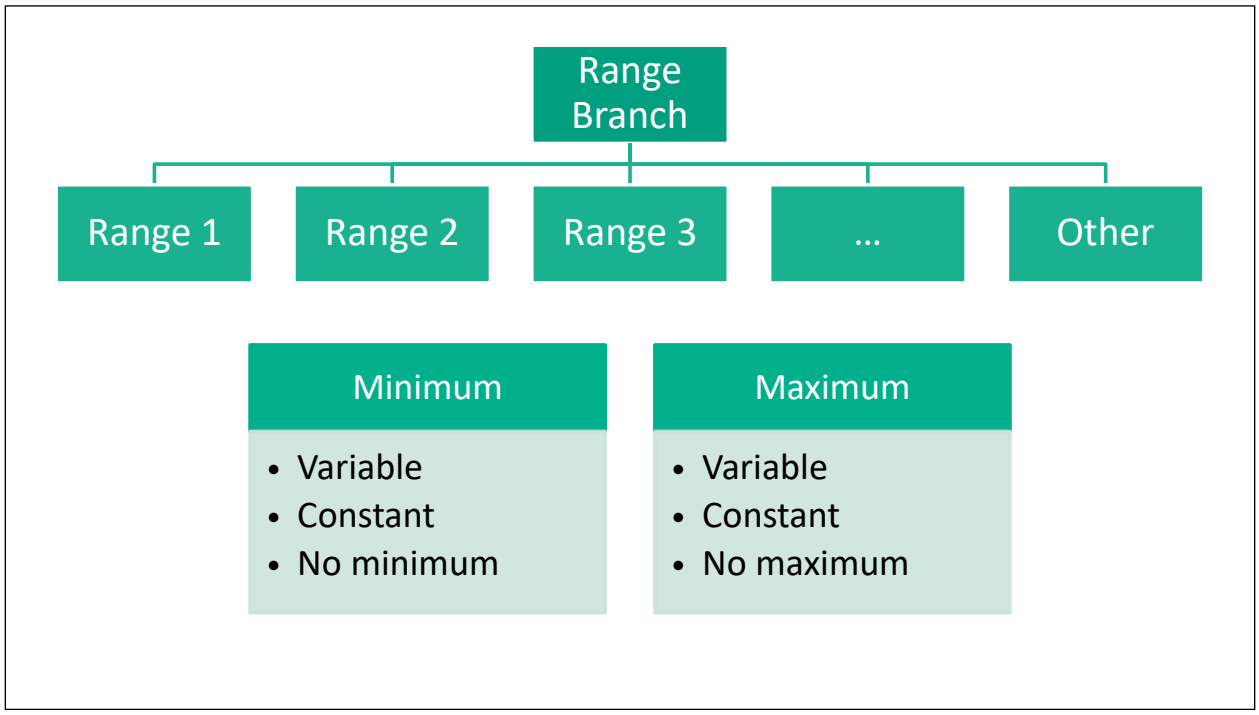


In a Yes/No branch, you define a single condition using a decision variable, a comparison operator, and a variable or constant. This type of branch always creates two paths in the flow, one for yes (when the condition is true) and one for no (when the condition is false).

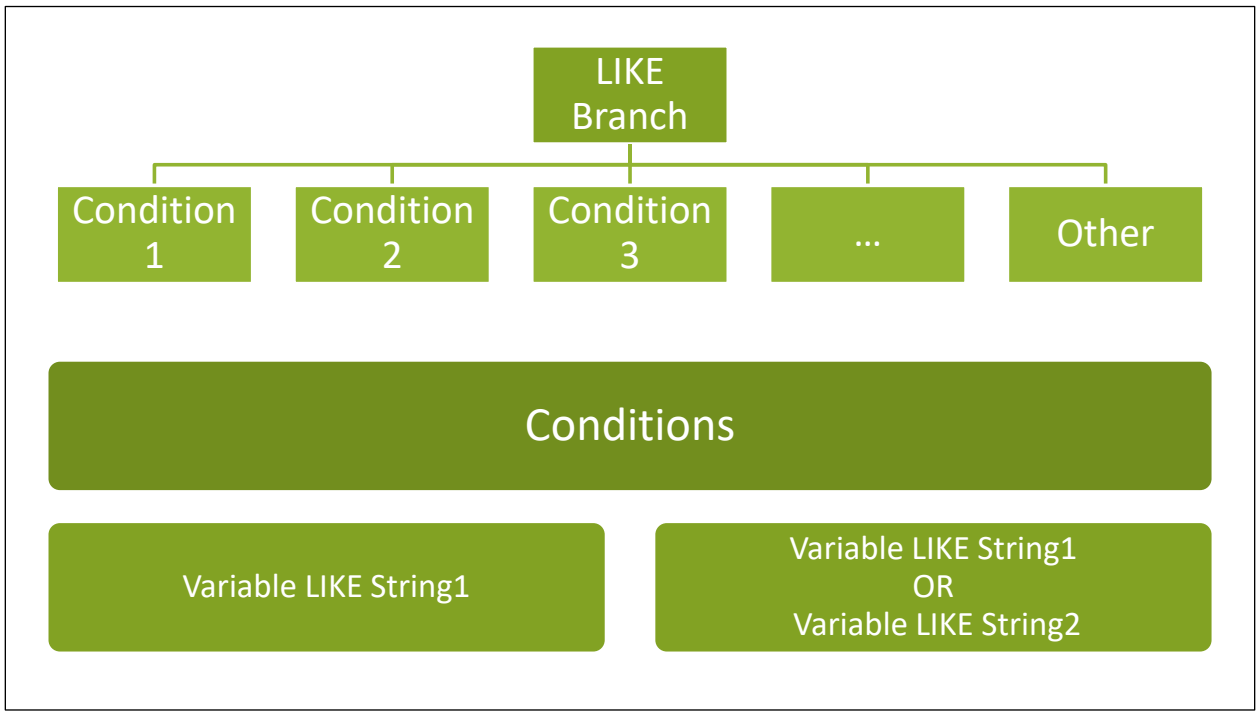


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

An Equals branch enables you to perform multiple comparisons based on equality and create an output path for each. You can create a condition based on a comparison to another variable or to a constant value. You can also combine multiple comparisons together in a condition.

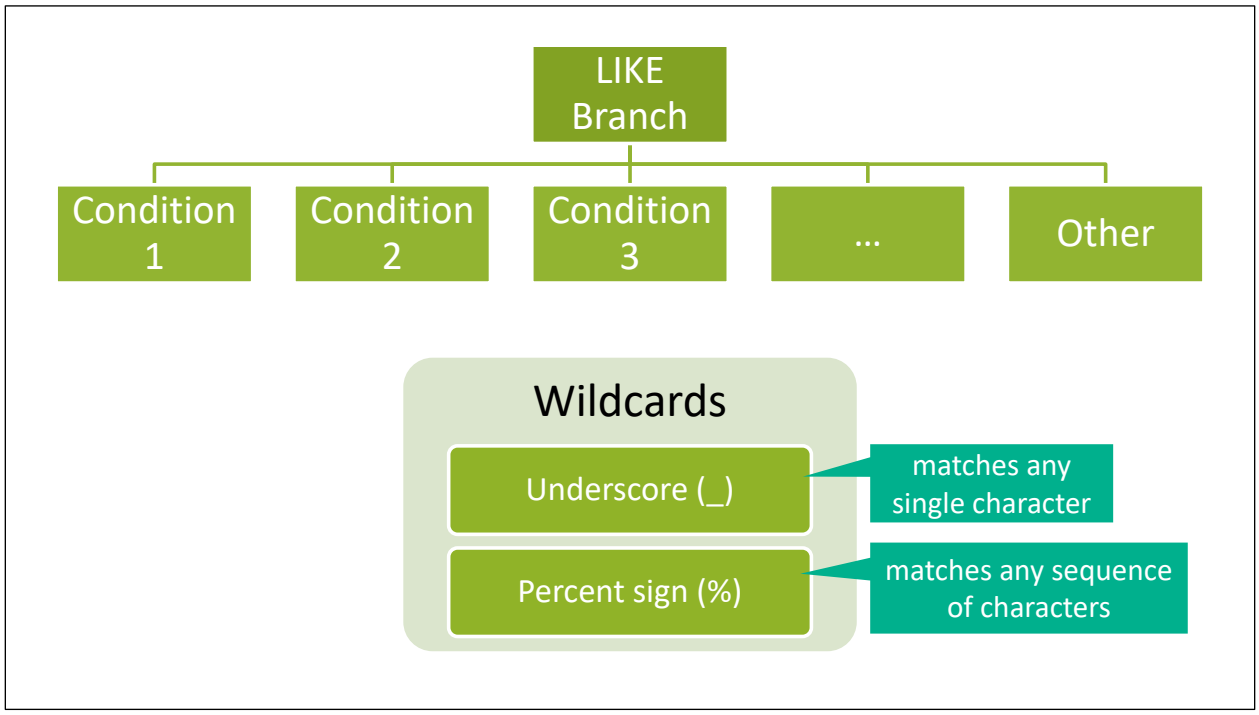


A Range branch enables you to compare the value of a numeric variable against one or more ranges of values. Each range condition includes a minimum and maximum. You can specify values for both using variables or constants. You can also omit the minimum or maximum for a range.

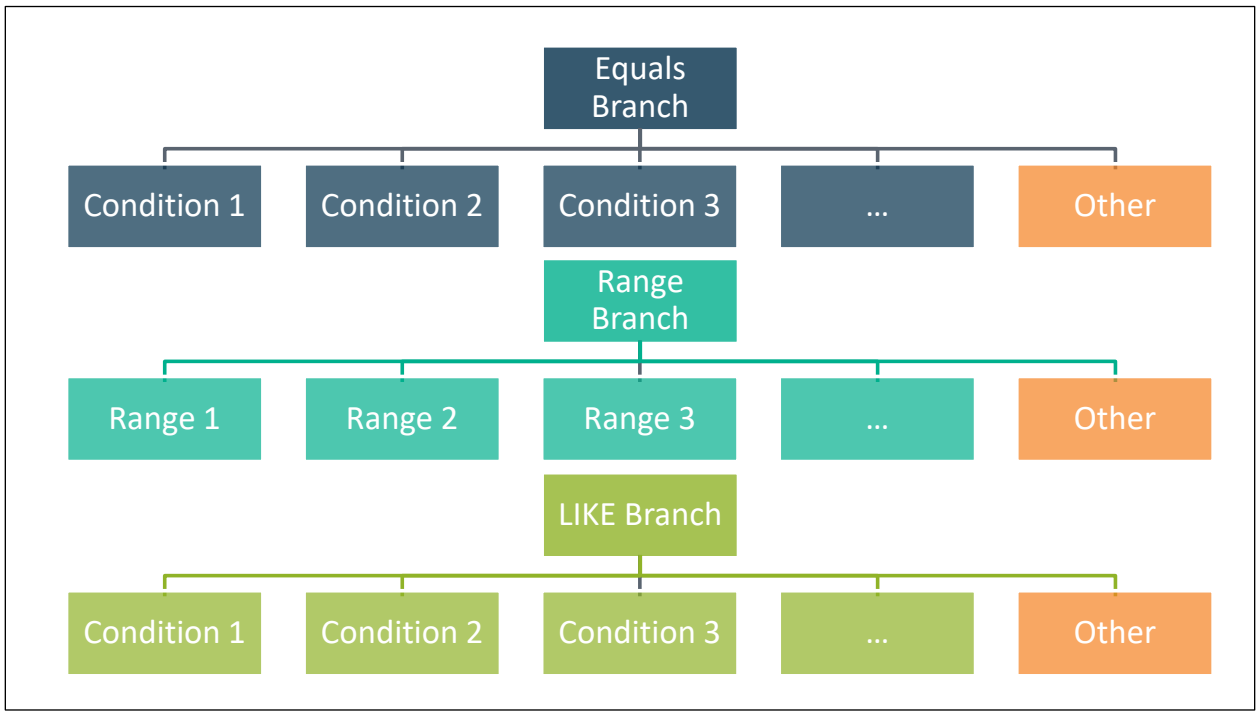


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

A LIKE branch enables you to compare the value of a character variable to one or more character strings with a pattern-matching specification using the LIKE operator. You can make a single comparison or combine multiple comparisons together in a condition.

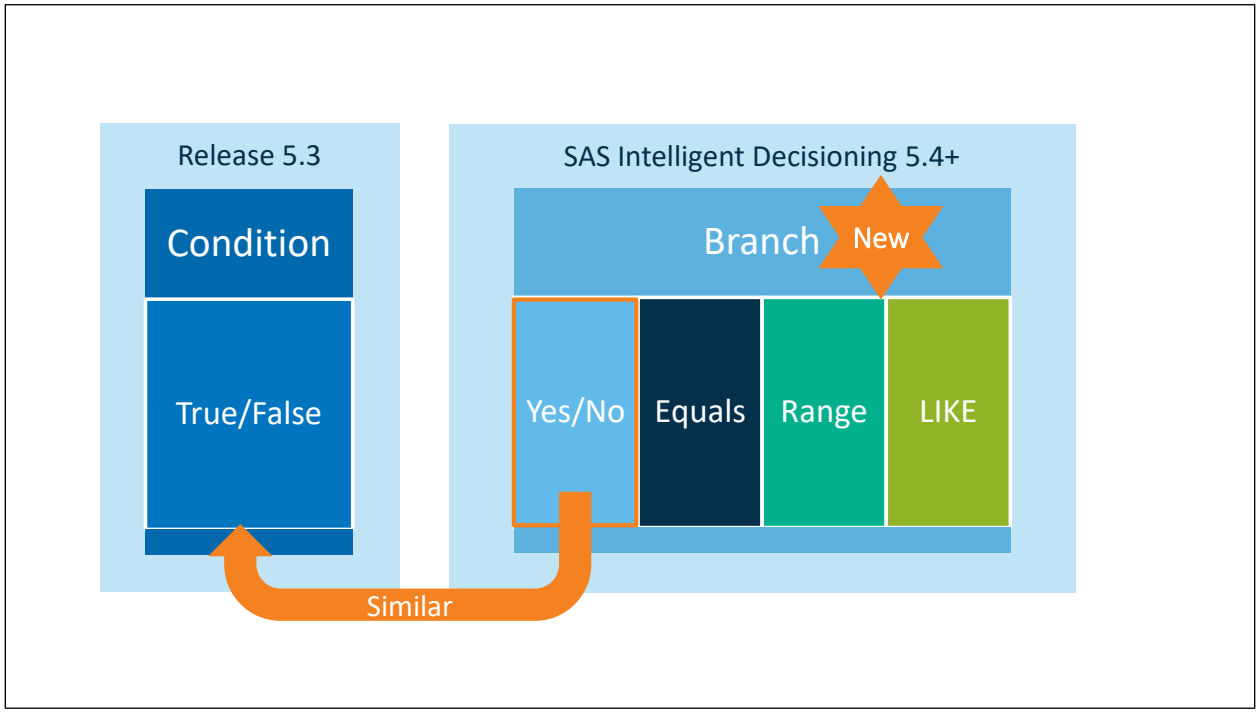


In LIKE expressions, you use the underscore and percent characters as wildcards in the pattern-matching specification. The underscore matches any single character, and the percent sign matches any sequence of zero or more characters.

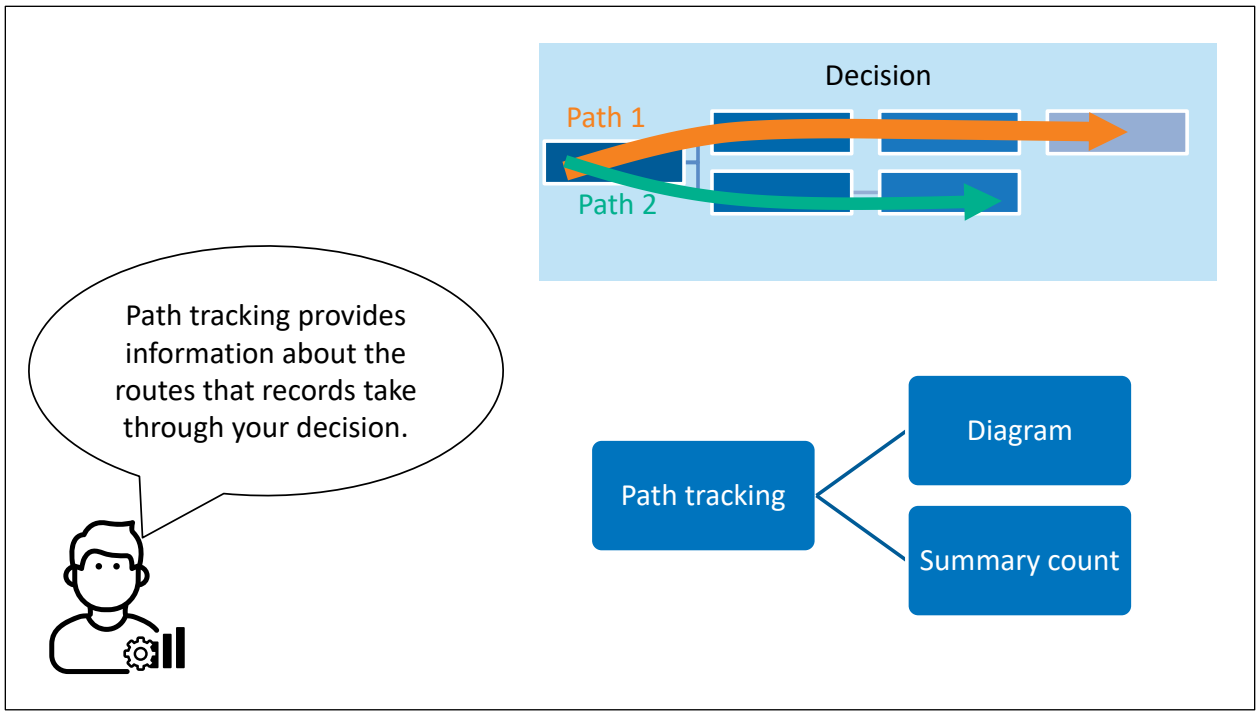


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Branches of type Equals, Range, or LIKE all create a branch labeled **Other** for any values that are not included in the branches that you create.

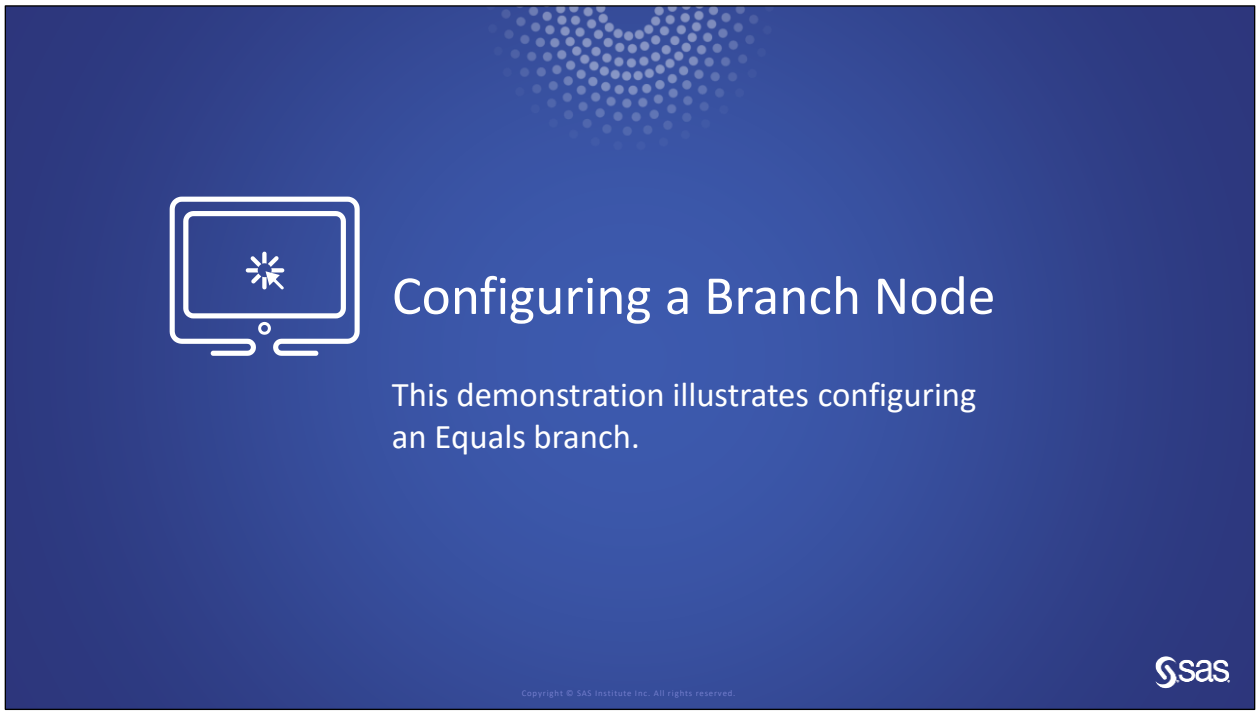


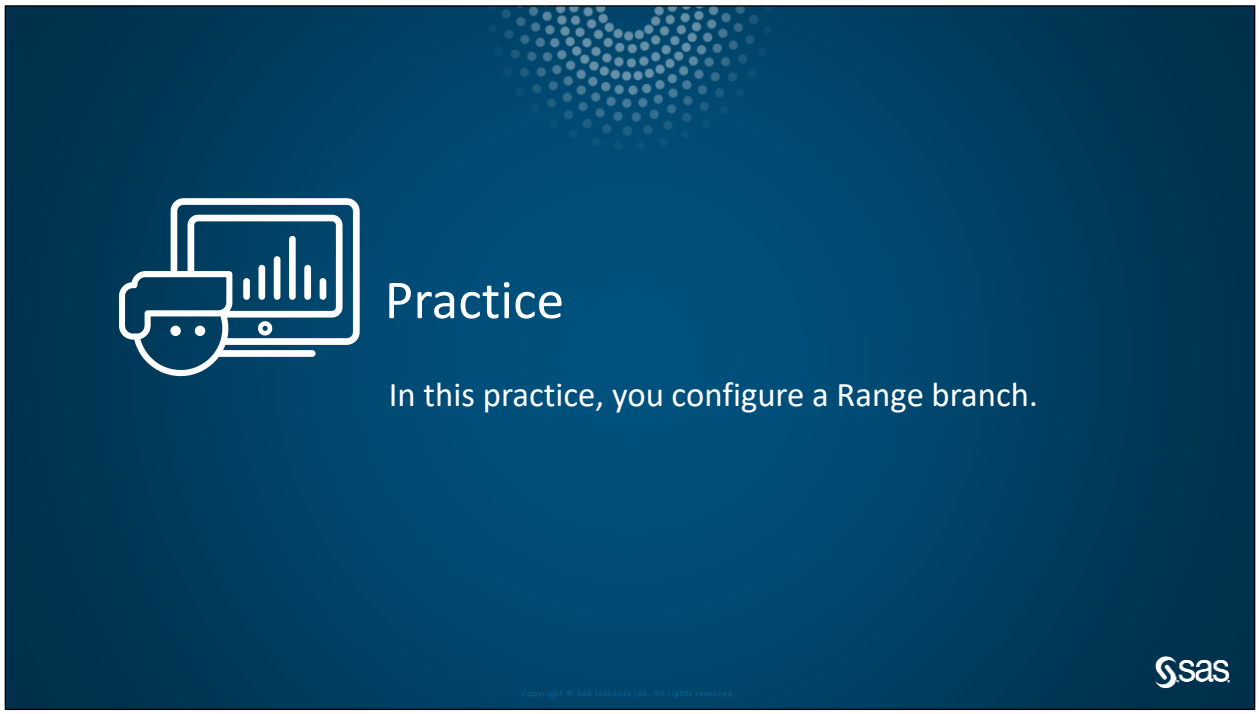
The branch node was introduced in SAS Intelligent Decisioning 5.4. In 5.3, you could perform processing similar to that of a Yes/No branch using a Condition node. If you open a decision that that was created in 5.3 and includes a Condition node, it is automatically converted to a Yes/No branch.



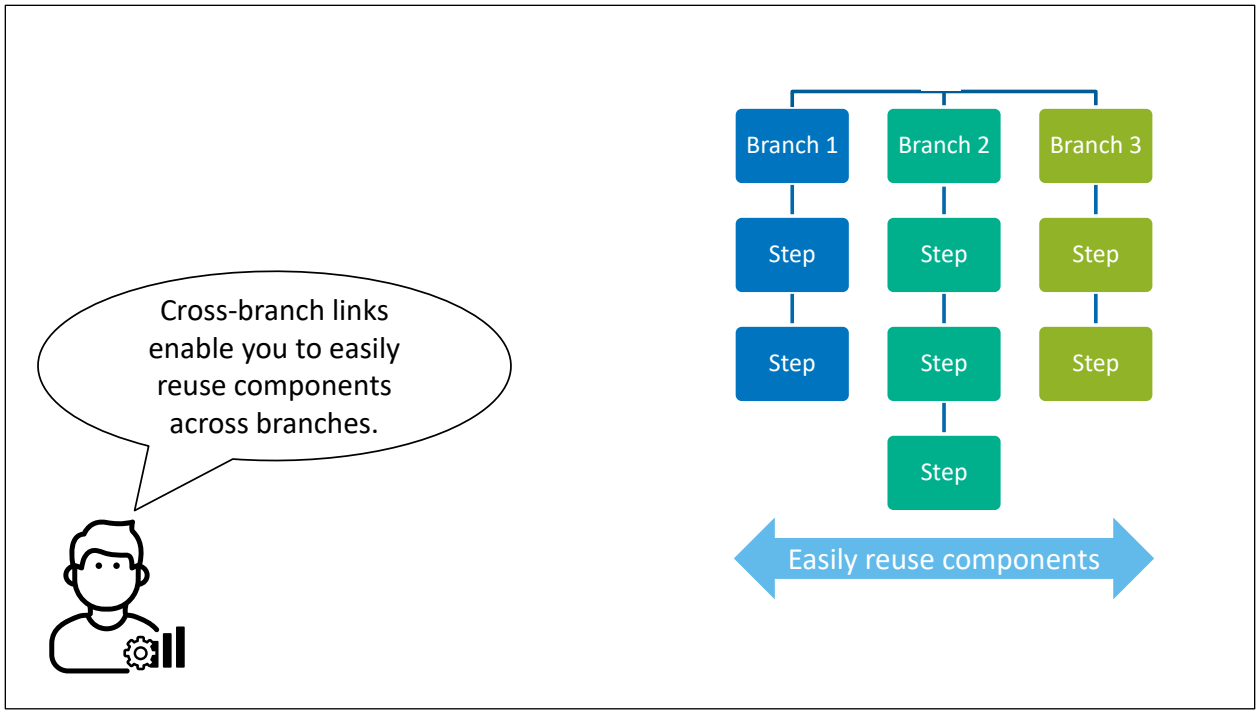
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Path tracking provides information about the routes that input records take through your decision. Path tracking results include a diagram that shows the flow of the input records through the nodes in the decision. It also includes a summary count of records passing through each node.

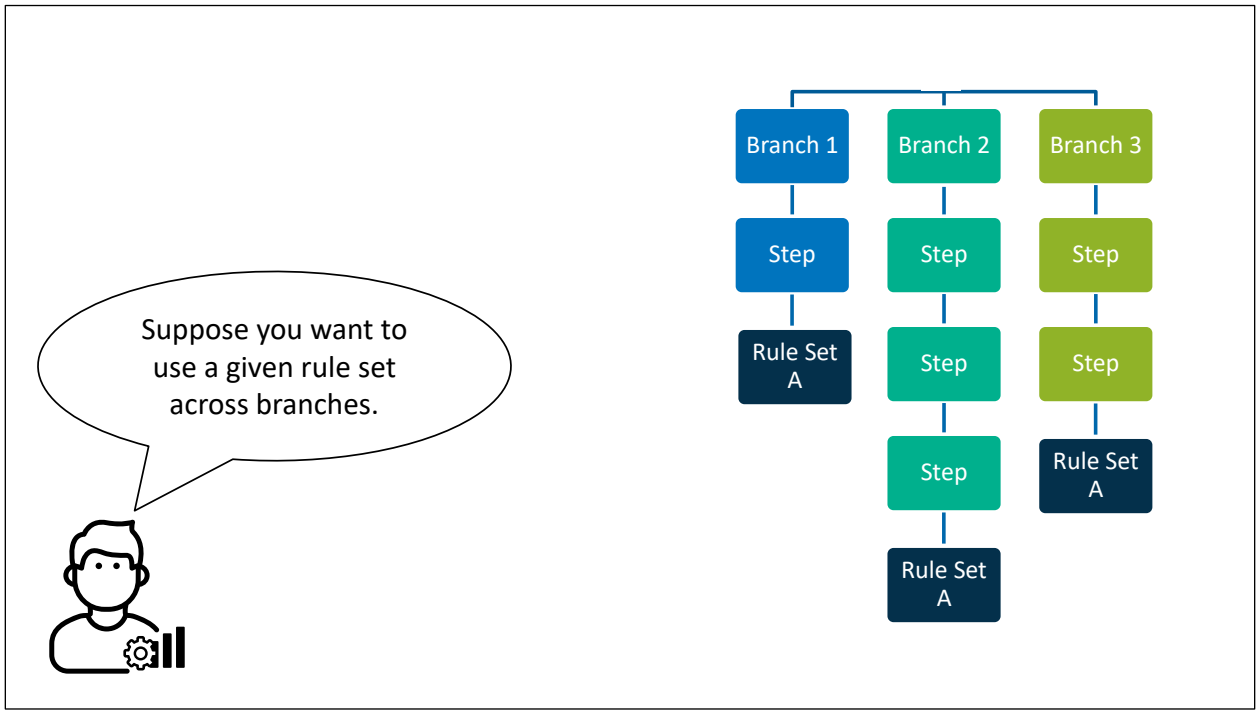




Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

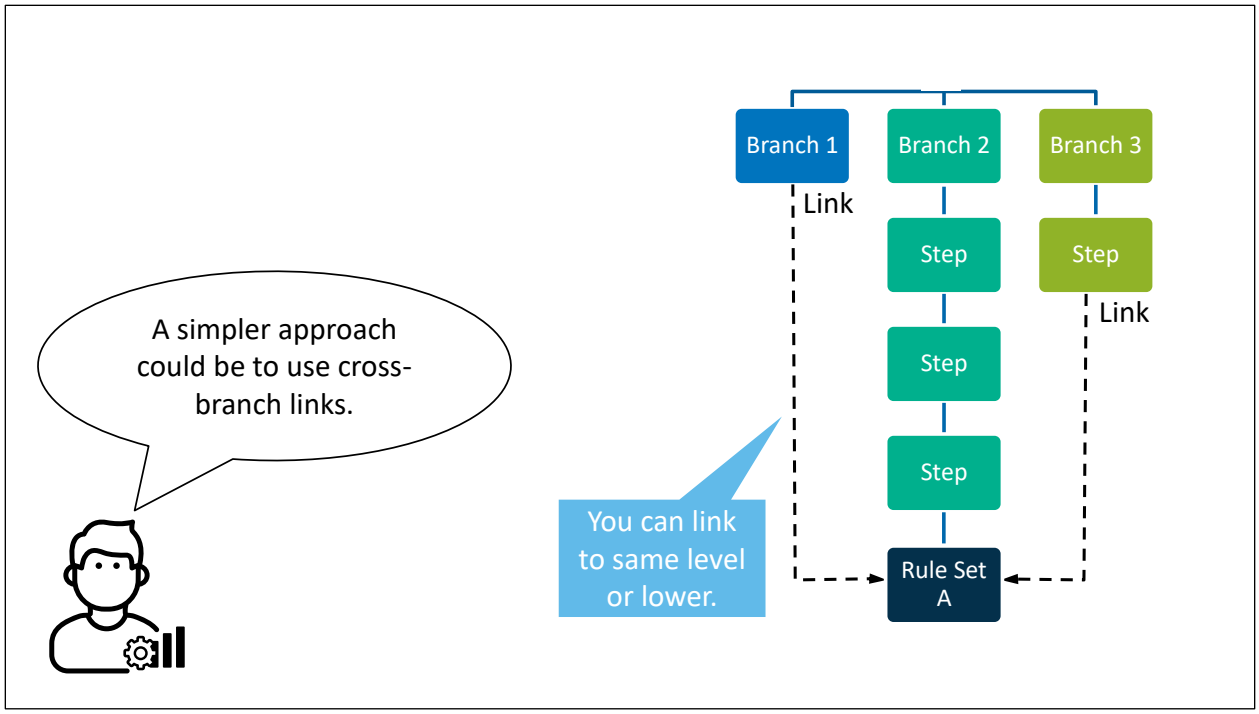


Cross-branch links enable you to easily reuse components across branches of a decision.

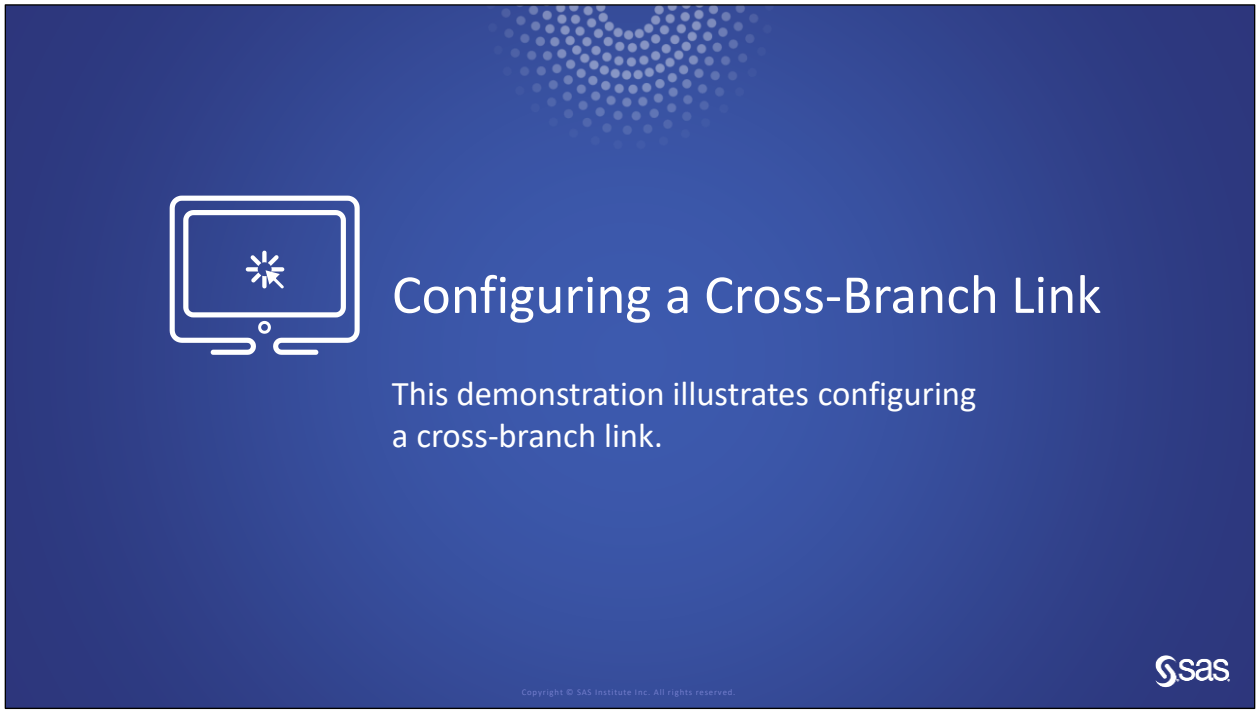


For example, suppose you want to use a given rule set across branches in a decision. One way to do so would be to duplicate the rule set across all the branches.

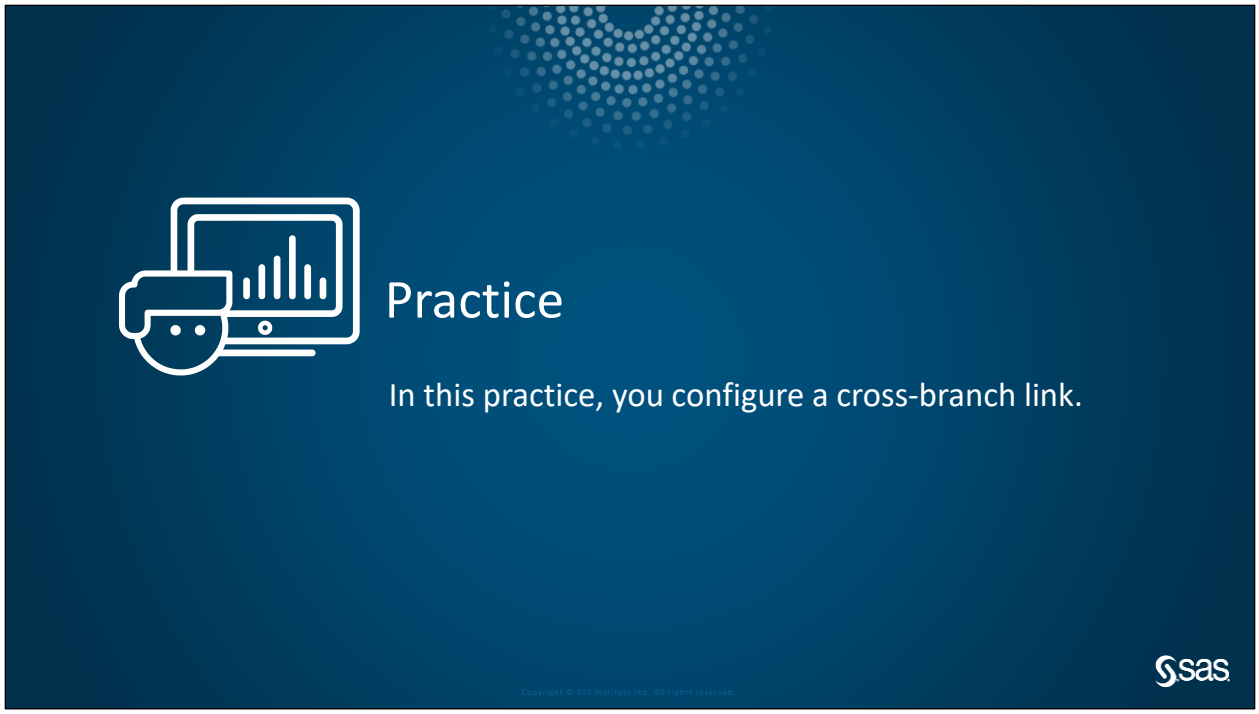
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



A simpler approach could be to add the rule set once and then use cross-branch links to link to it from the other branches. You can add a cross-branch link to a node at a level that is the same as or lower than the level of the source node.

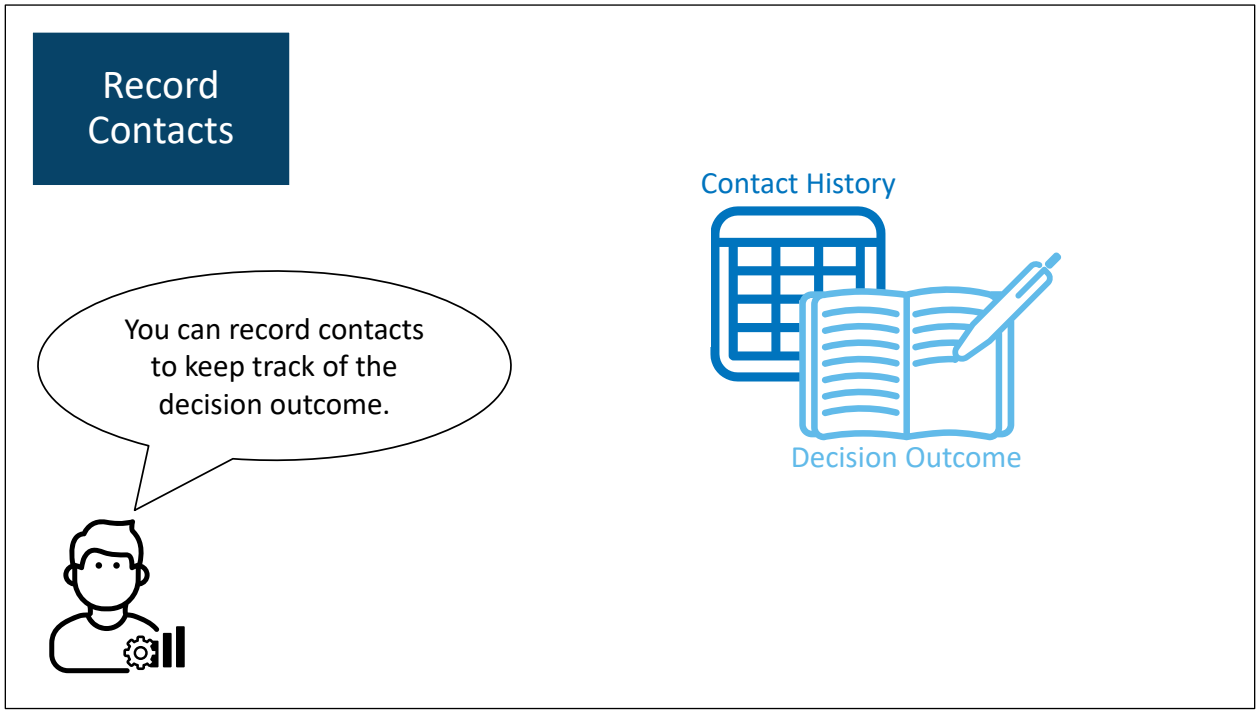


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



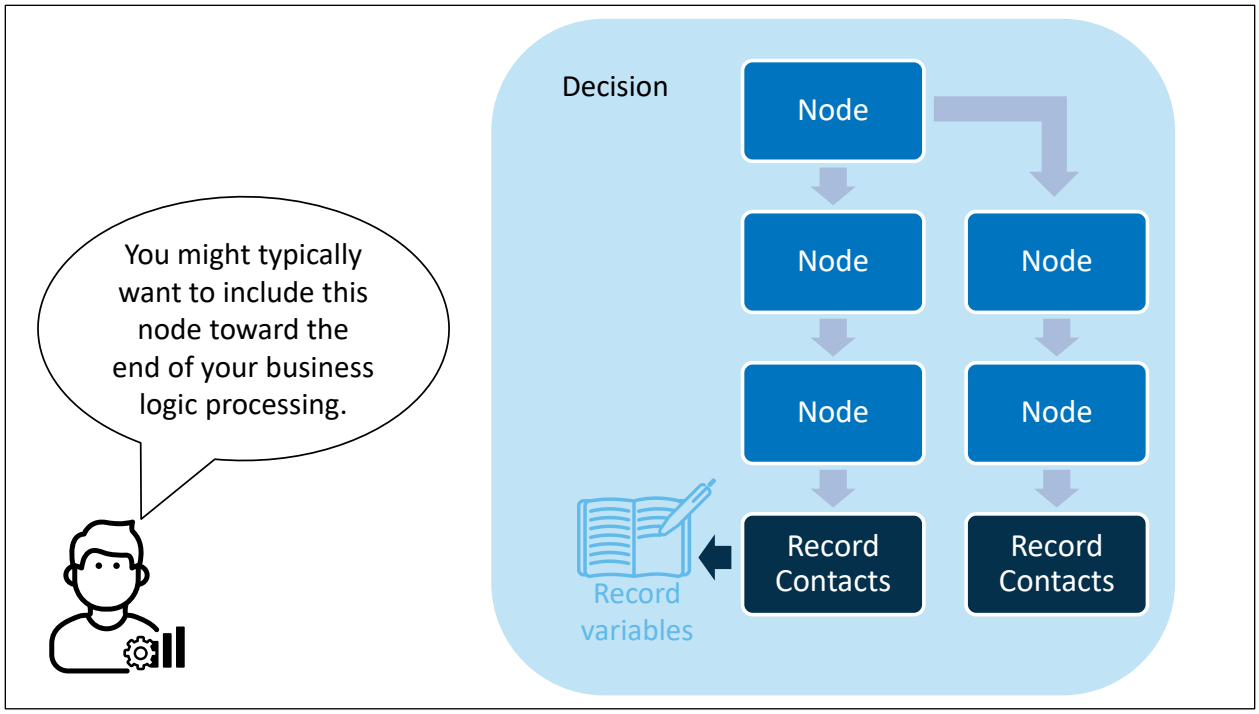
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.5 Record Contacts

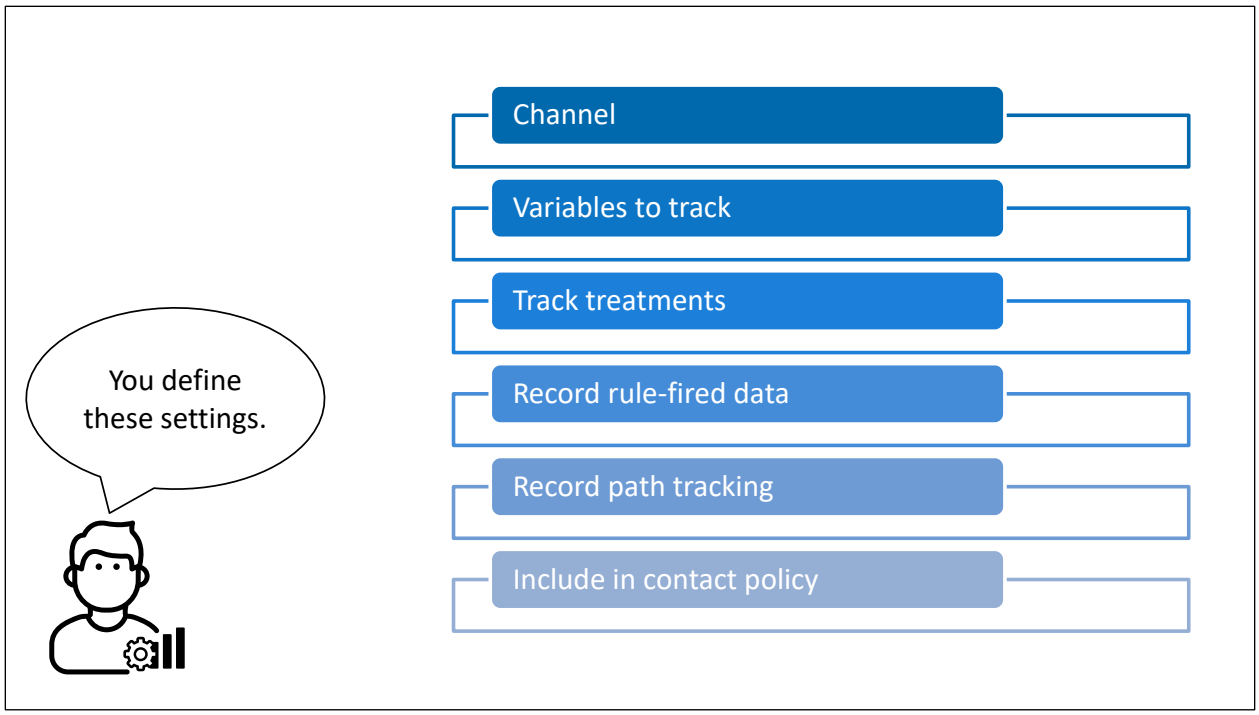


You can record contacts to keep track of the outcome of the decision.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

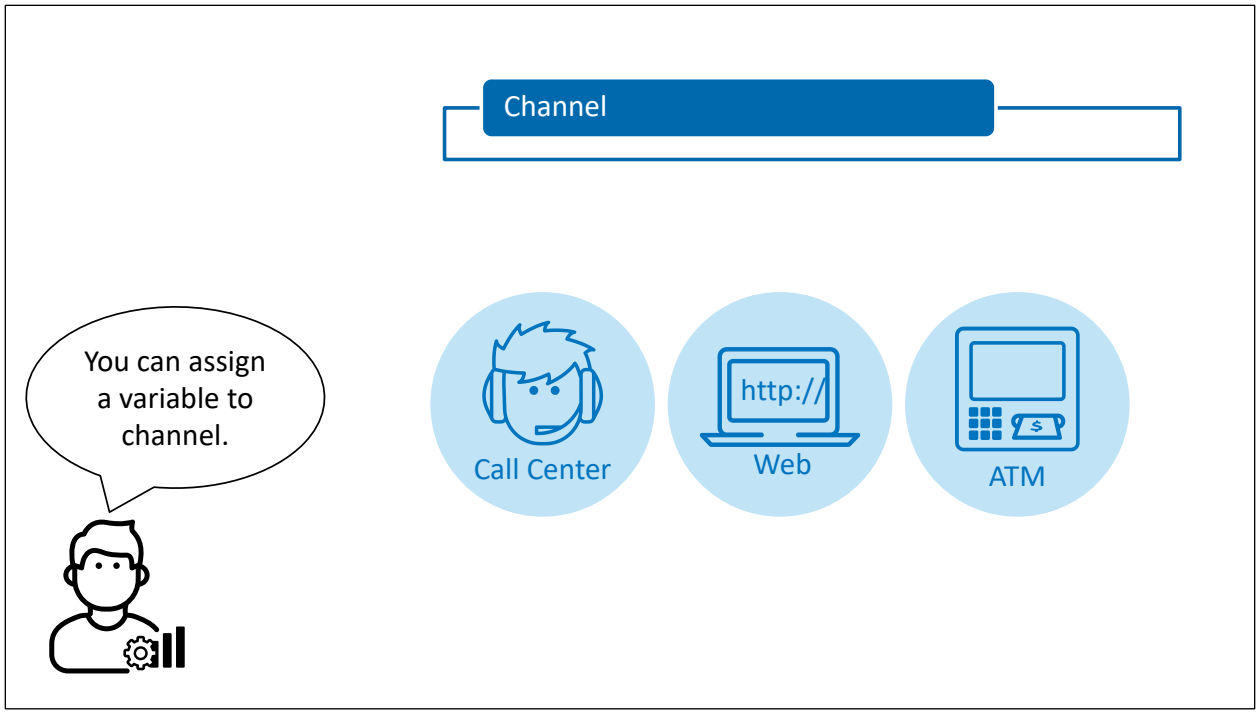


You use a record contacts node to record the values of specific variables at a specific point in a decision. You can include multiple record contacts nodes in a decision. For example, if your decision includes multiple paths, you might want to include a record contacts node on each path. You can include a record contacts node anywhere in a decision flow. However, you might typically want to include this node toward the end of your business logic processing.

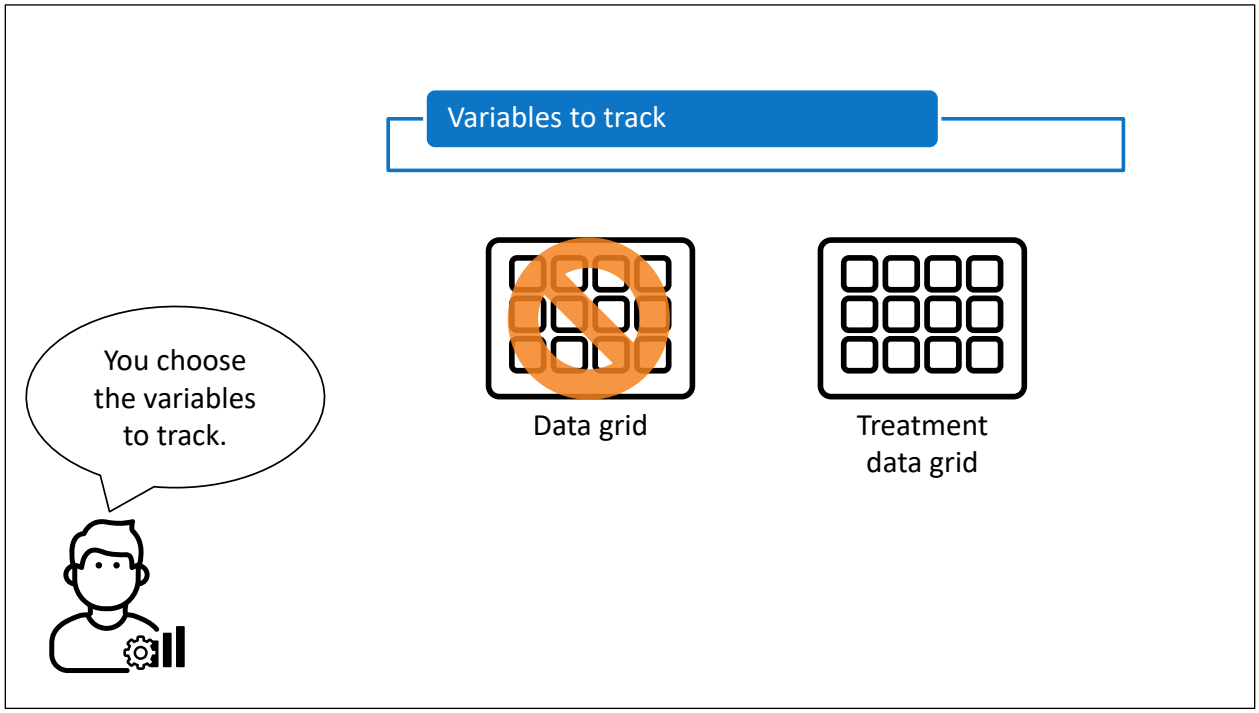


When you configure a record contacts node, you specify the settings shown here.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

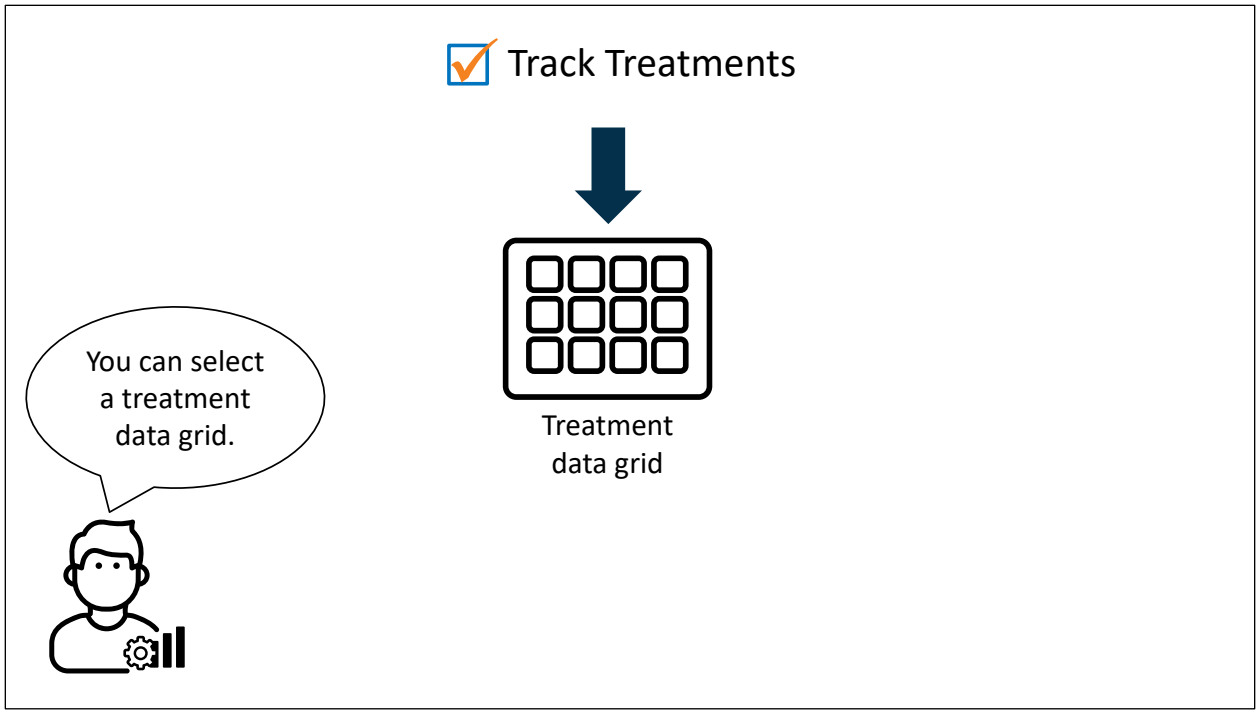


If the decision flow includes a variable that corresponds to the channel that the decision is executing in, you can assign this variable to channel. For example, you might want to record whether the decision is executing in the call center, web, or ATM channel.

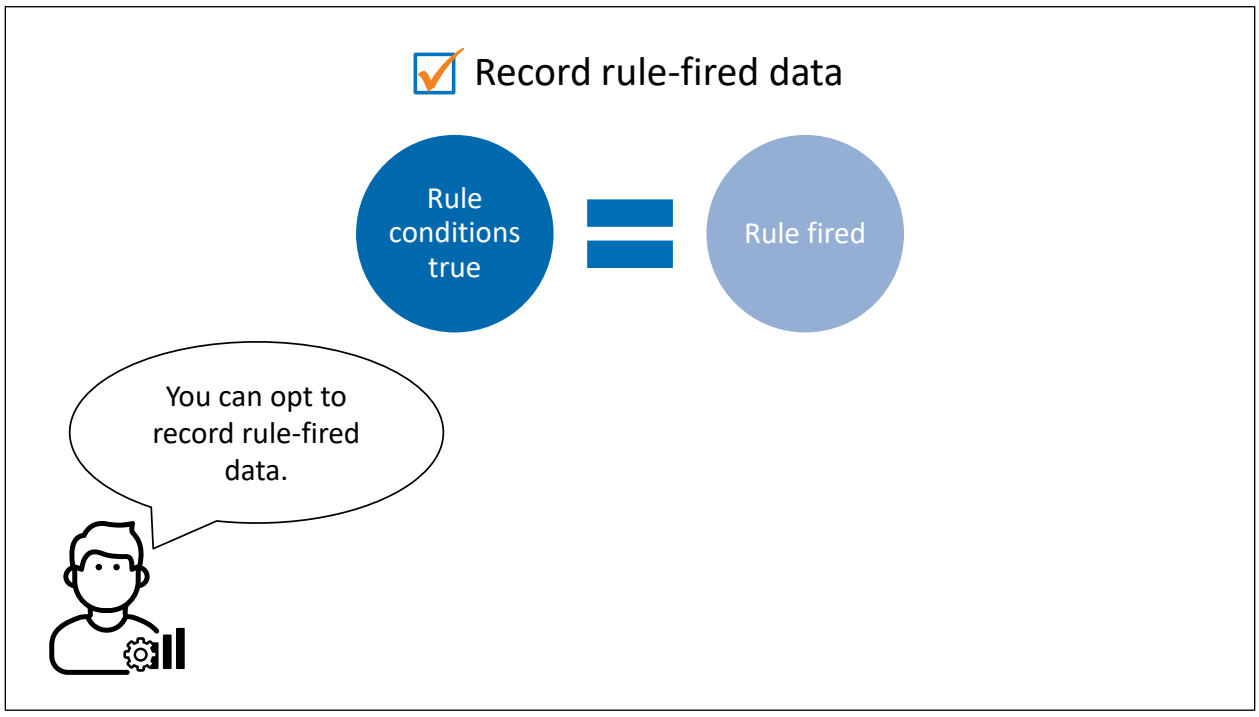


You choose the variables to track. You cannot select data grid variables (except for data grids corresponding to treatment groups) because of the amount of space required to store the data. The details of data grids and treatments are discussed later.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

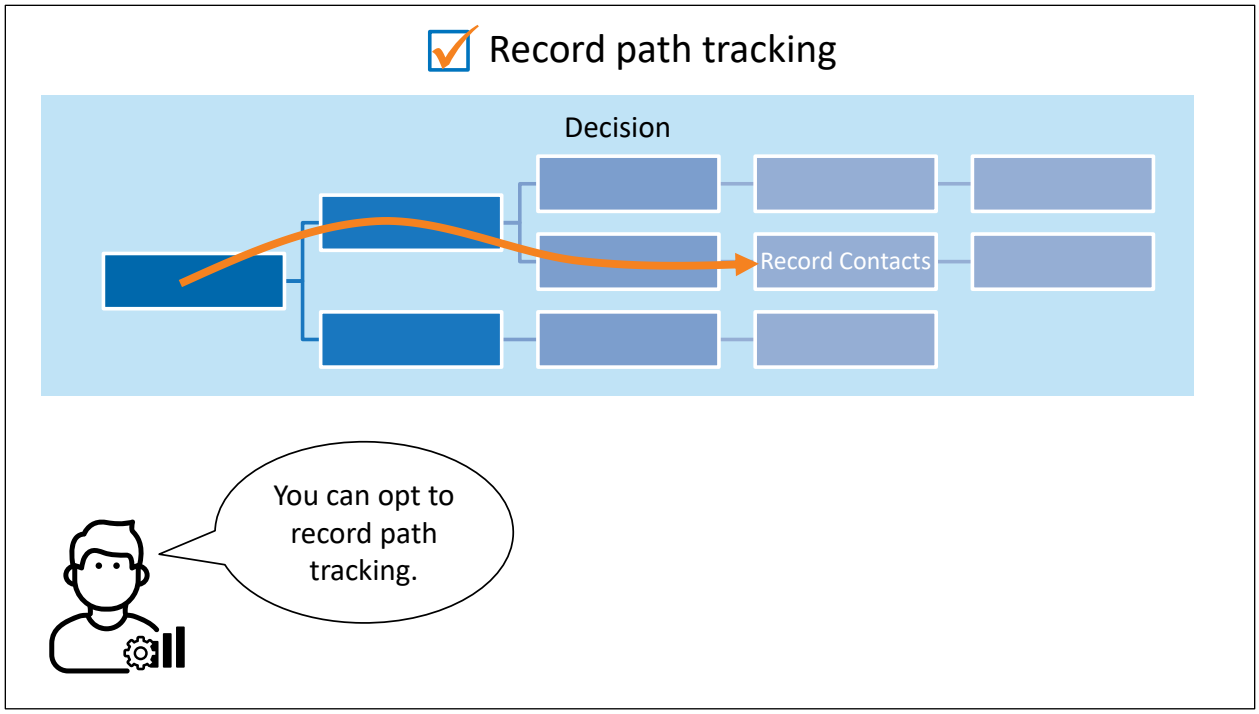


If you choose to track treatments, you select the treatment data grid for the set of treatments that you want to track.

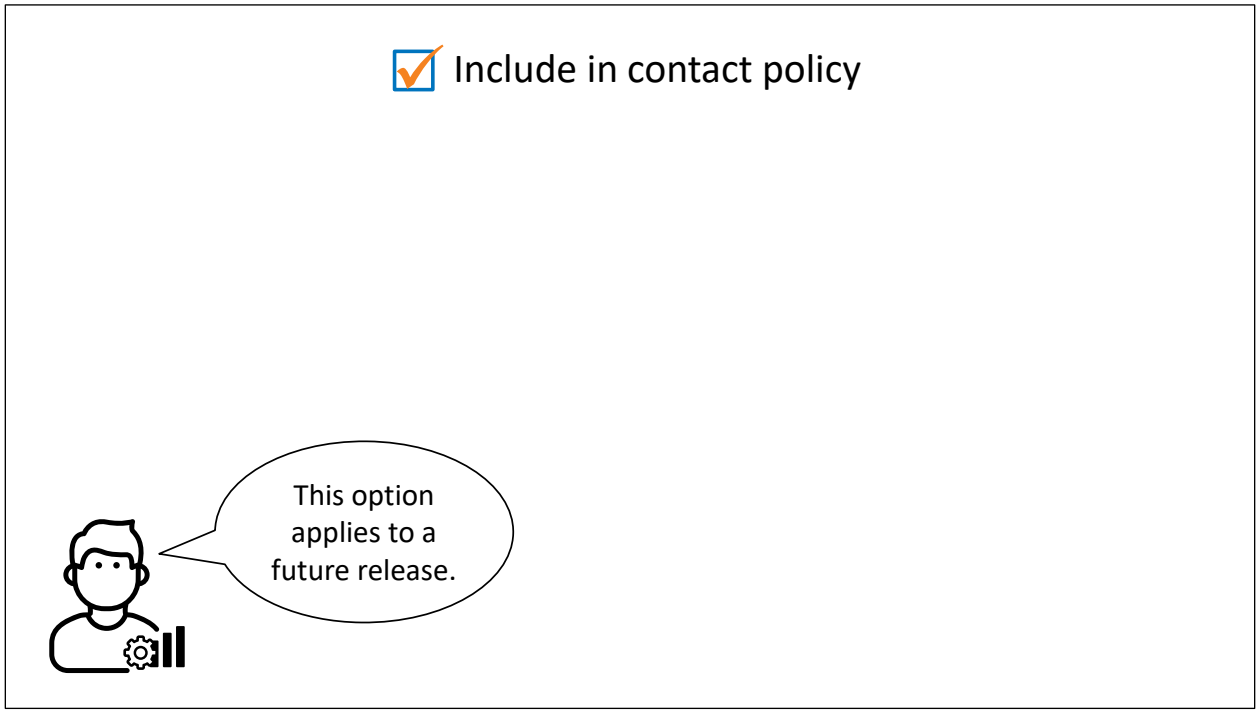


You can opt to record rule-fired data. If a rule's conditions evaluate to true, then the rule is said to have *fired*. When you enable this option, SAS Intelligent Decisioning records information about rules that fired in the decision flow prior to the record contacts node. If your decision calls another decision, rule-fired data for that decision is not recorded.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

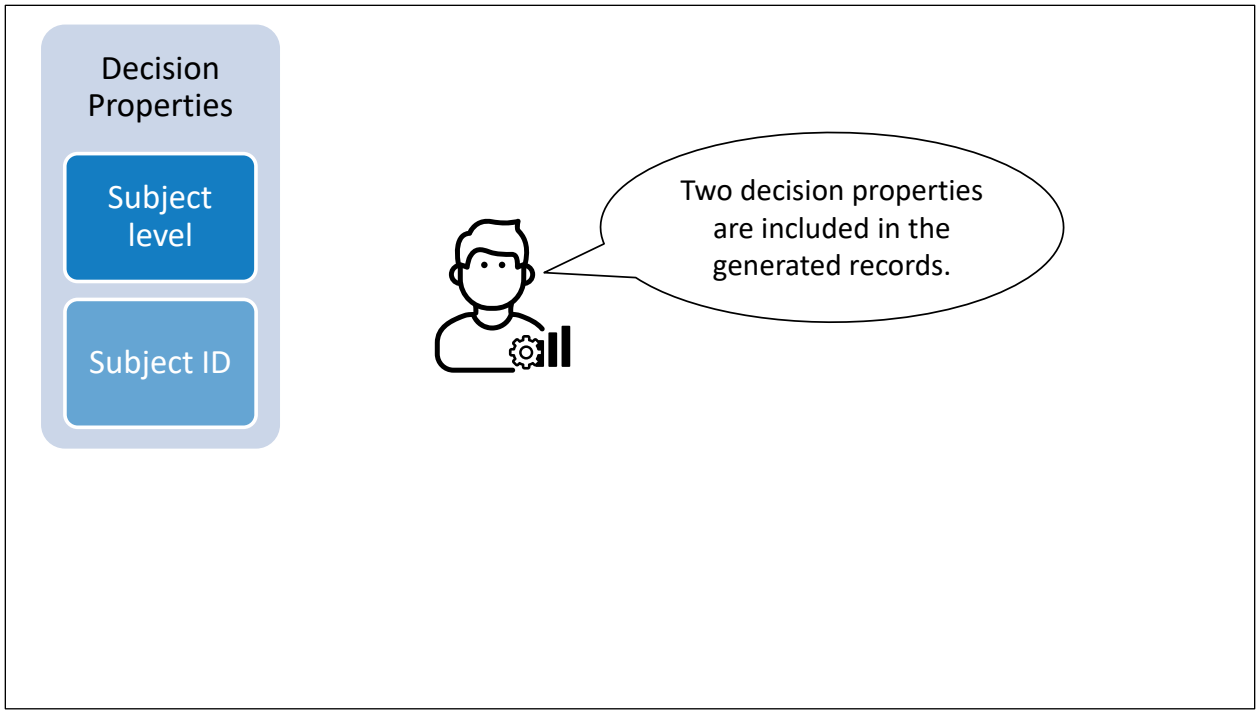


You can choose to record path tracking. When you enable this option, SAS Intelligent Decisioning records information about the path taken in the decision flow prior to the record contacts node. If your decision calls another decision, path tracking data for that decision is not recorded.

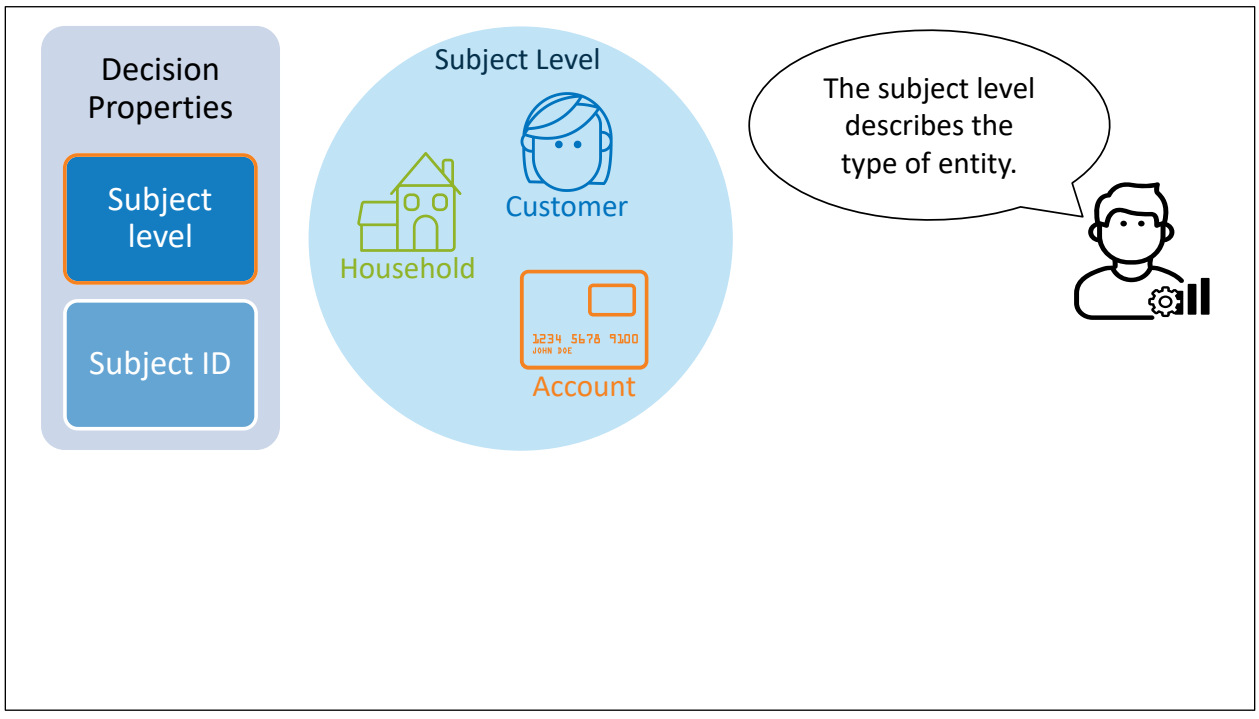


You can enable the **Include in contact policy** option. In a future release of SAS Intelligent Decisioning, when this option is enabled, you will be able to use the results in contact policy rules to limit the treatments that are returned by the decision based on prior interactions.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

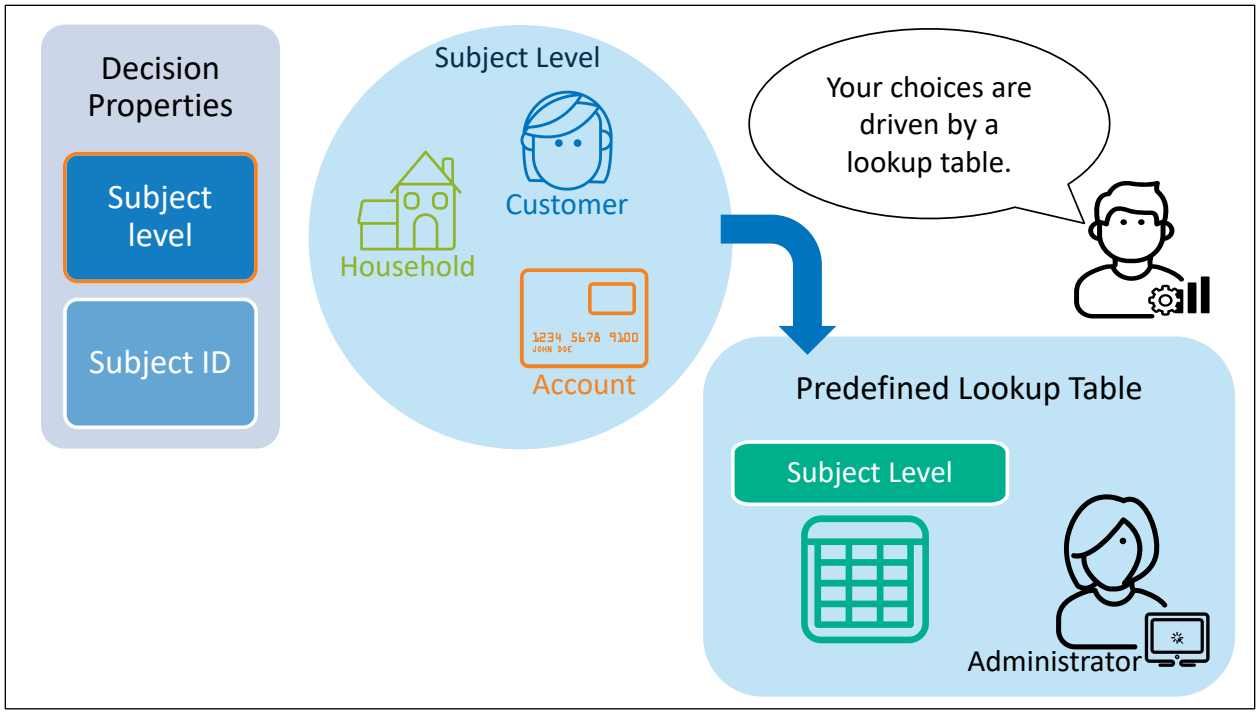


There are two properties of the decision that are included in the records that are generated for the record contacts node: subject level and subject ID.

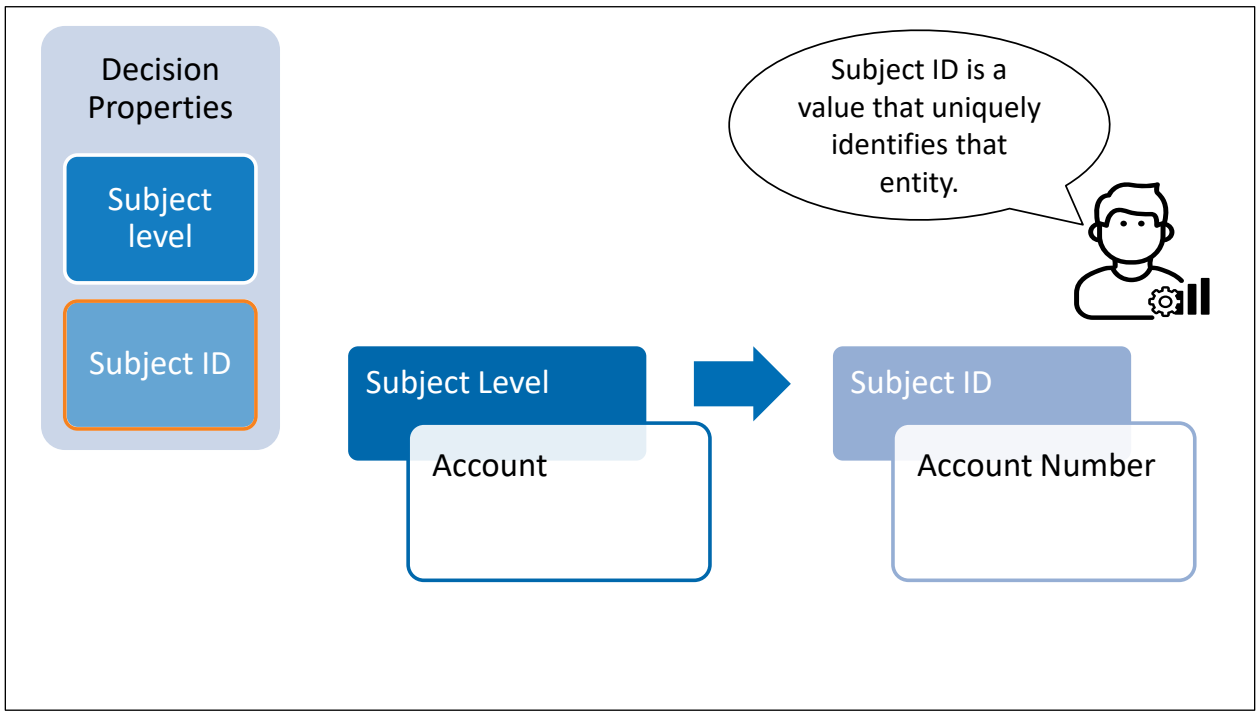


The subject level describes the type of entity that the decision corresponds to, such as a customer, household, or account.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

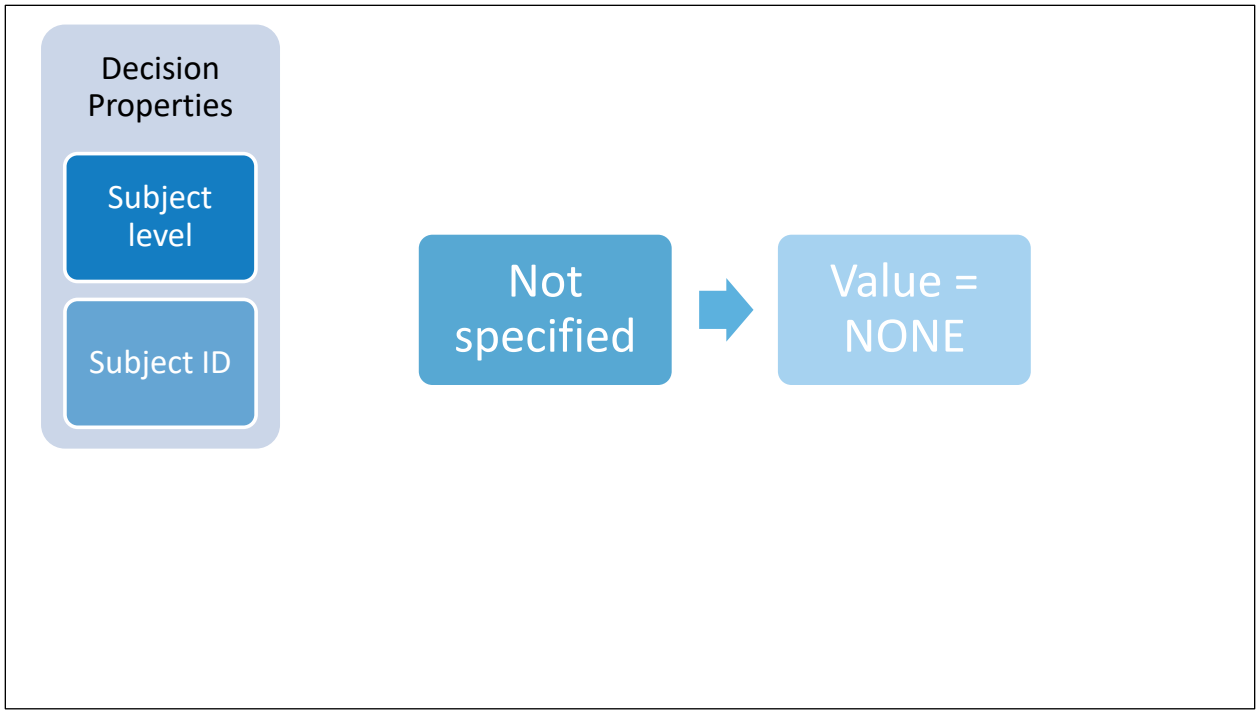


Your choices for subject level are driven by the Subject Level lookup table, which can be customized by the administrator.

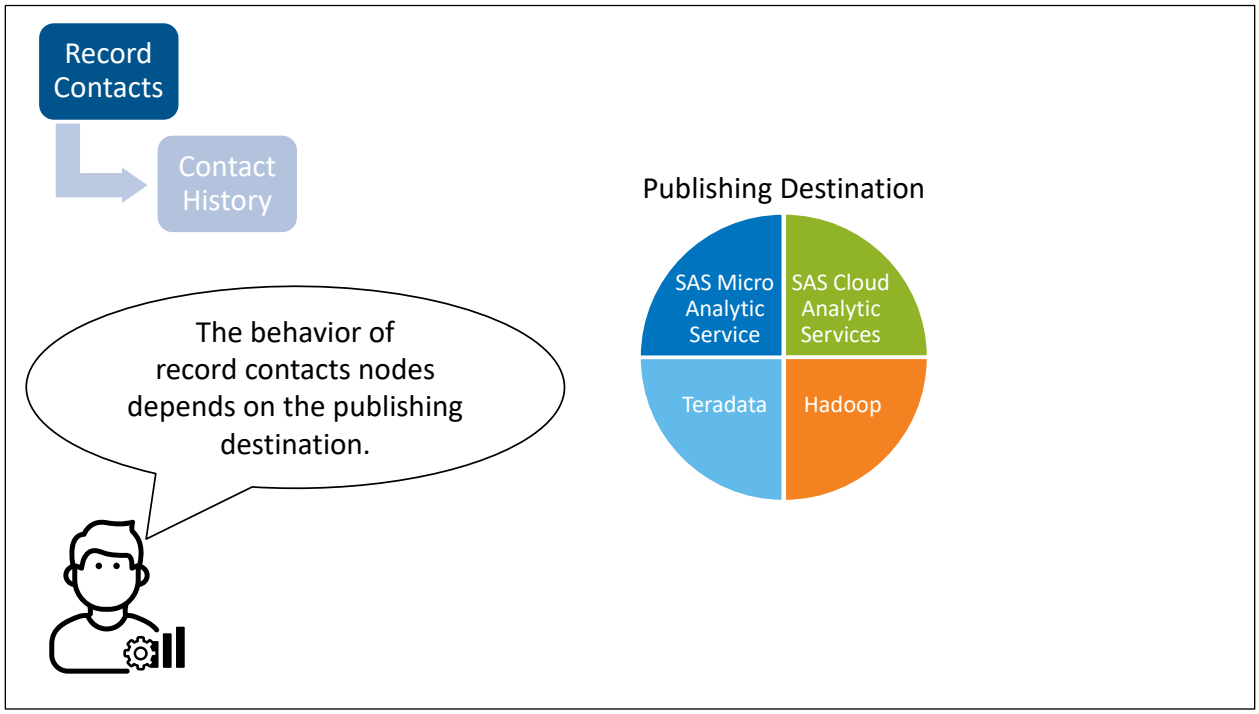


Subject ID is a value that uniquely identifies that entity. For example, if you choose an account as the subject level, you could choose the account number as the subject ID.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

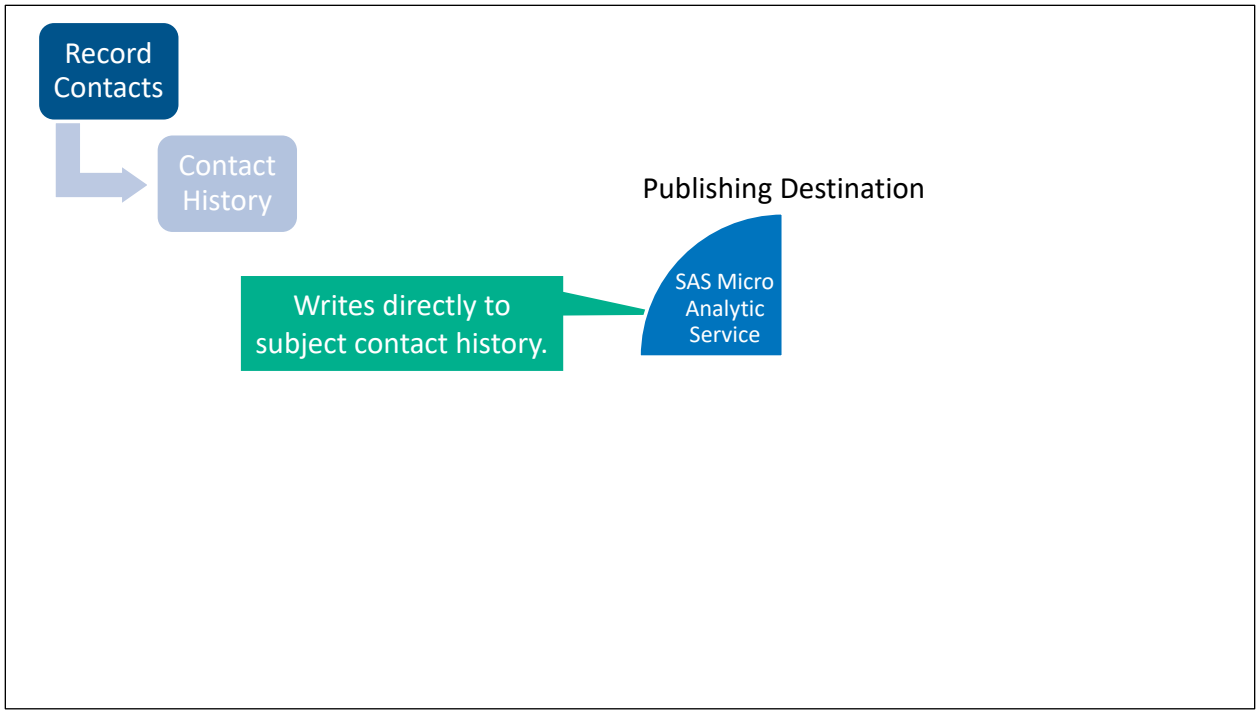


If you do not specify the subject ID or subject level, they are recorded as NONE.

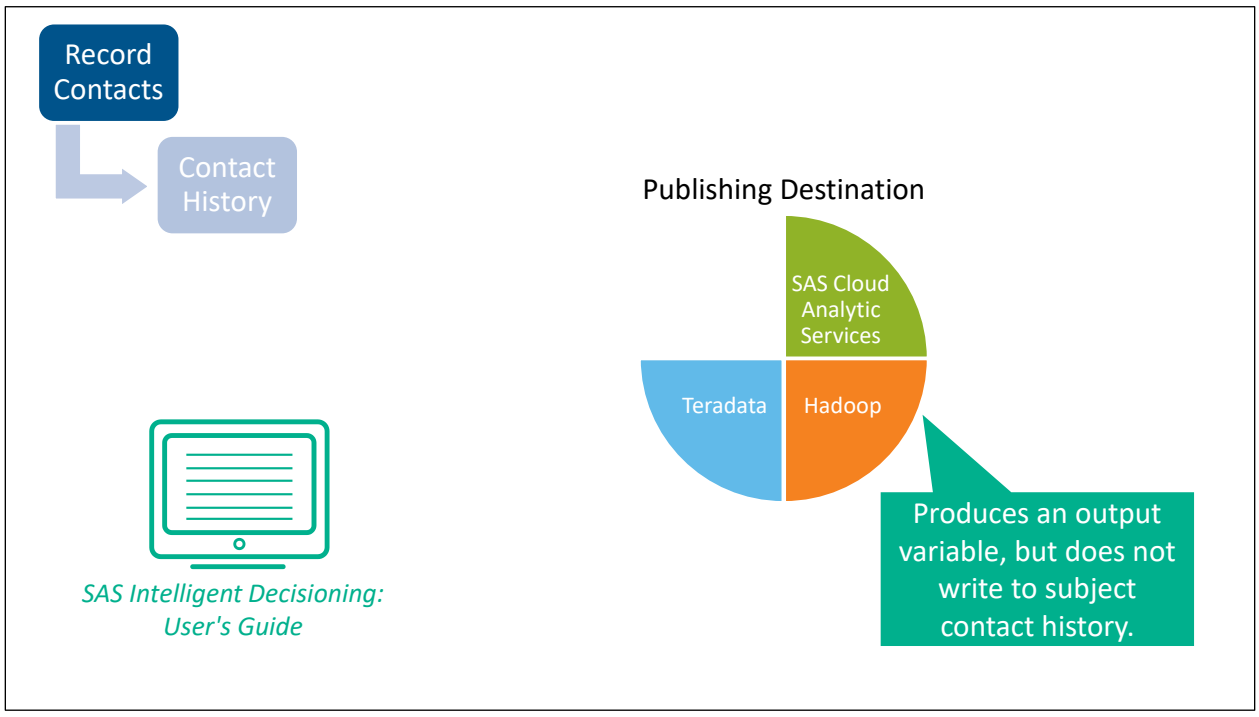


The behavior of record contacts nodes differs based on whether the decision is published to a SAS Micro Analytic Services destination or to a destination of another type.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



For decisions that are published to a SAS Micro Analytic Services destination, a record contacts node writes a record to the subject contact history.



When publishing to CAS, Hadoop, or Teradata, this process produces an output variable, but it does not write to subject contact history. You must create a separate process to load the data. For more information, refer to *SAS Intelligent Decisioning: User's Guide*.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Configuring a Record Contacts Node

This demonstration illustrates configuring a record contacts node.

Copyright © SAS Institute Inc. All rights reserved.

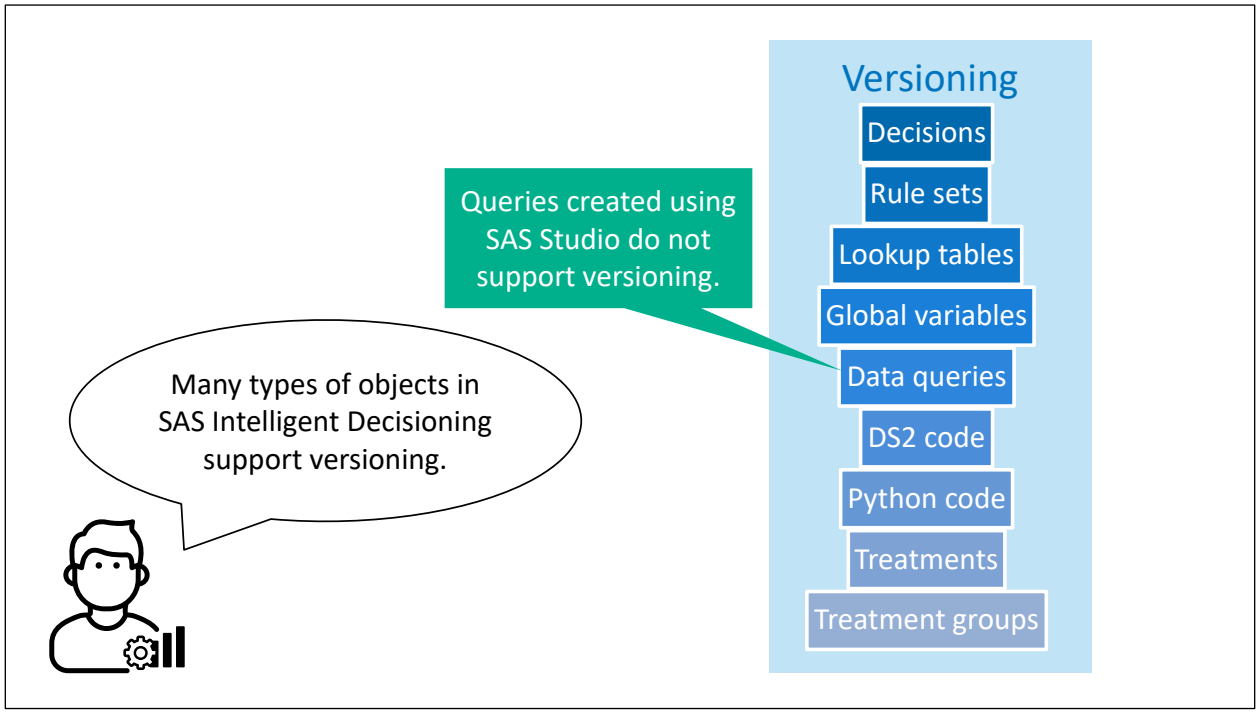
Practice

In this practice, you configure a record contacts node.

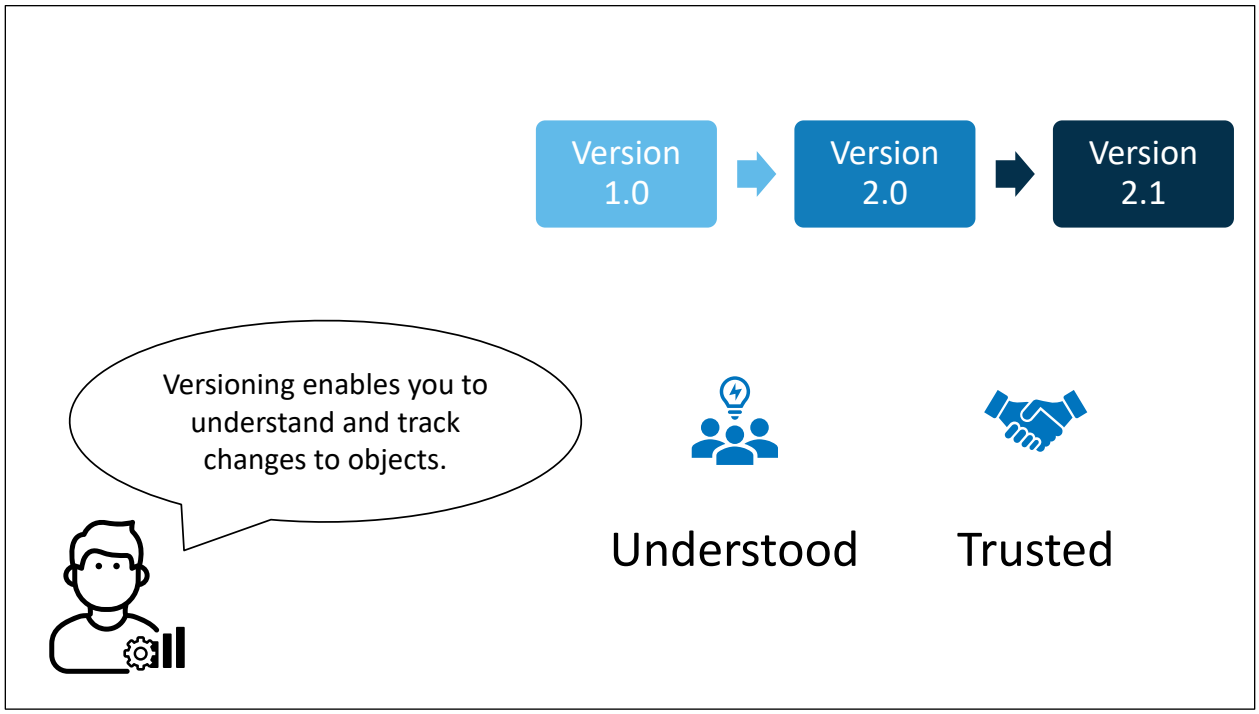
Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.6 Object Versioning

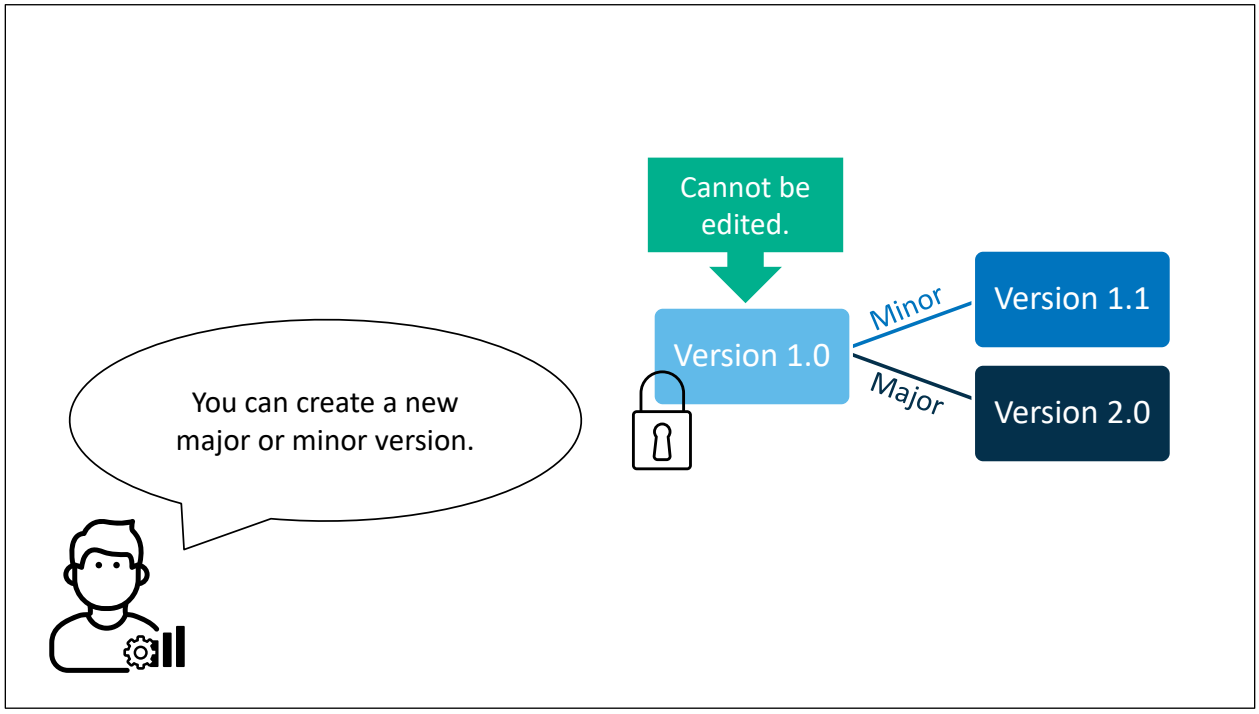


Many types of objects in SAS Intelligent Decisioning support versioning. The specifics of what is supported depends on the object type. Versioning is supported for custom code files with the exception of data queries that were created using SAS Studio.

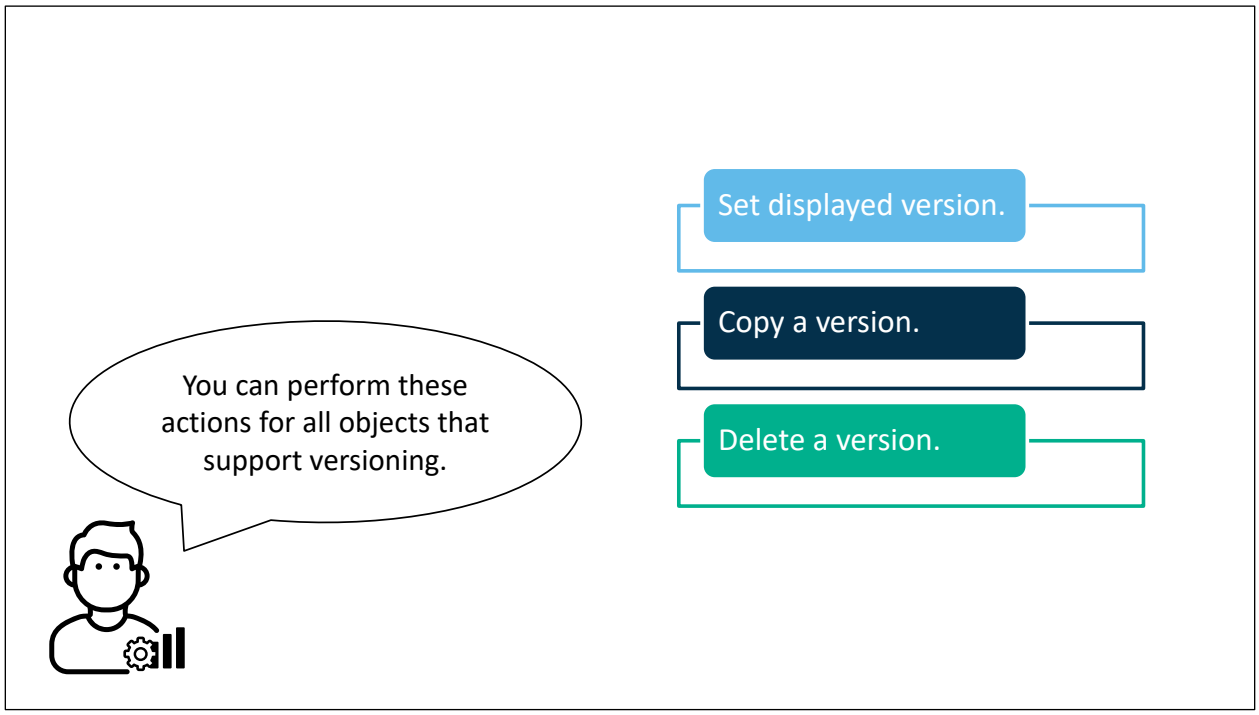


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Versioning enables you to understand and track changes to objects, providing governance to ensure that your decisions are understood and trusted across your organization.

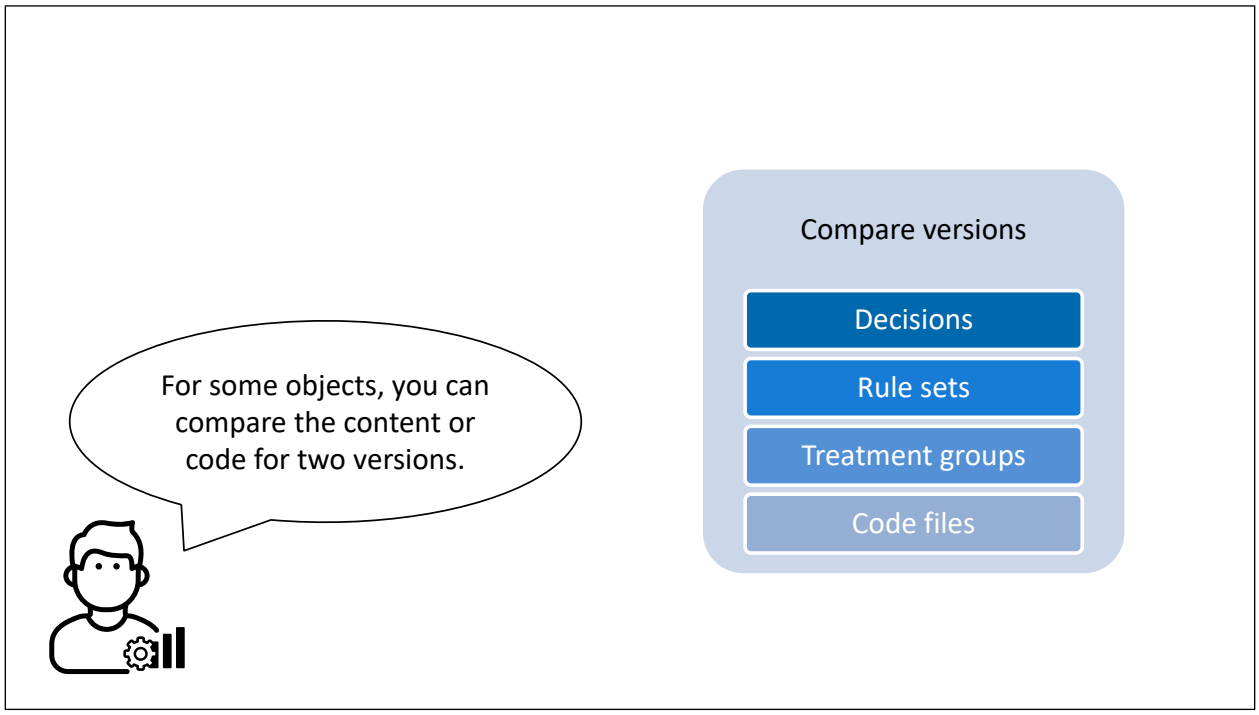


When you create a new version of an object, you specify whether the revision is Major or Minor. When you create a new version, the previous version is automatically locked and cannot be edited.

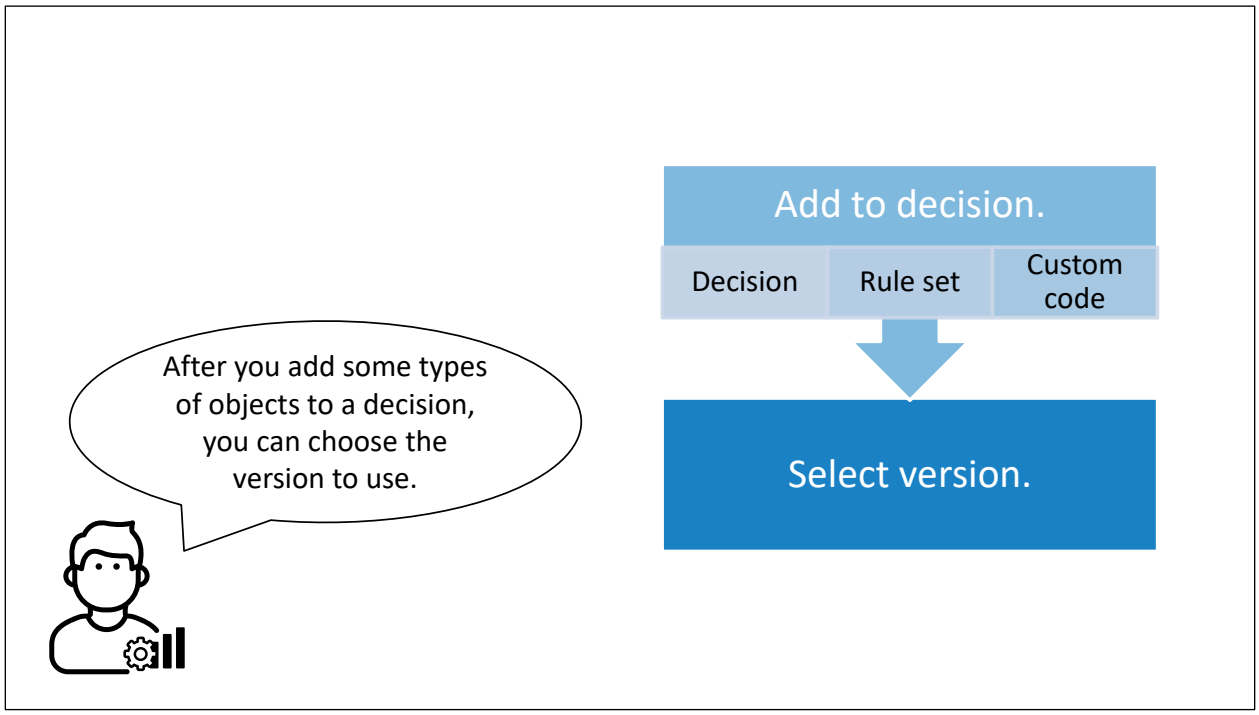


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

For objects with multiple versions, you can select the version that is active in the user interface. You can copy from another version. You can also delete a version, but only if you have permissions to delete the object itself.



For some objects, you can compare the content or code for two versions.



After you add a decision, rule set, or custom code file to a decision (with the exception of data queries created using SAS Studio), you can select the version to use.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Creating and Comparing Rule Set Versions

This demonstration illustrates creating a new version of a rule set and comparing versions.

Copyright © SAS Institute Inc. All rights reserved.

Practice

In this practice, you compare versions of a decision.

Copyright © SAS Institute Inc. All rights reserved.

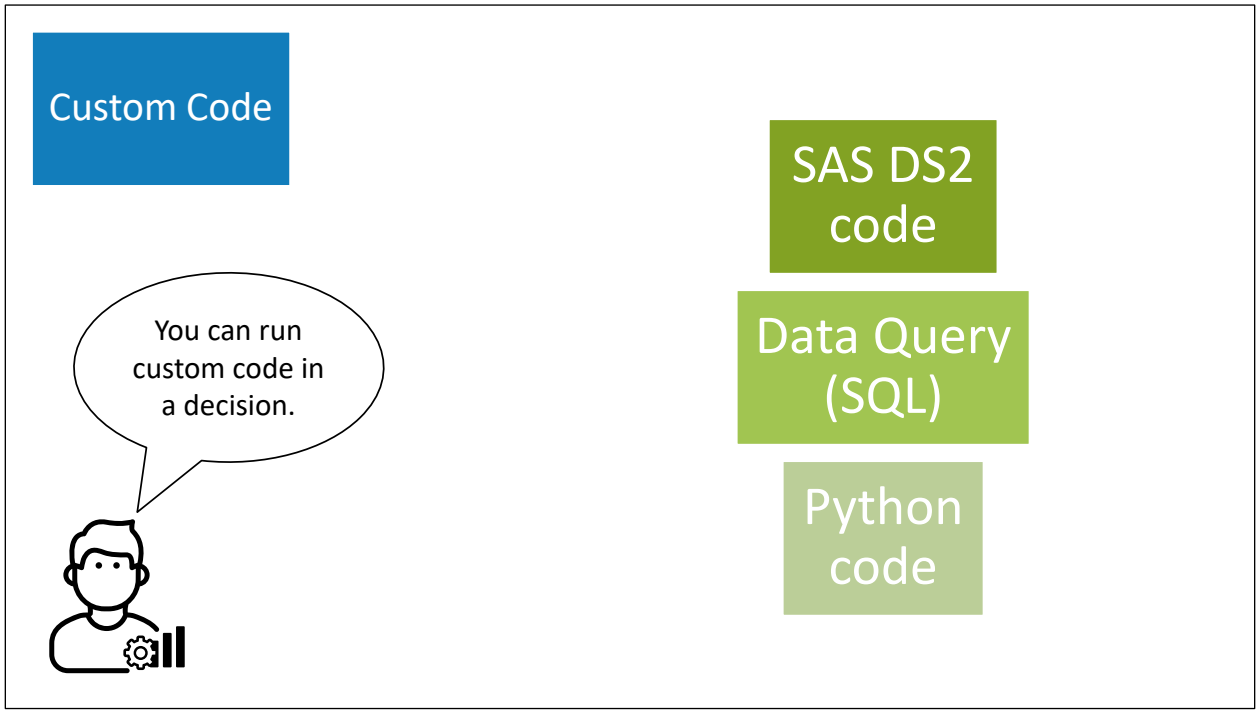
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

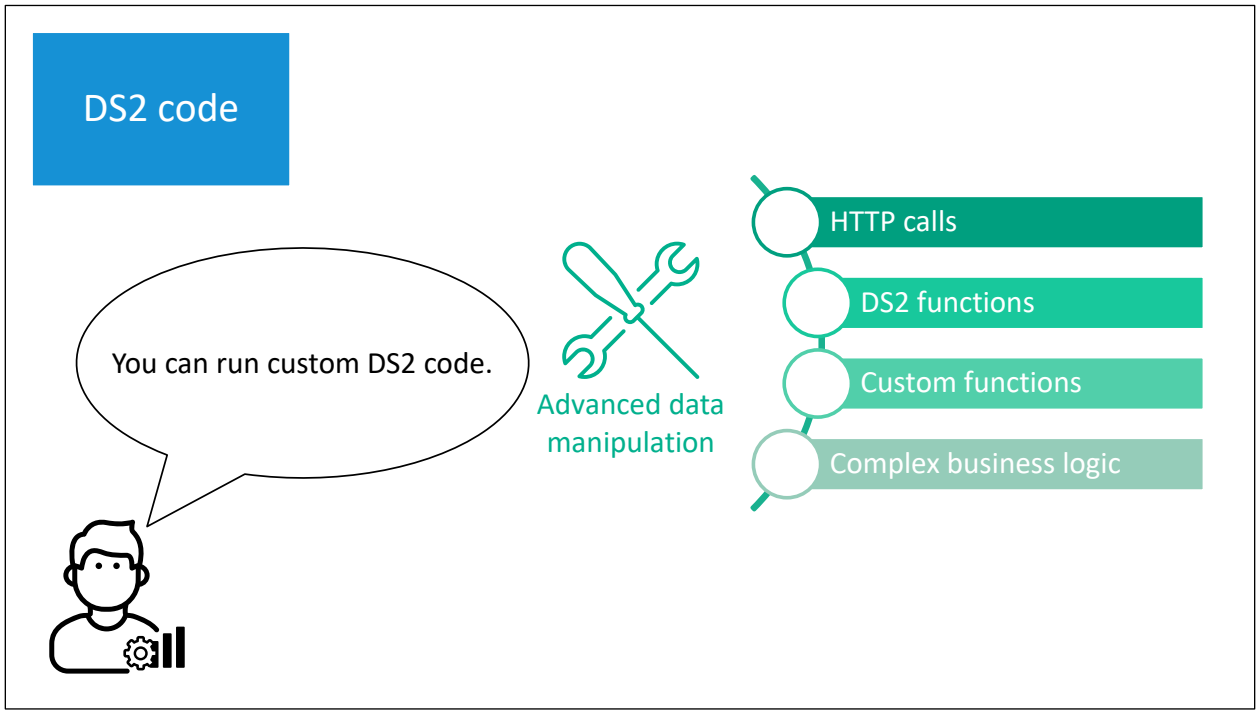
Lesson 3 Advanced Topics

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.1 Custom Code

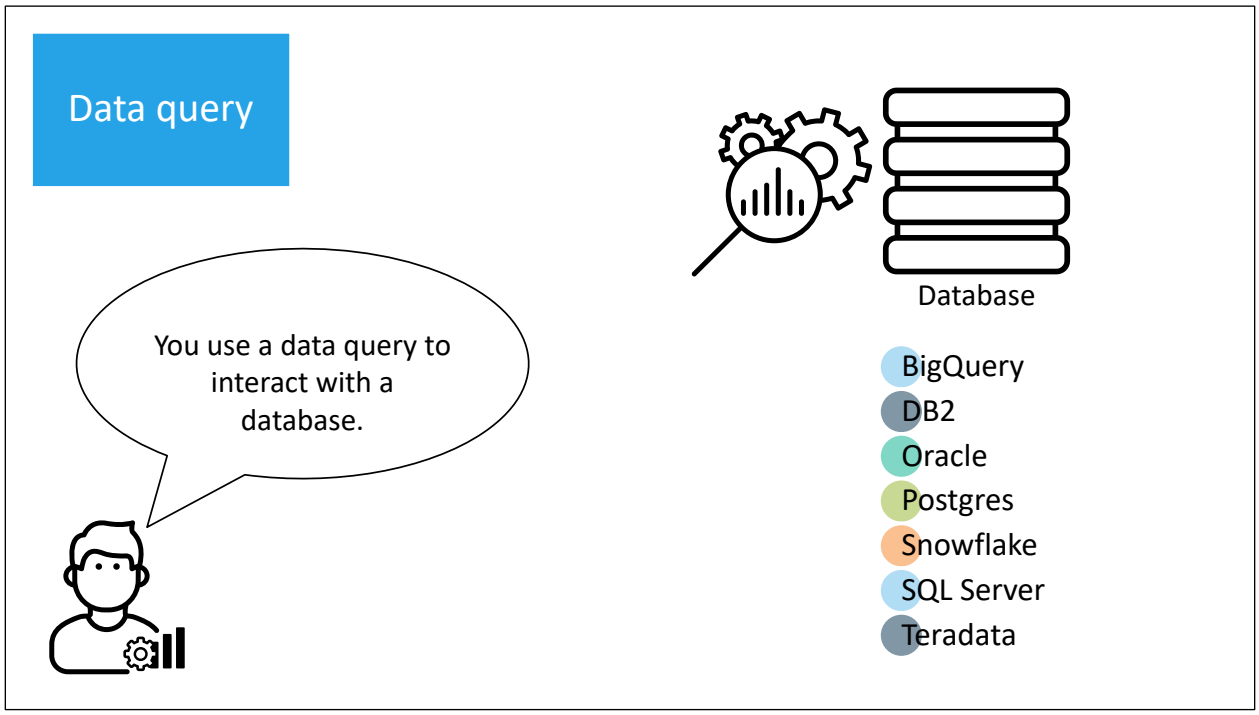


You can run custom code that was written using the SAS DS2 language, SQL, or the Python language.

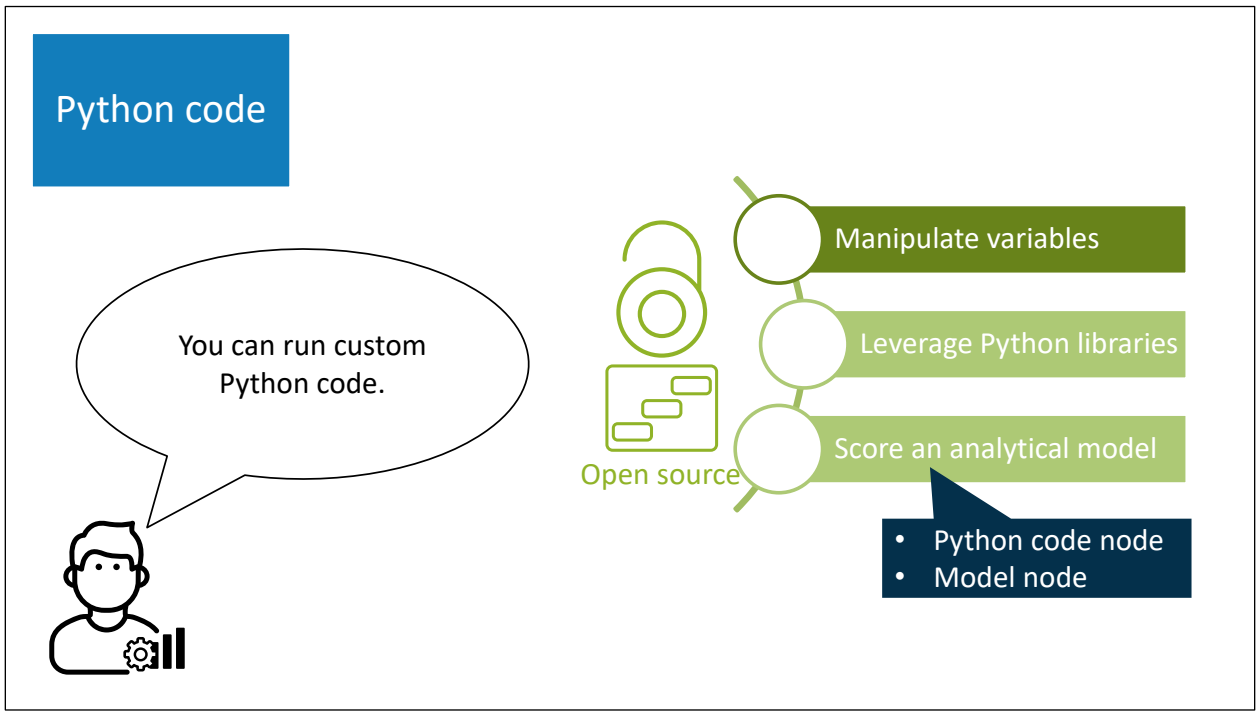


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

DS2 is a SAS proprietary programming language that is appropriate for advanced data manipulation. In a DS2 code node, you could make HTTP calls to REST APIs, or use DS2 or custom functions to manipulate decision variable values. You could also use DS2 code to implement complex business logic that might be difficult to implement with other decision nodes.

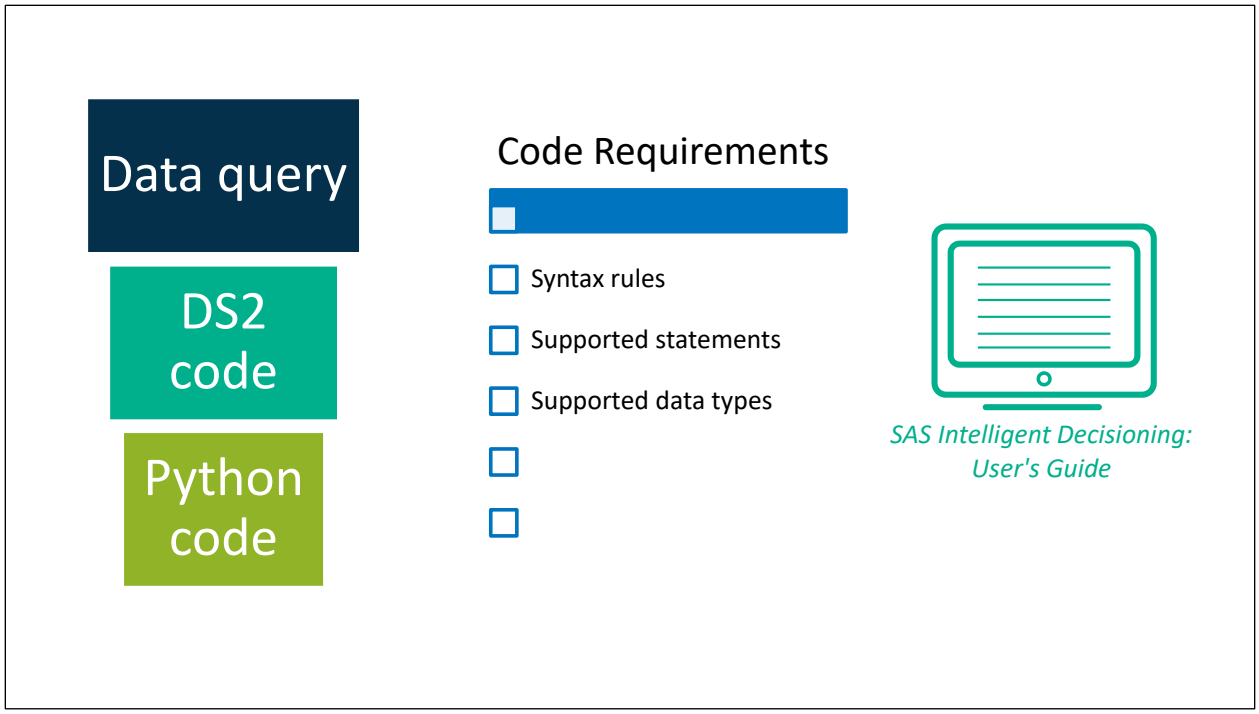


You can use a data query to interact with supported databases including BigQuery, DB2, and Oracle using SQL.



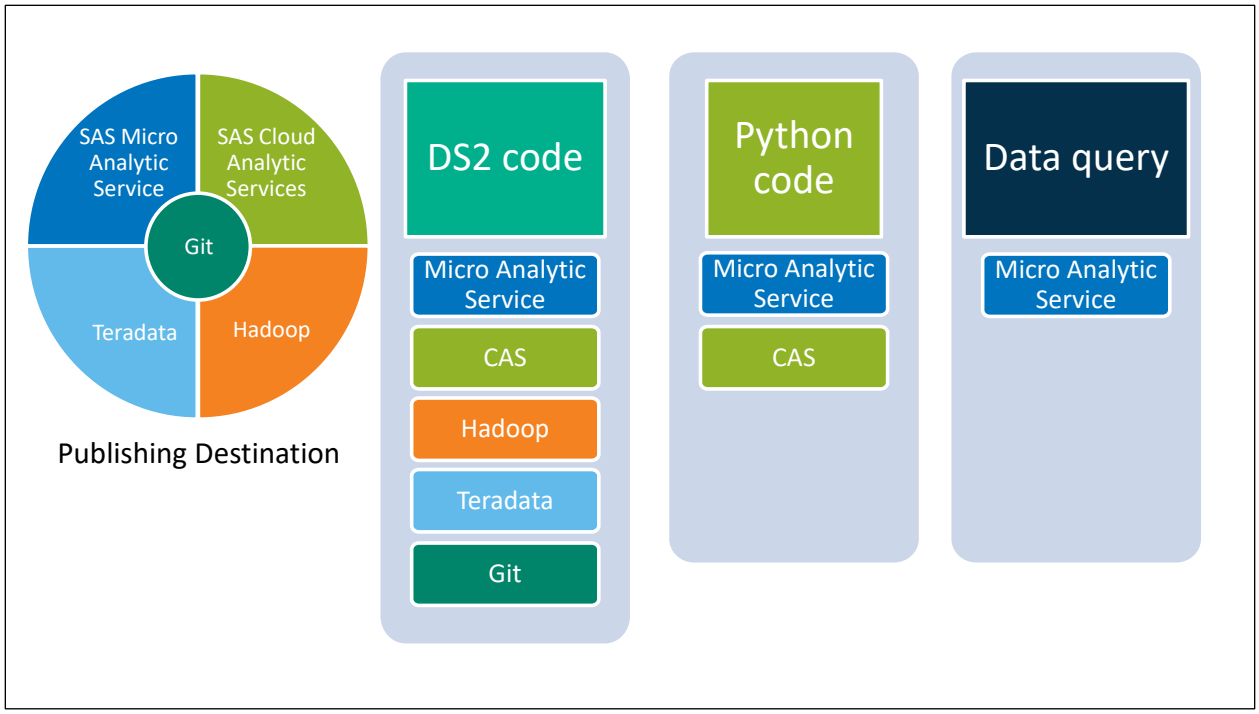
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Python is a popular open source language. You could use a Python code node to manipulate decision variable values or to create business logic that leverages Python libraries. You could also use Python code to produce a score for an analytical model. In this situation, you might alternatively choose to use a Model node. Extra steps are required to use a Model node, but these steps might be worthwhile if you are surfacing the Python model to multiple SAS applications.



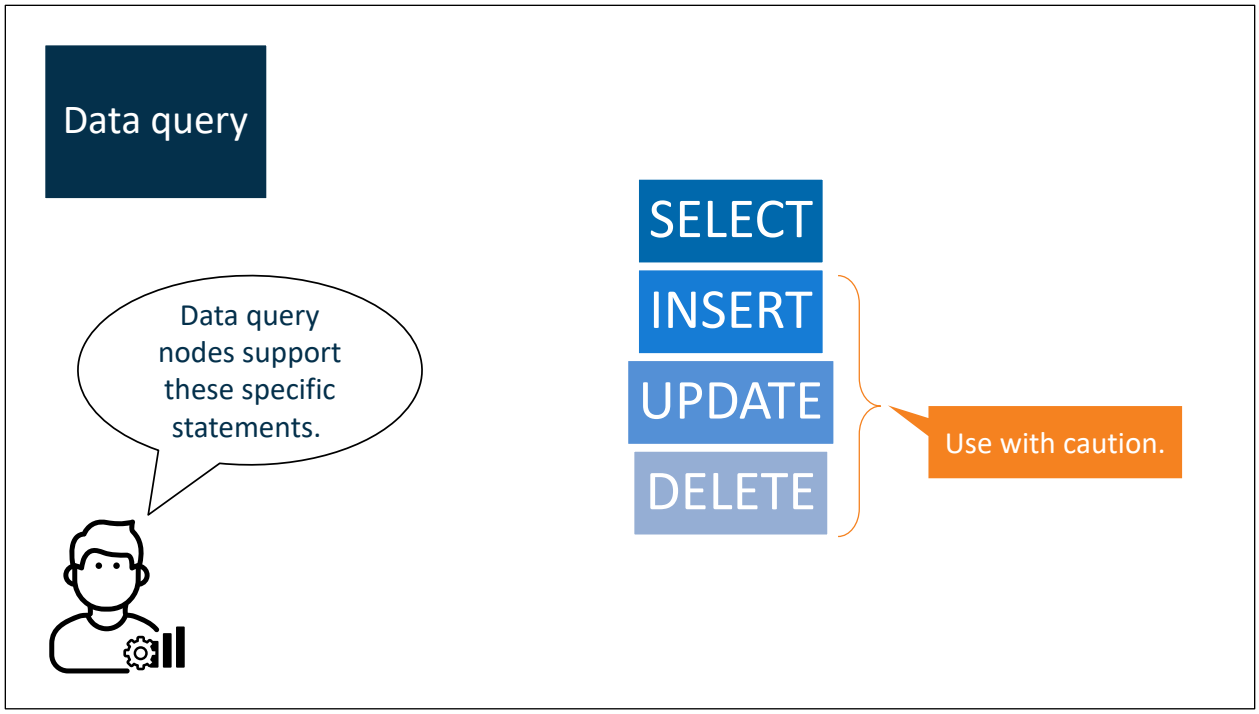
When you write custom code for use in Intelligent Decisioning, there are specific requirements for each of the three types of code. There might be specific syntax rules or limitations on support for certain statements or data types. Refer to *SAS Intelligent Decisioning: User's Guide* to understand the requirements for each type of code.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

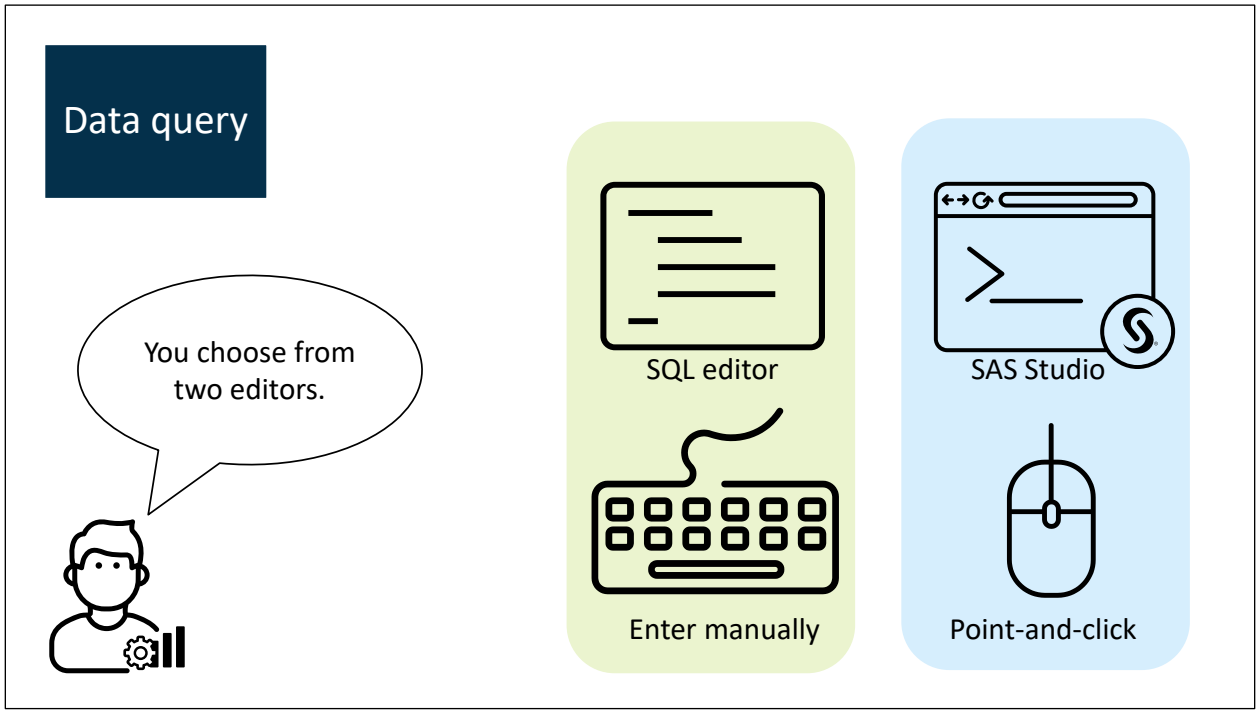


SAS DS2 code is supported for all publishing destinations. You can publish decisions that contain Python code files to SAS Micro Analytic Service and SAS Cloud Analytic Services (CAS) destinations only. You can publish decisions that include data queries to SAS Micro Analytic Service destinations only.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

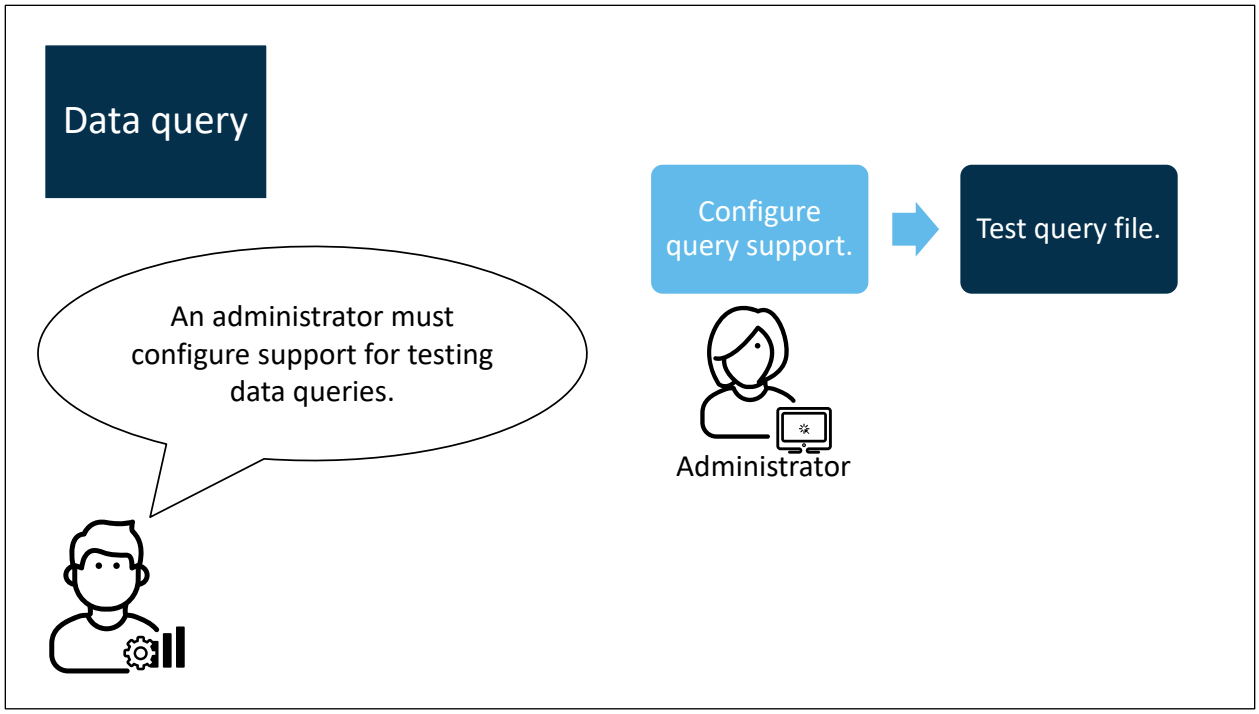


Data query nodes support SELECT, INSERT, UPDATE, and DELETE statements. They do not support any data definition language (DDL) statements such as ALTER or DROP that alter the structure of the table. You should use INSERT, UPDATE, and DELETE with caution to protect performance.

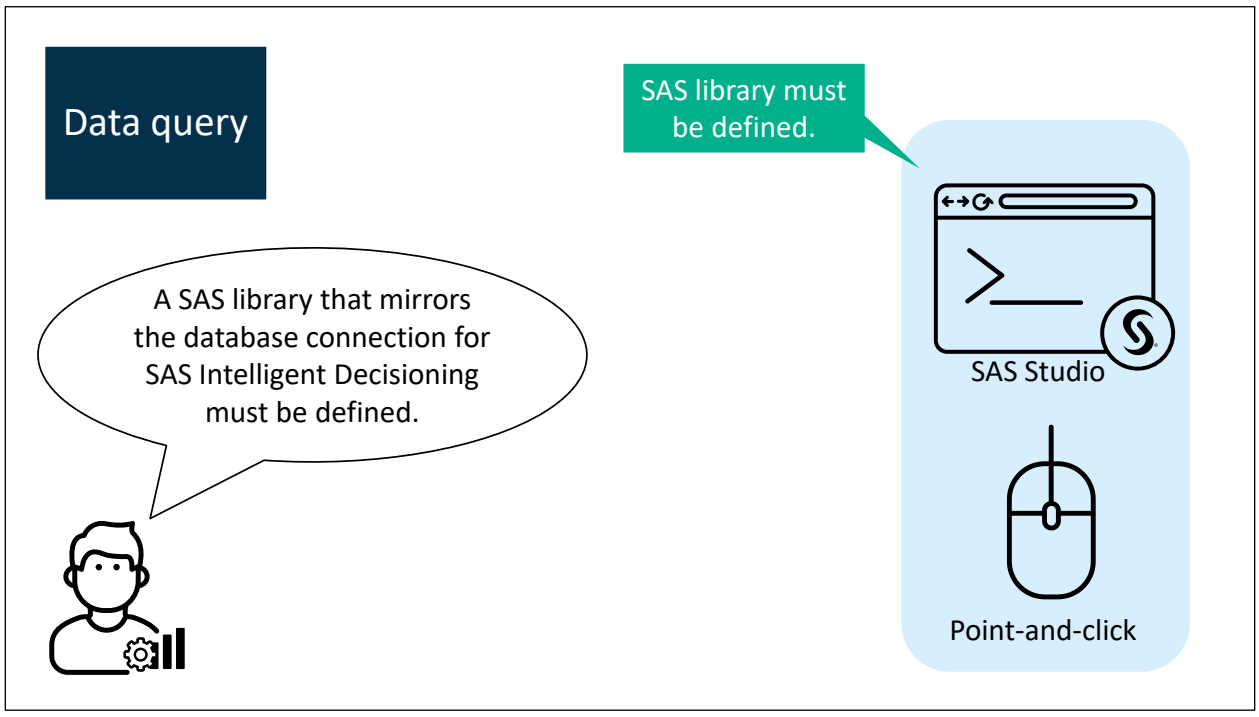


When you create a new data query in SAS Intelligent Decisioning, you can use the SQL editor or SAS Studio. When you use the SQL editor, you enter the query code manually. SAS Studio provides a visual point-and-click interface that you can use to generate the query code.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

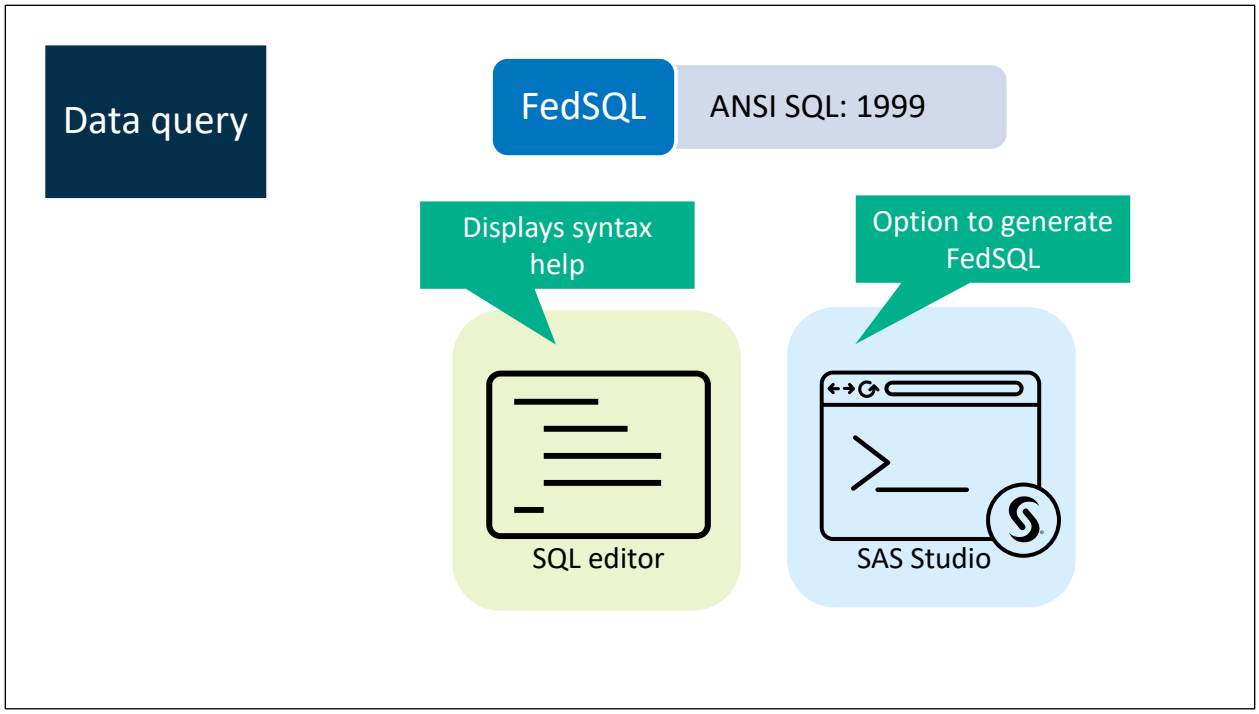


In order to test data queries and decisions that contain them, an administrator must configure support for SQL query files. This configuration is performed in SAS Environment Manager.

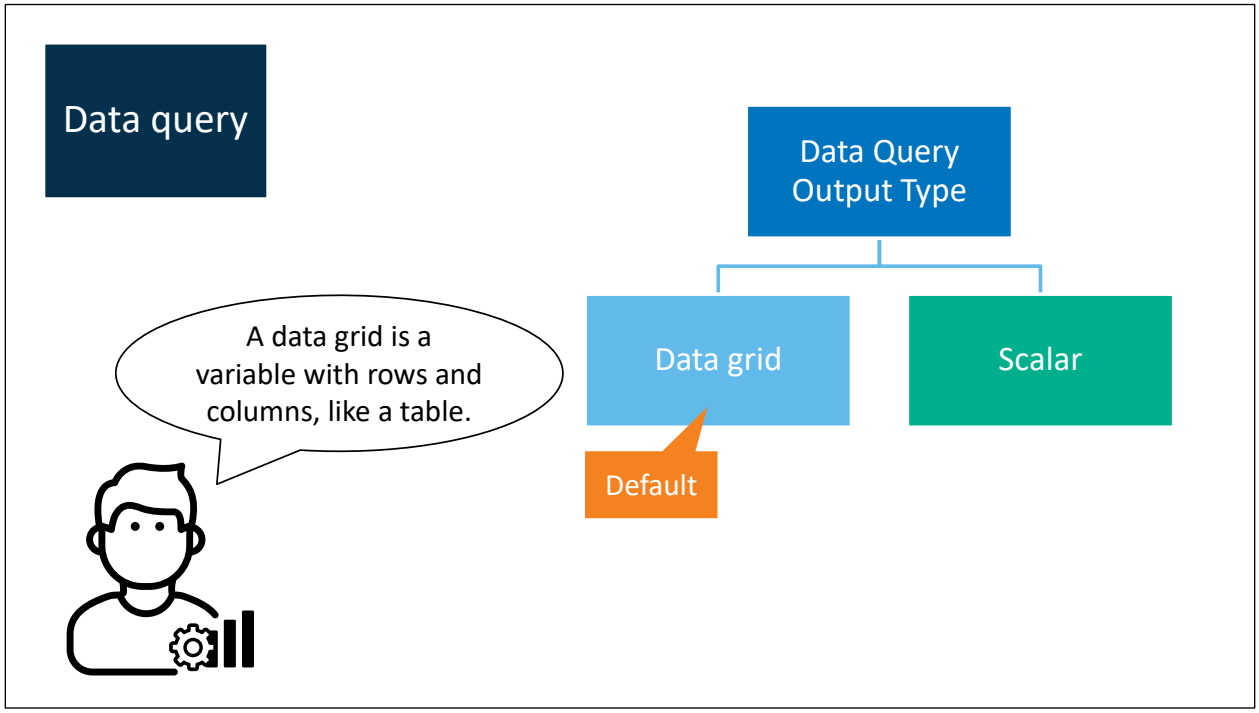


When you use SAS Studio to create a query for use in SAS Intelligent Decisioning, a SAS library must be defined that mirrors the database connection that was configured for use with the solution. SAS Studio does not need to reference the live data used by Intelligent Decisioning, but the table and column references must be consistent.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

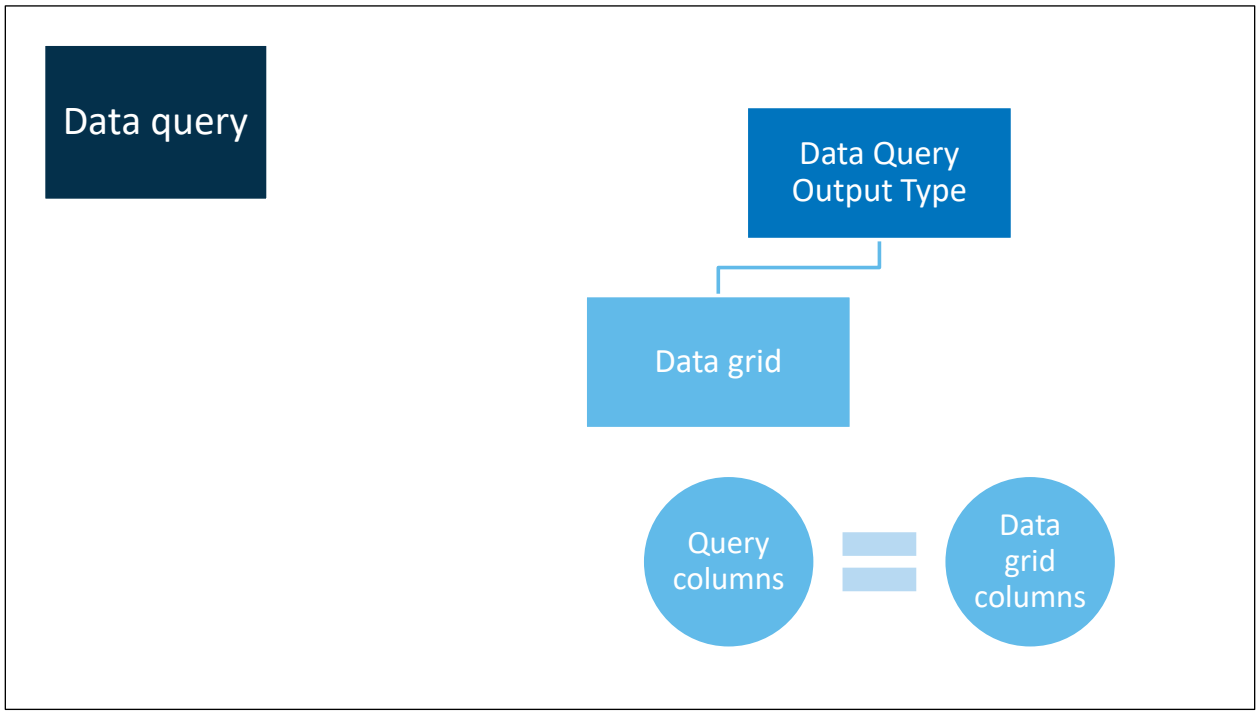


Data queries for SAS Intelligent Decisioning must be written in the FedSQL language. FedSQL is the SAS implementation of the ANSI SQL:1999 core standard. When you use the SQL editor, syntax help for FedSQL is displayed at the top of the editor window. You can also refer to *SAS FedSQL Language Reference*. When you use SAS Studio, you must enable an option to generate FedSQL. Be aware that some data repositories such as Postgres are case sensitive in terms of table and column names.

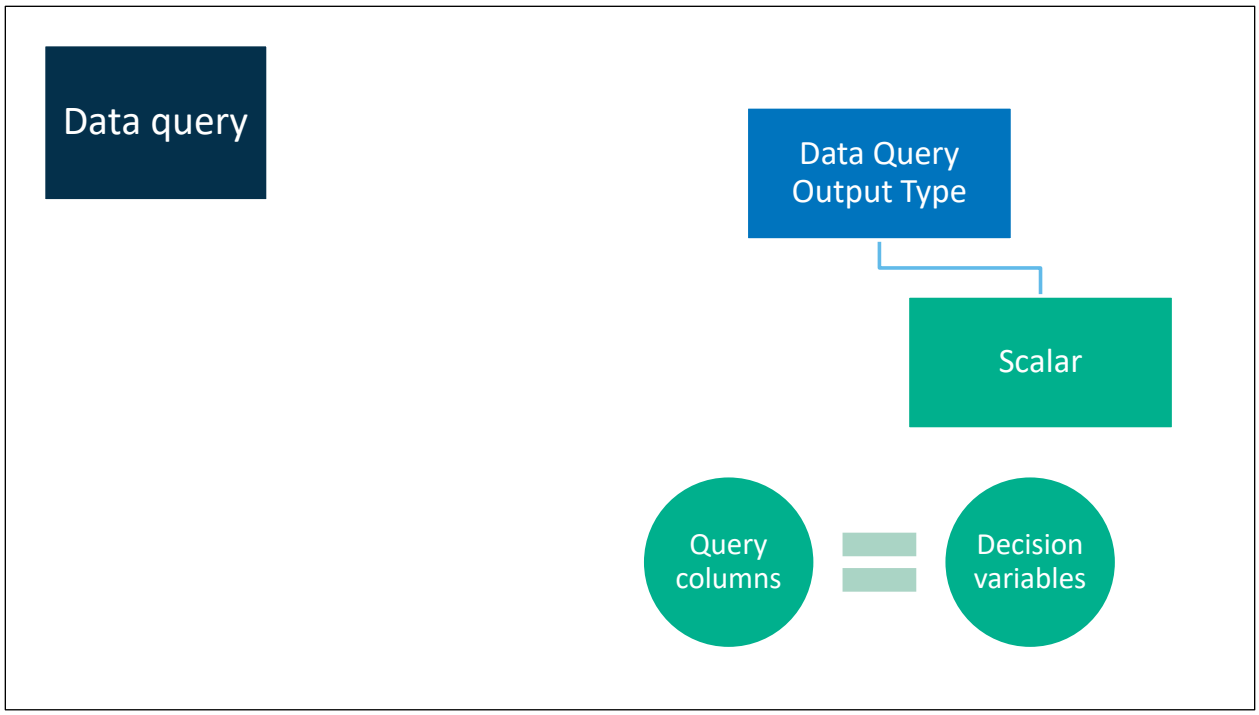


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

By default, a data query file returns a data grid. A data grid is a variable with rows and columns, like a table. However, you can specify that the query returns a data set with a single row of scalar variables.

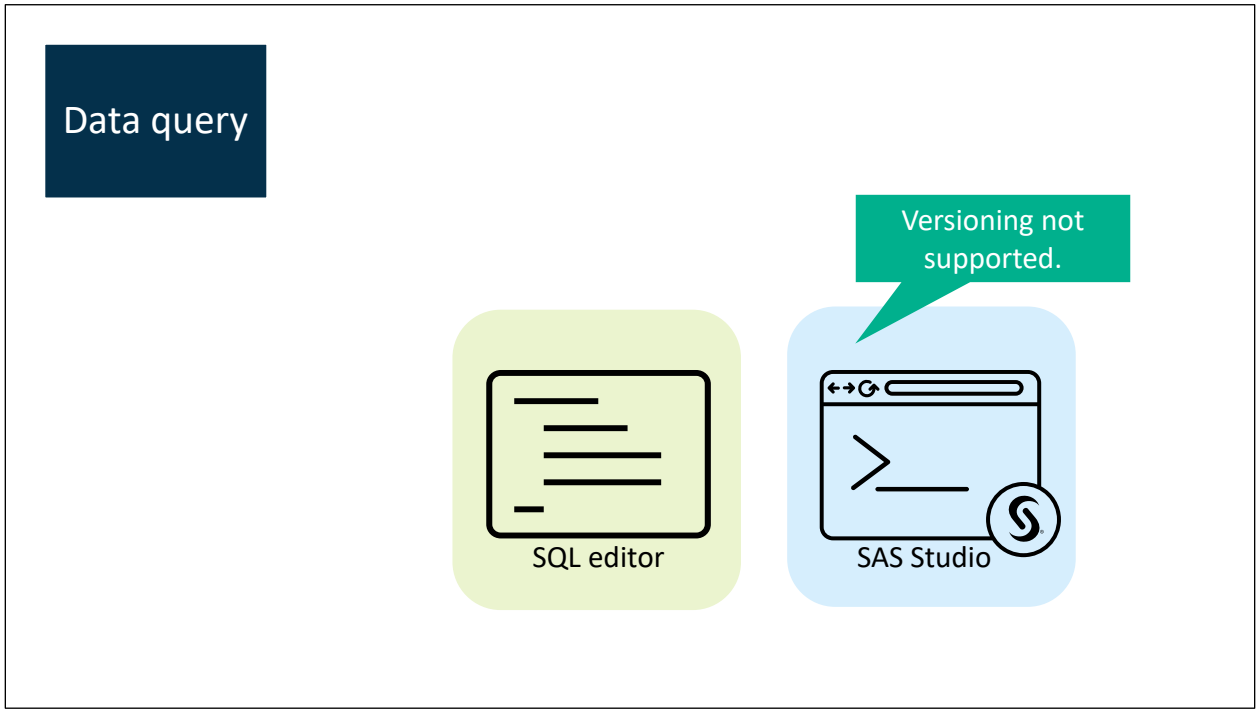


When a data query returns a data grid, the columns specified in the query are the columns of the data grid. The data grid can have multiple rows.

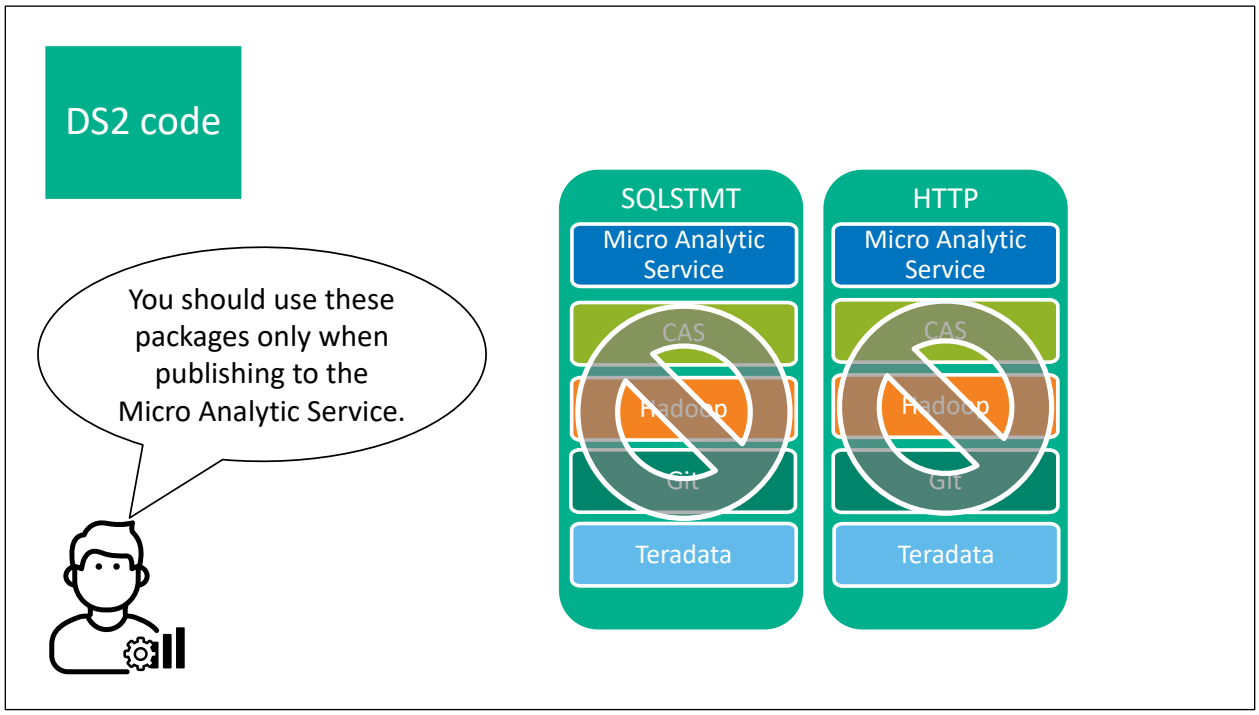


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

When a data query returns scalar variables, the columns specified in the query are decision variables of the same type. If the query matches multiple rows, only the first row is returned.



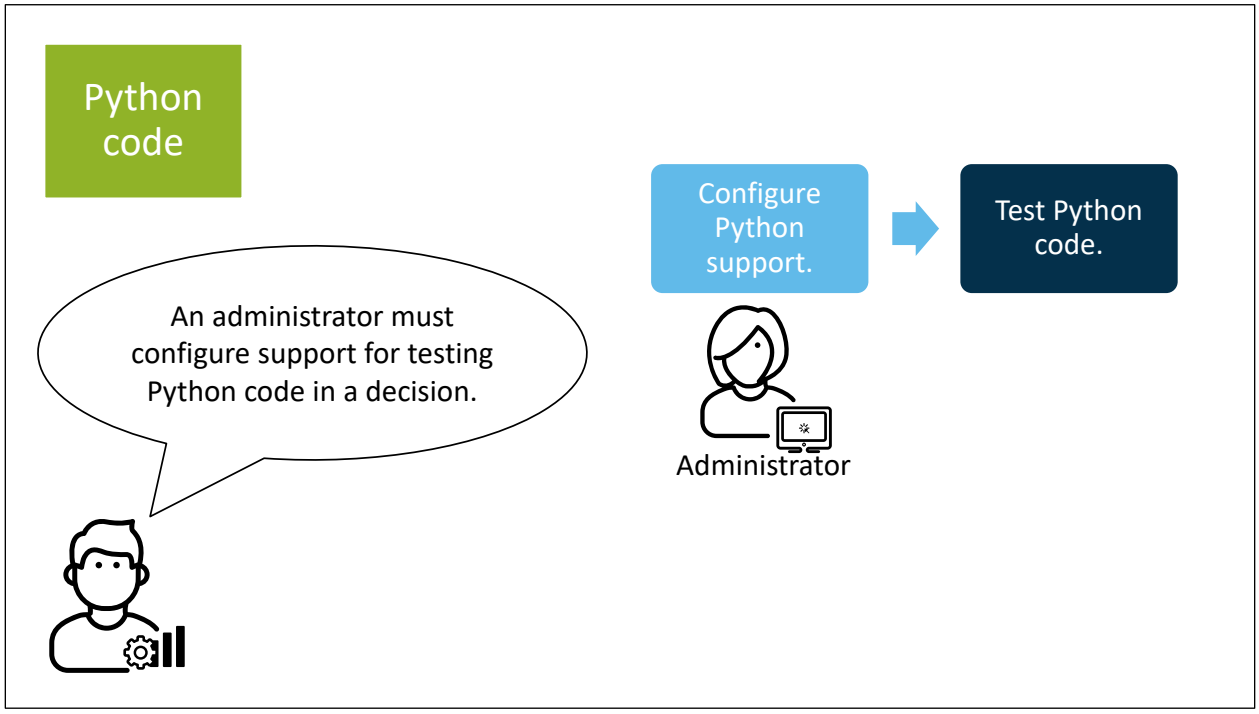
Versioning is not currently supported for queries created in SAS Studio. However, that might change in a future release.



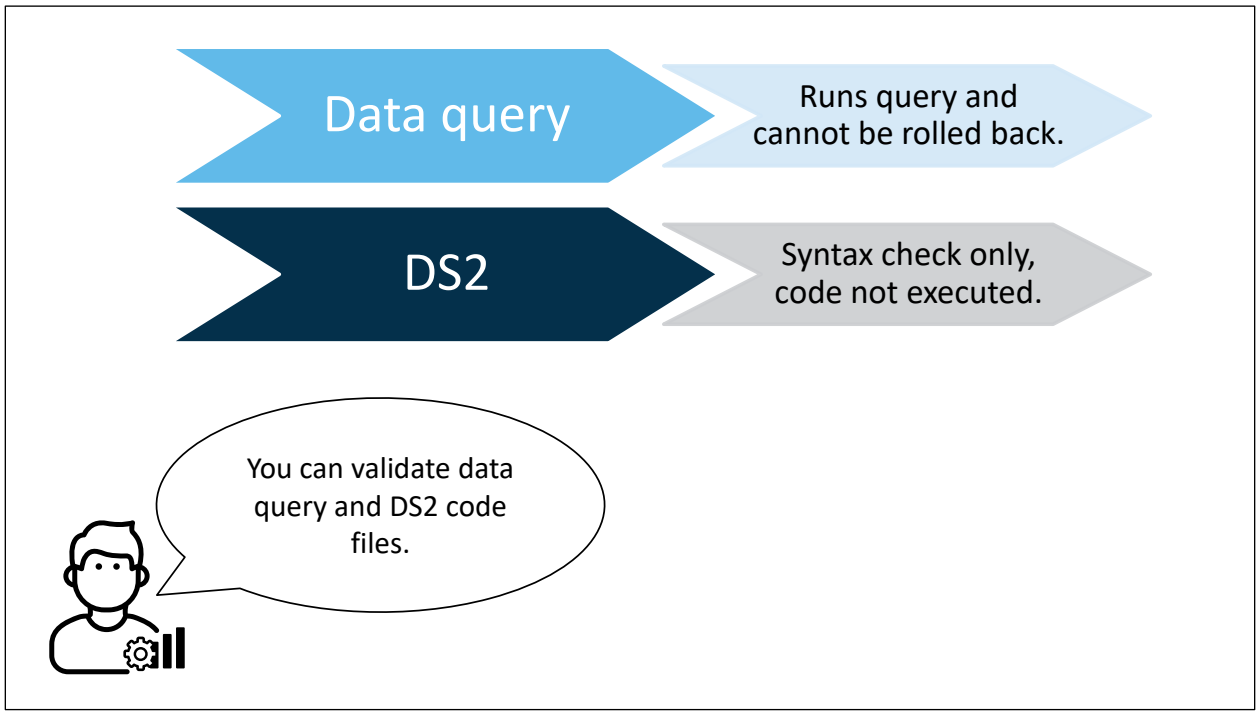
When using the DS2 code file, there are considerations for two predefined DS2 packages: SQLSTMT and HTTP. You should use SQLSTMT only when publishing to the Micro Analytic Service.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The use of HTTP is strongly discouraged for CAS, Hadoop, Git and Teradata unless you are processing a small amount of data.

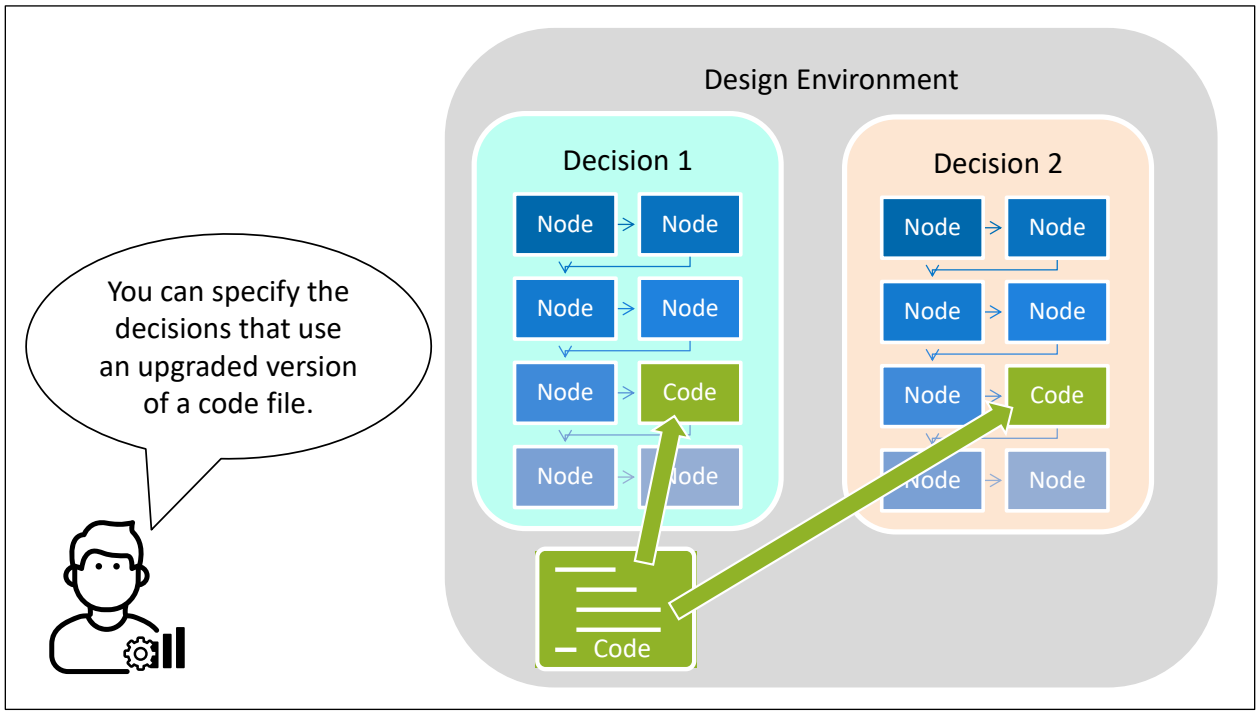


In order to run a test for a decision that contains a Python code file, you or an administrator must configure support for Python.

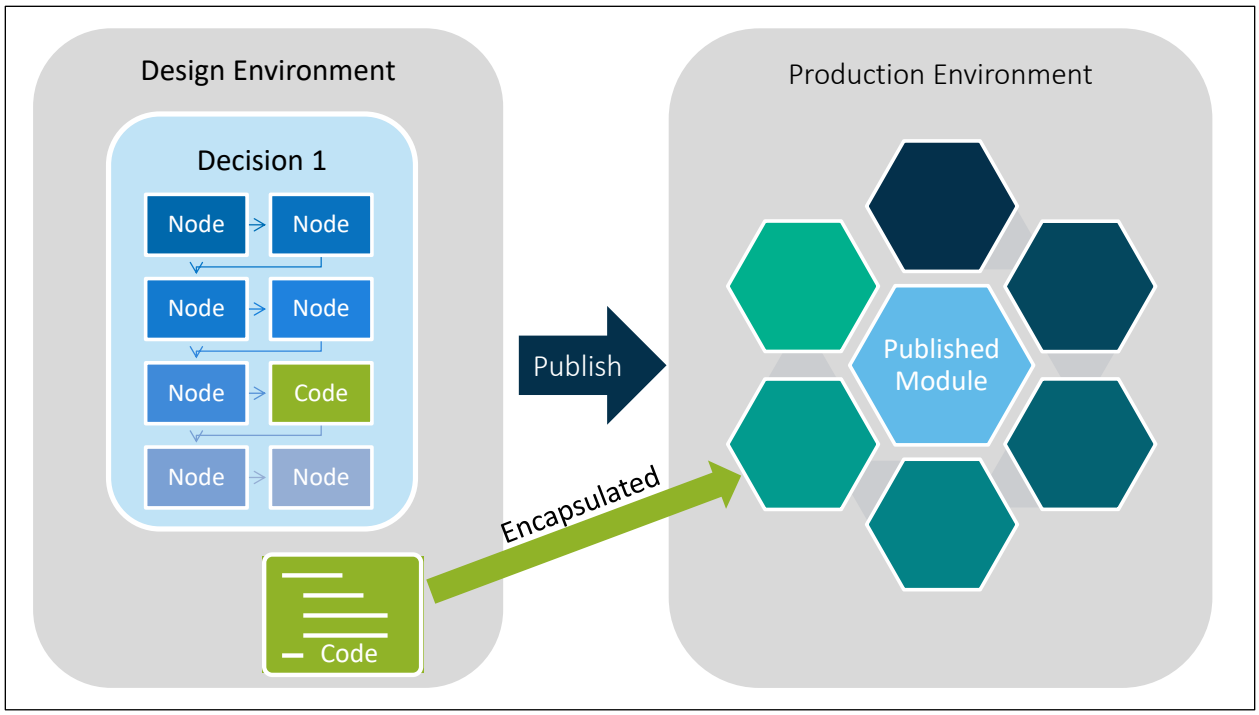


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

You can validate data query and DS2 code files. Validation of a data query actually runs the query. If the query performs an INSERT, UPDATE, or DELETE, these actions are completed and cannot be rolled back. Validation of a DS2 file performs a syntax check only and does not execute the code.



If you create a new version of a code file that is already used in other decisions, you can upgrade the decisions to use the new version.



Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

When you publish a decision with a DS2 or data query code node, the latest version of the code is encapsulated in the published module. Python code files are published with a name corresponding to the decision, so there could be multiple files if the same decision is published multiple times with the same code file. There is no sharing of the code after it is published.

3.01 Activity

In the virtual lab, sign into SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the L at the top right of the interface and select **Help Center**. Locate the documentation for developing SQL code for data query files. How do you specify decision variables in a query? How do you specify input variables?

Hint: Search for the text **sql code** and click the link to the documentation for data query files.

3.01 Activity – Correct Answer

In the virtual lab, sign into SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the L at the top right of the interface and select **Help Center**. Locate the documentation for developing SQL code for data query files. How do you specify decision variables in a query? How do you specify input variables?

Hint: Search for the text **sql code** and click the link to the documentation for data query files.

Answer: You enclose decision variables in braces and specify input variables with a question mark, such as in the following example:

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

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Creating a Data Query for Use in a Decision

This demonstration illustrates using the SQL editor to create a data query.

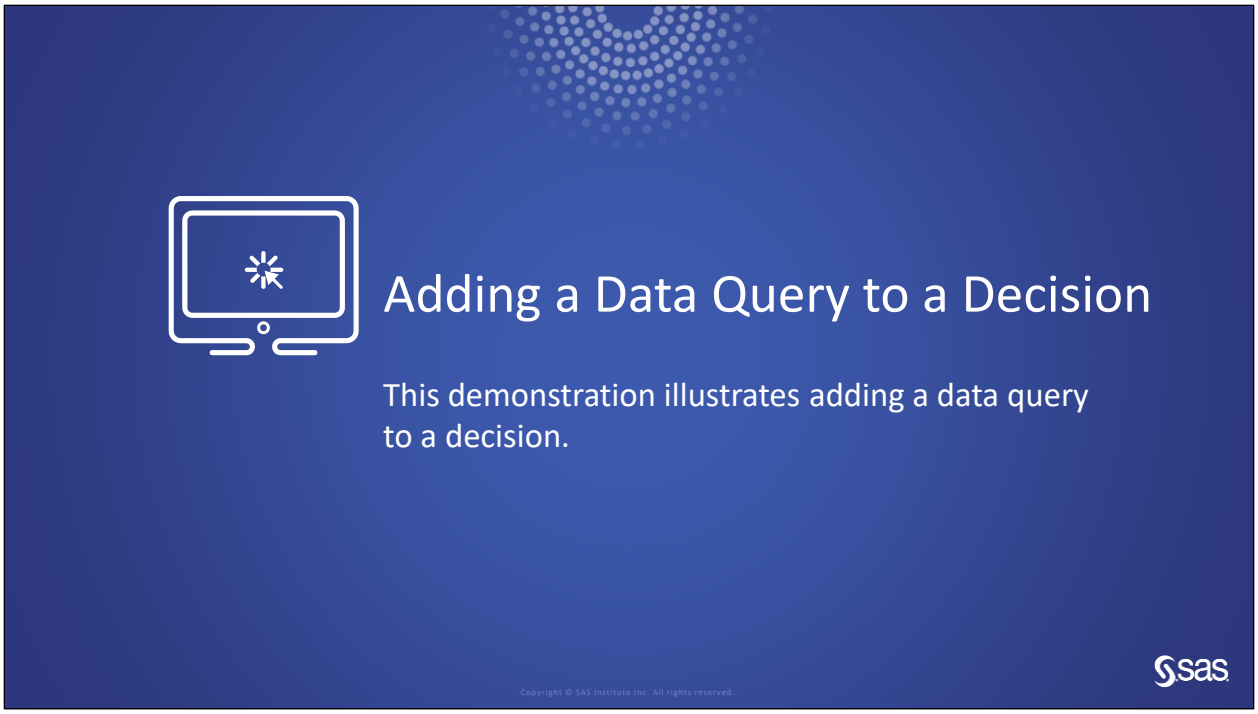
Copyright © SAS Institute Inc. All rights reserved.

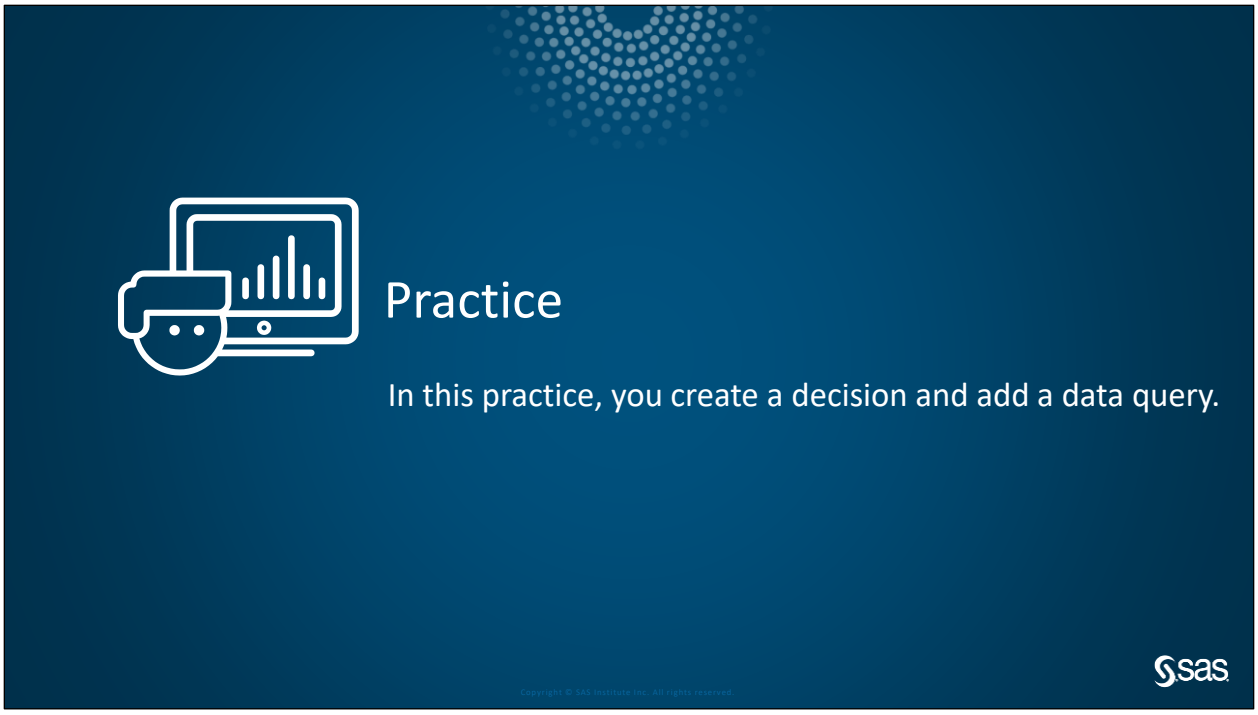
Practice

In this practice, you create a data query using the SQL editor.

Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.02 Activity

In the virtual lab, sign into SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the L at the top right of the interface and select **Help Center**. Locate the documentation for DS2 code files. What DS2 data types are supported for Intelligent Decisioning?

Hint: Search for the text **ds2 code** and click the link to the documentation for DS2 code files.

3.02 Activity – Correct Answer

In the virtual lab, sign into SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the L at the top right of the interface and select **Help Center**. Locate the documentation for DS2 code files. What DS2 data types are supported for Intelligent Decisioning?

Hint: Search for the text **ds2 code** and click the link to the documentation for DS2 code files.

Answer: The supported data types are double, varchar, and package datagrid.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Adding DS2 Code to a Decision

This demonstration illustrates adding DS2 code to a decision.

Copyright © SAS Institute Inc. All rights reserved.

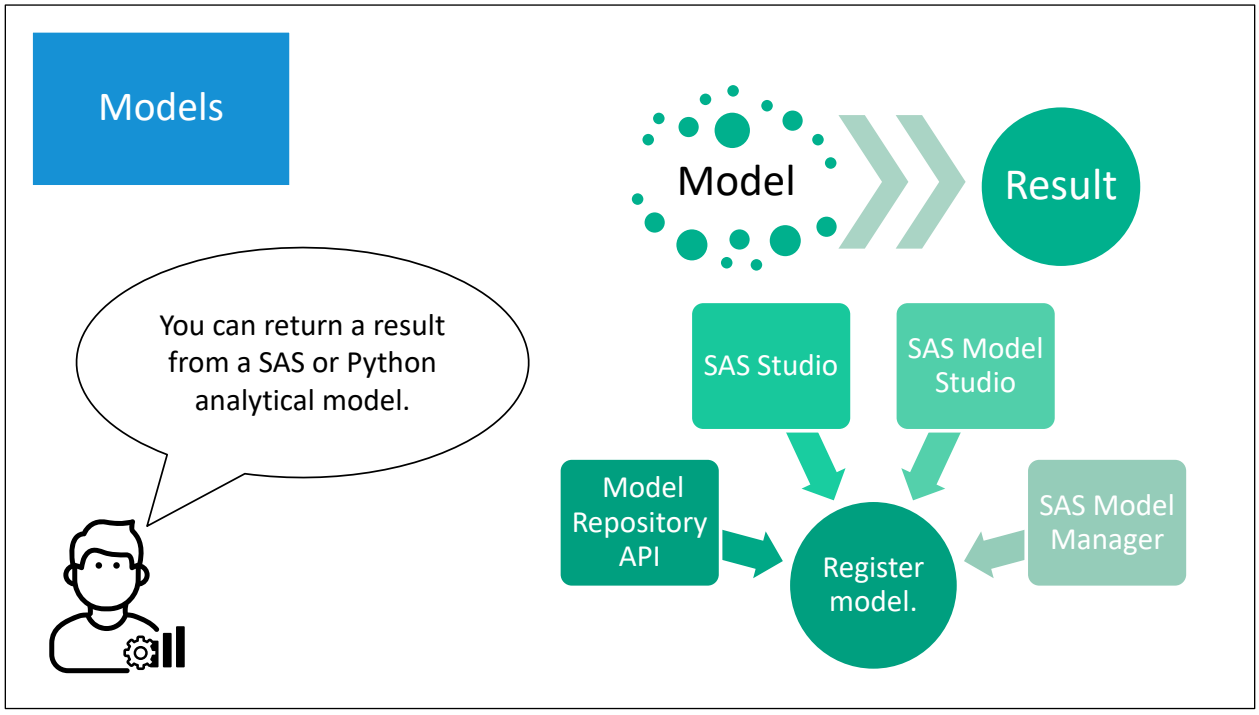
Practice

In this practice, you define a DS2 code file for use in a decision.

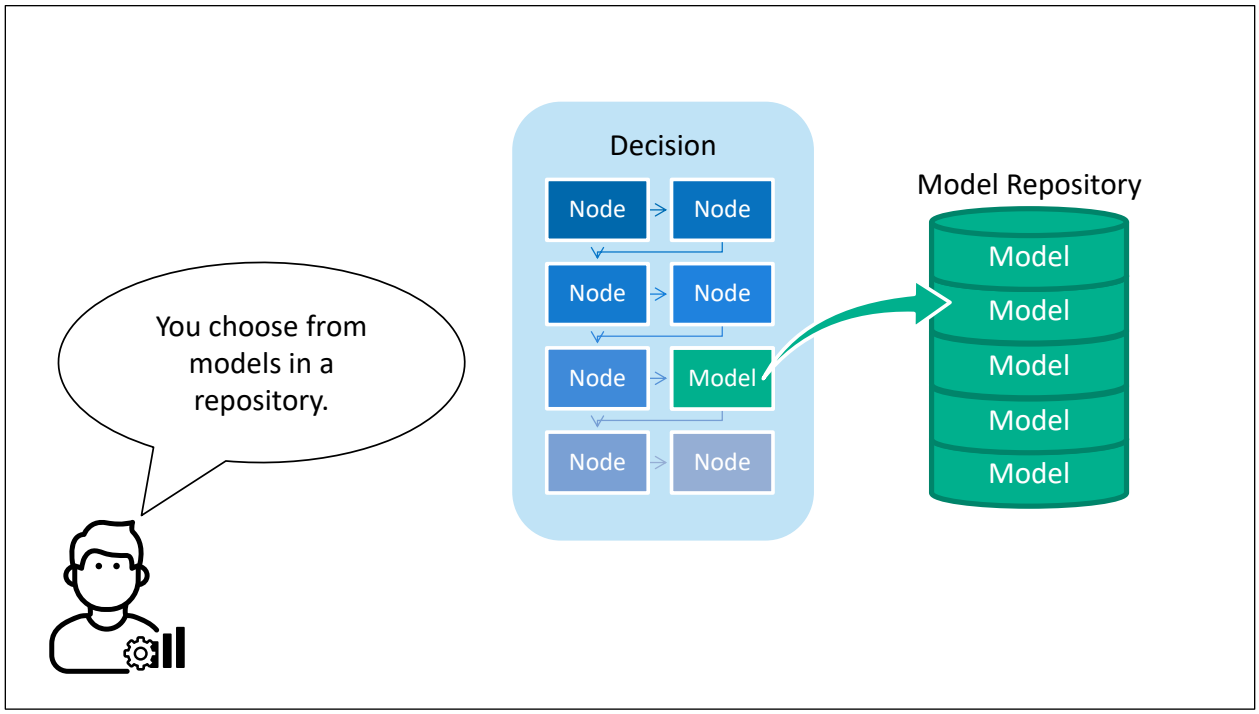
Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.2 Models

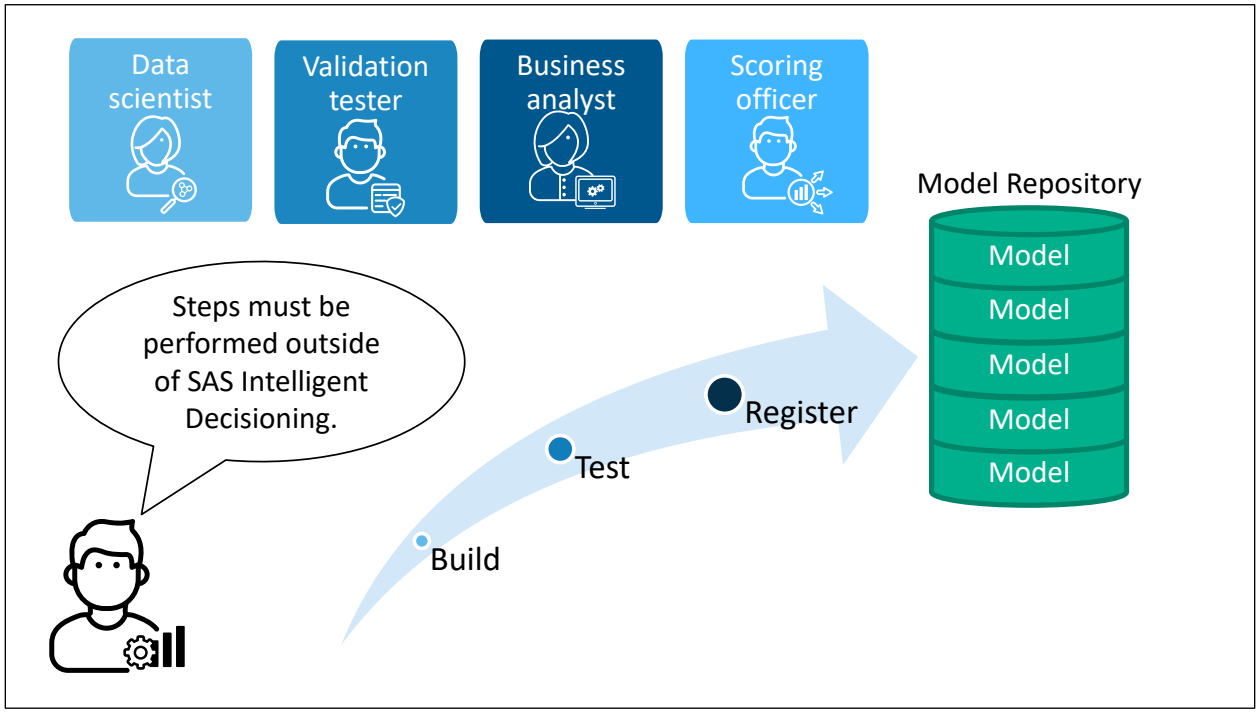


In a decision, you can potentially return a result from an analytical model to use in decision processing. The model can be a SAS model or a Python model. In order to use a model with SAS Intelligent Decisioning, it must be registered using one of several different applications.

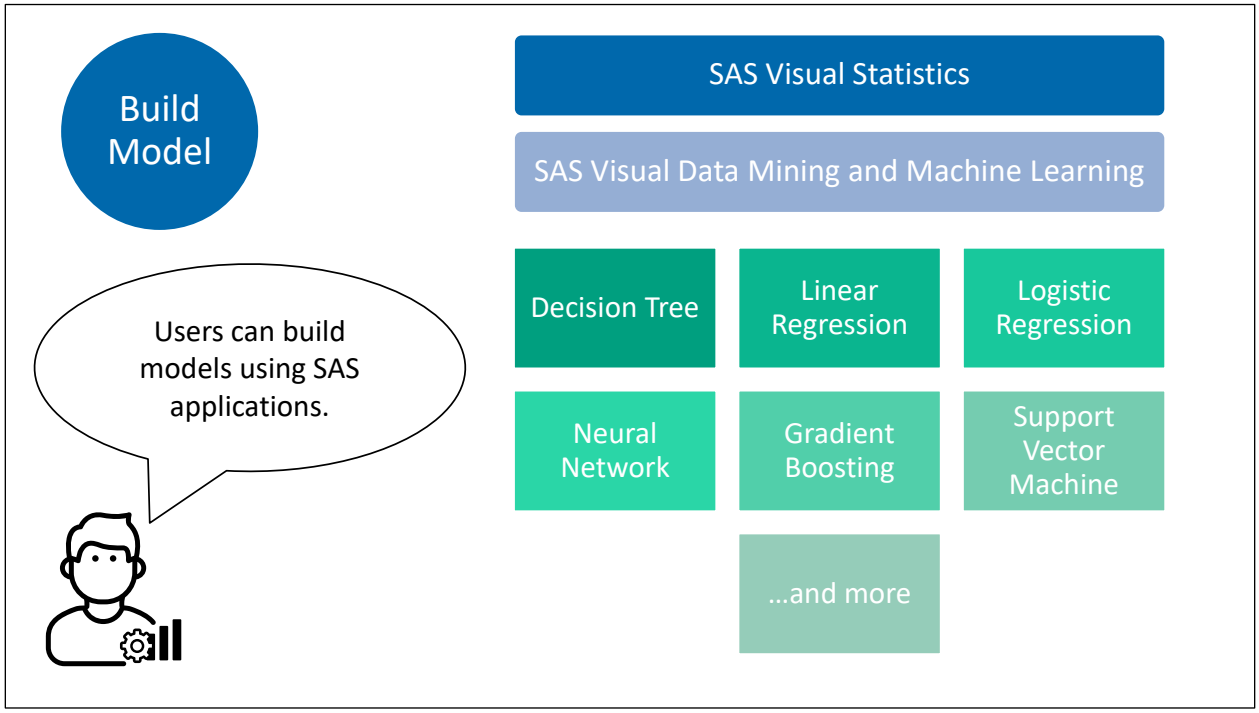


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

When you add a model to a decision, you choose from among models that have been registered in a model repository.

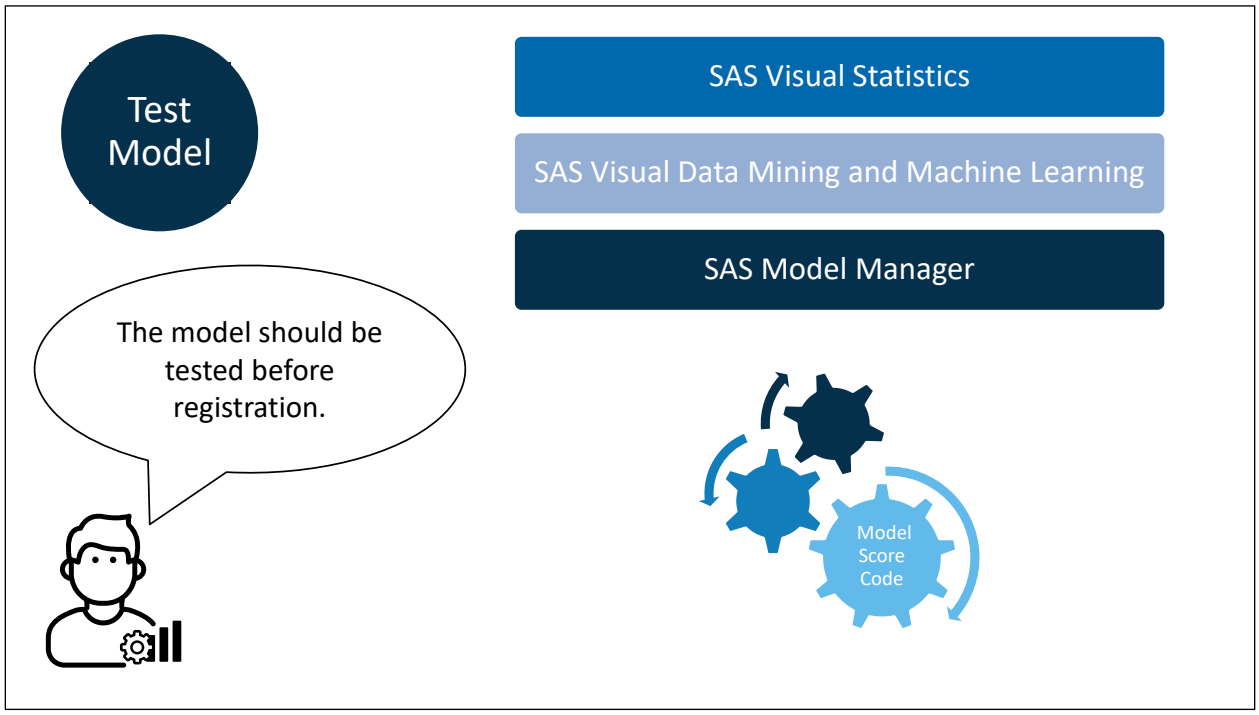


There are multiple tasks that must be performed to make a model available in a model repository. The model must be built, tested, and then finally registered in the repository. These activities are performed outside of SAS Intelligent Decisioning, and are likely performed by other users in your organization.

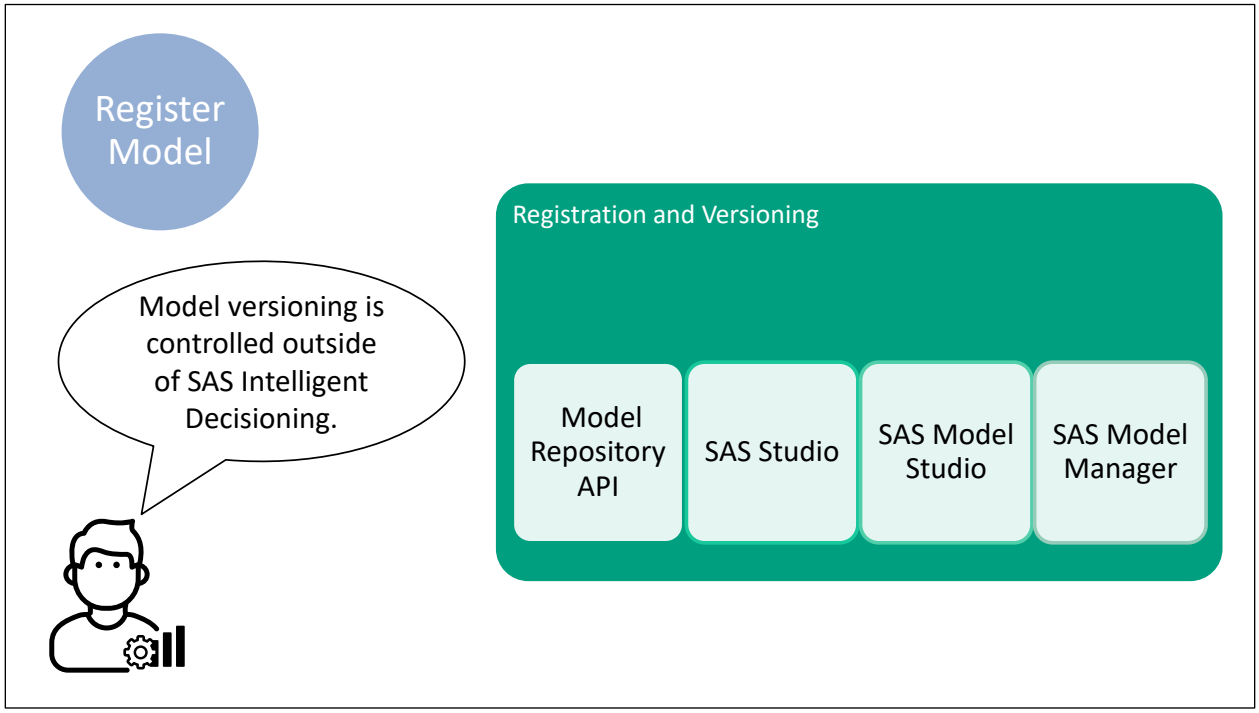


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Users can build models using SAS applications such as SAS Visual Statistics and SAS Visual Data Mining and Machine Learning. SAS Intelligent Decisioning supports many types of models, including those shown here.

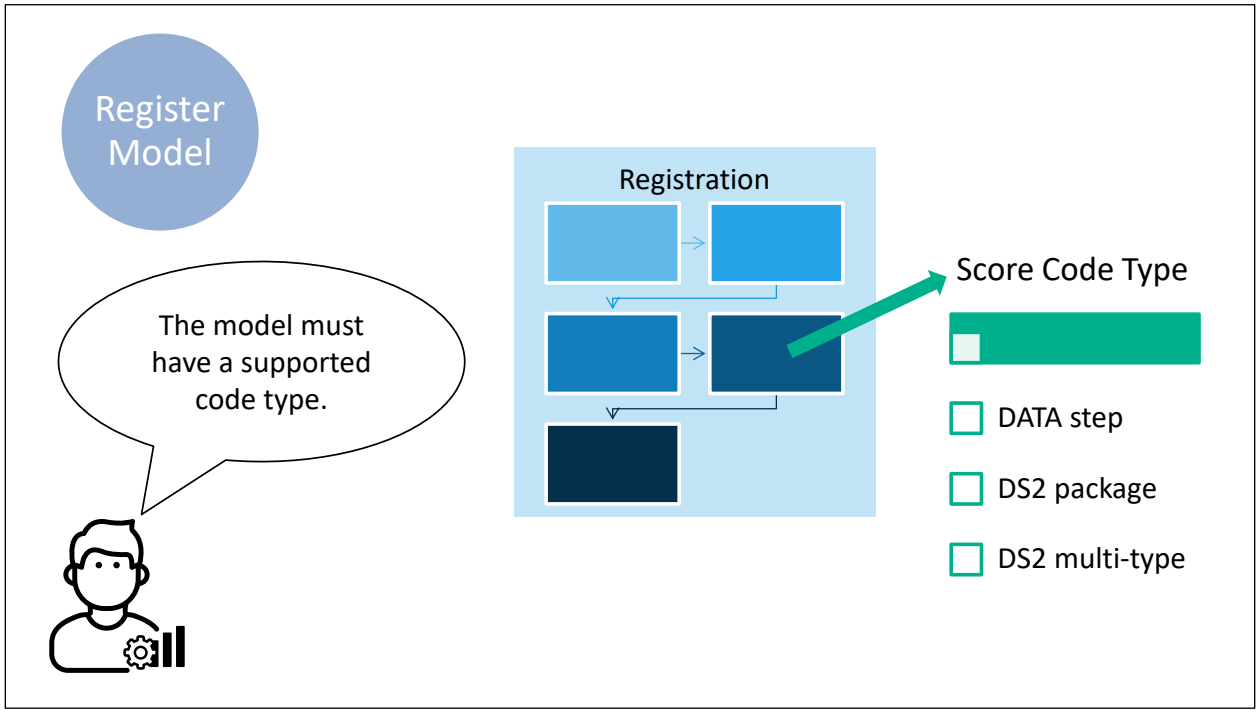


The model score code should be tested before the model is registered in the repository. Testing can potentially take place using the application that was used to build the model. A user can also test the model score code in SAS Model Manager.

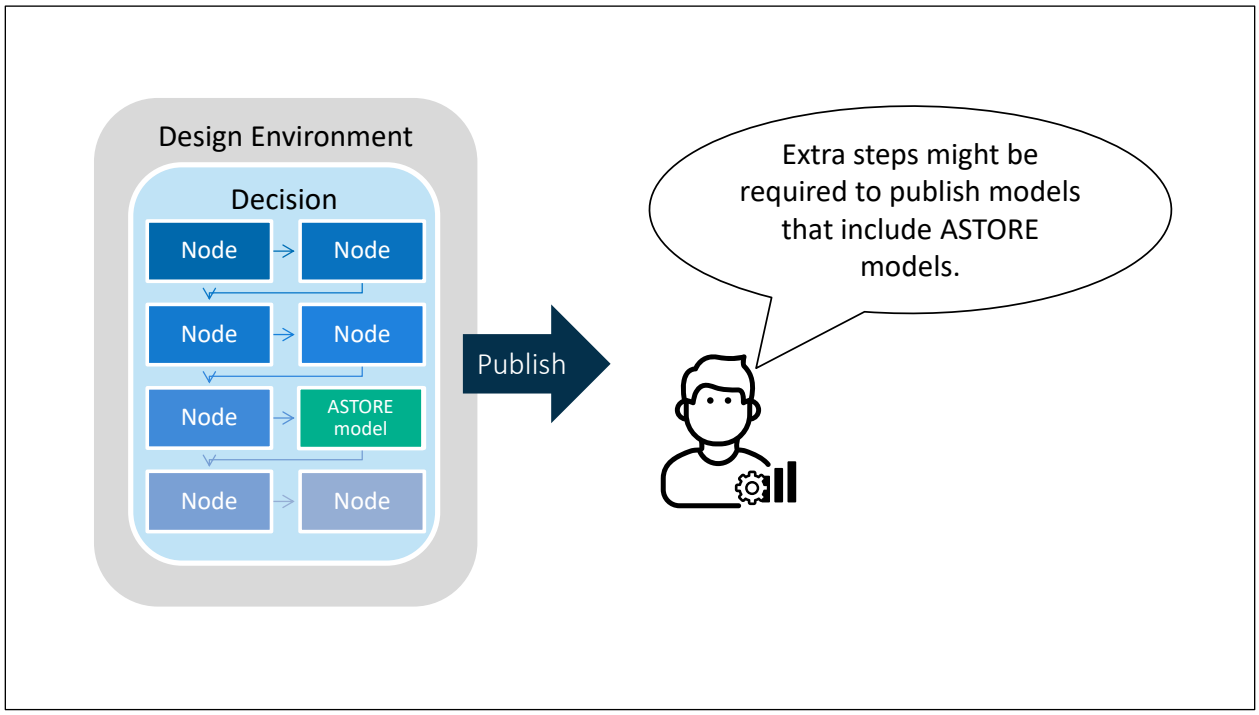


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Recall that models are registered in the repository using one of several applications. Model versioning is controlled outside of SAS Intelligent Decisioning.

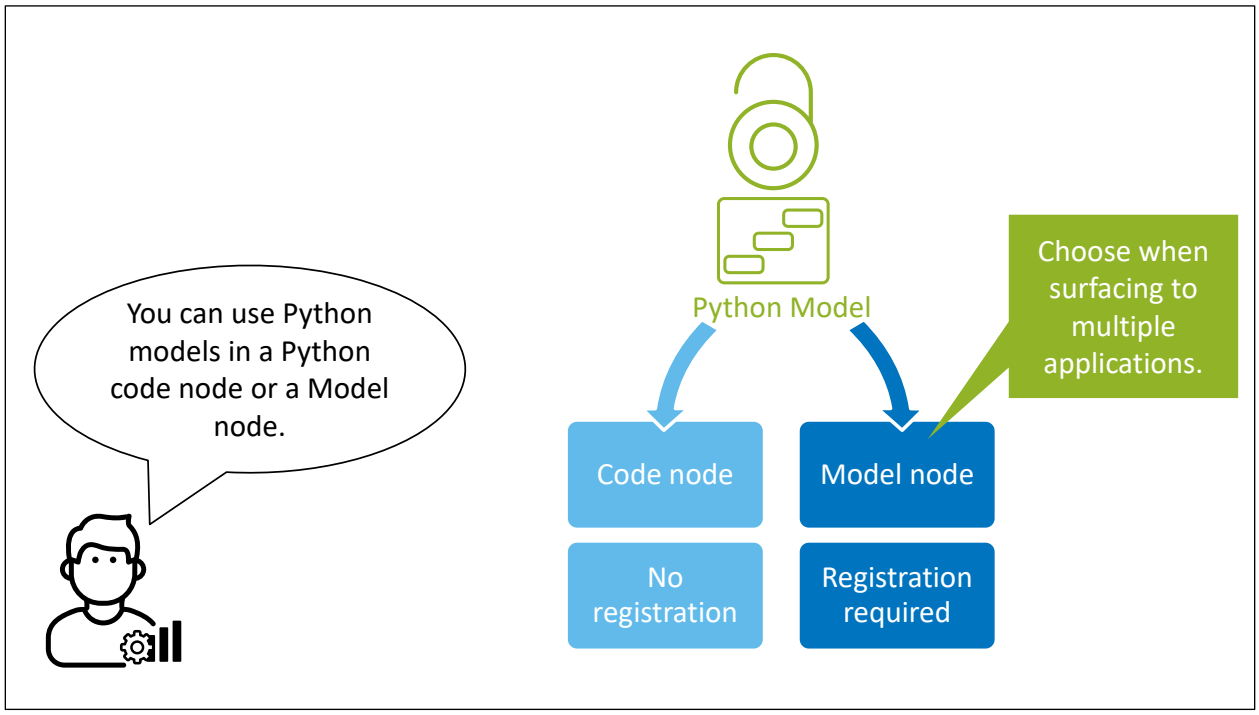


During registration, a user specifies the score code type value. In order to be used in Intelligent Decisioning, the score code type must be DATA step, DS2 package, or DS2 multi-type.



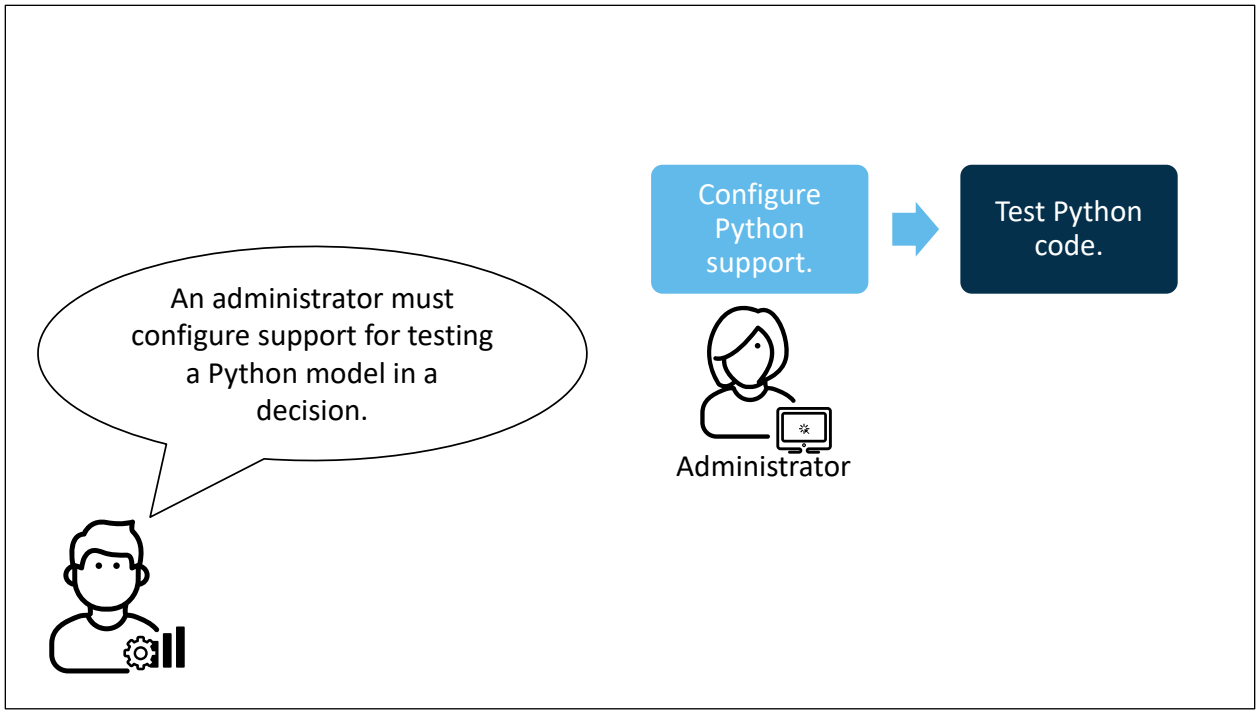
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

If you use an analytic store (or ASTORE) model in your decision, there are potentially some extra steps that you need to perform before you publish the decision. Refer to *SAS Intelligent Decisioning: User's Guide* for more information.

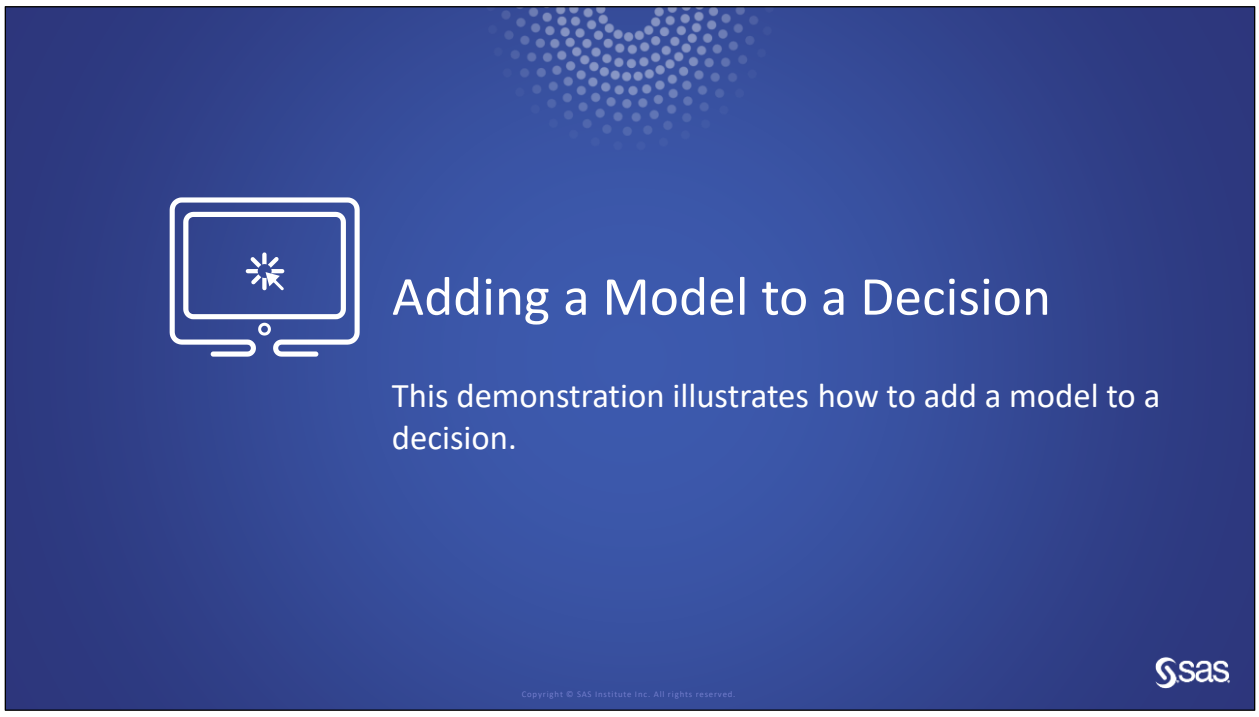


In the case of Python models, you might be able to include the score code in a decision using either a Python code node or a Model node. Registration is not required for a code node, but is required for the Model node. You might choose to perform the extra step of registration if you are surfacing the Python model to multiple SAS applications.

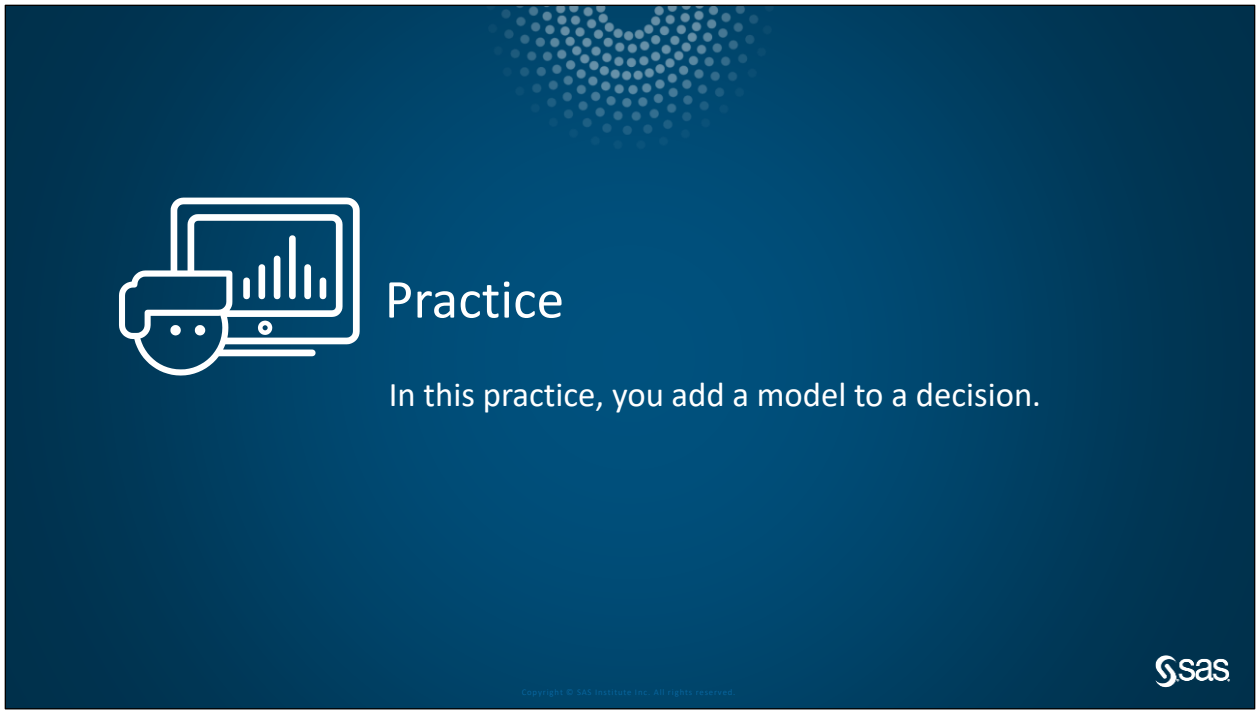
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



In order to run a test for a decision that contains a Python model, you or an administrator must configure support for Python.

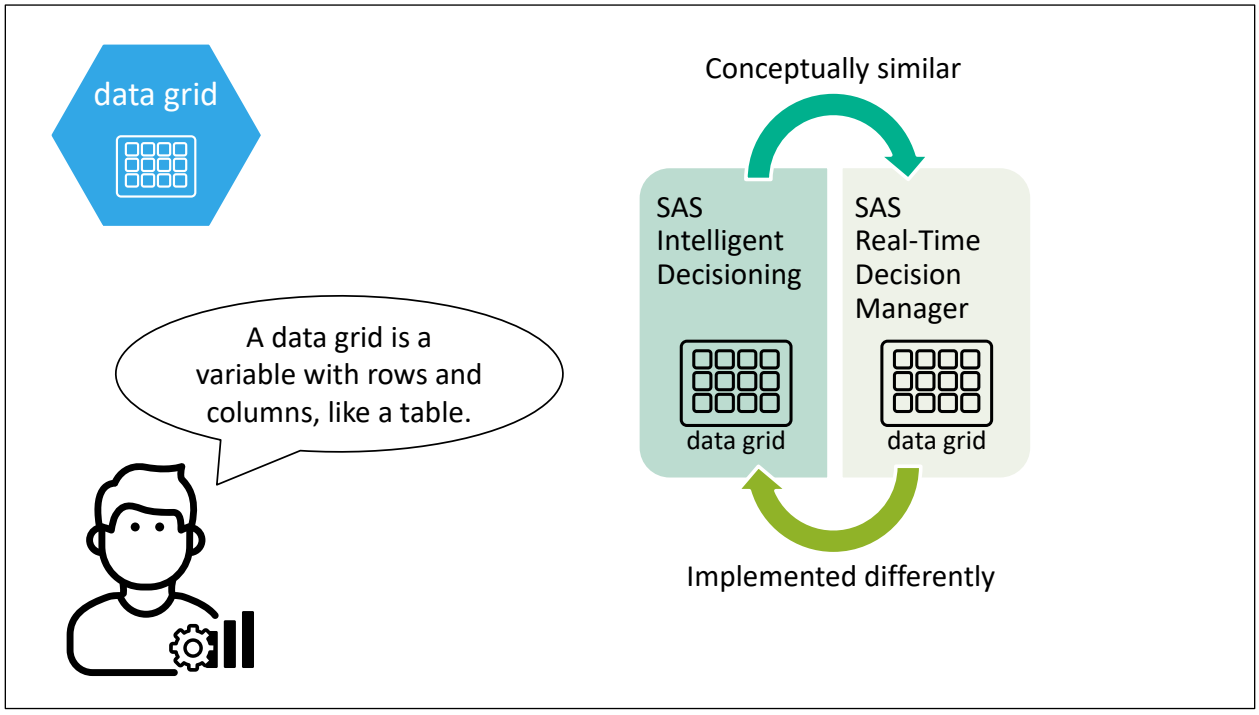


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

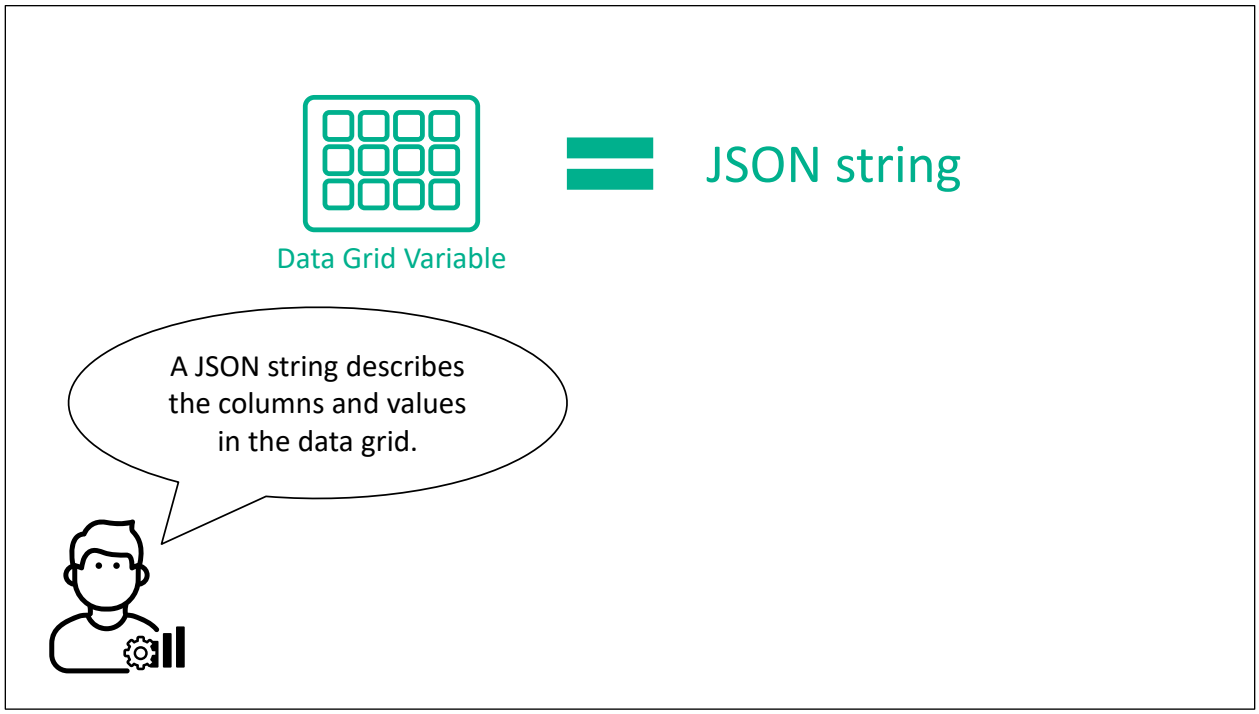


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.3 Data Grids

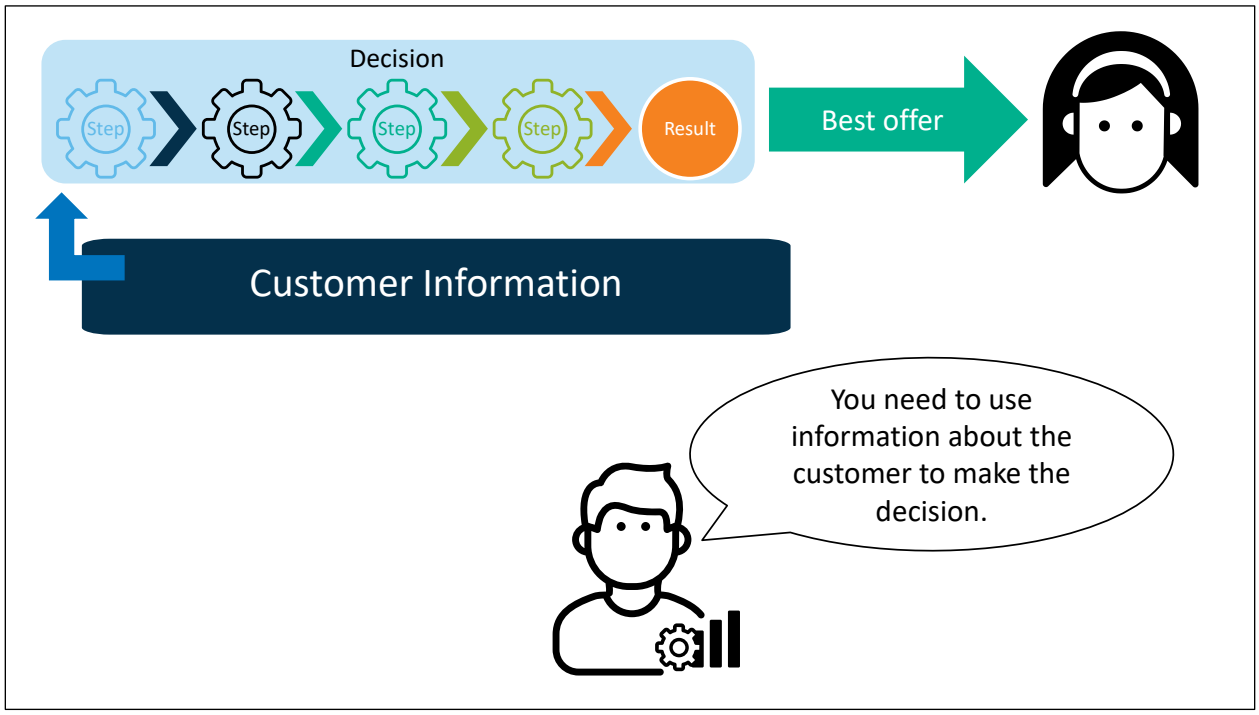


A data grid is a variable with rows and columns, like a table. If you are familiar with data grids in SAS Real-Time Decision Manager, they are conceptually similar in SAS Intelligent Decisioning. However, they are implemented differently.

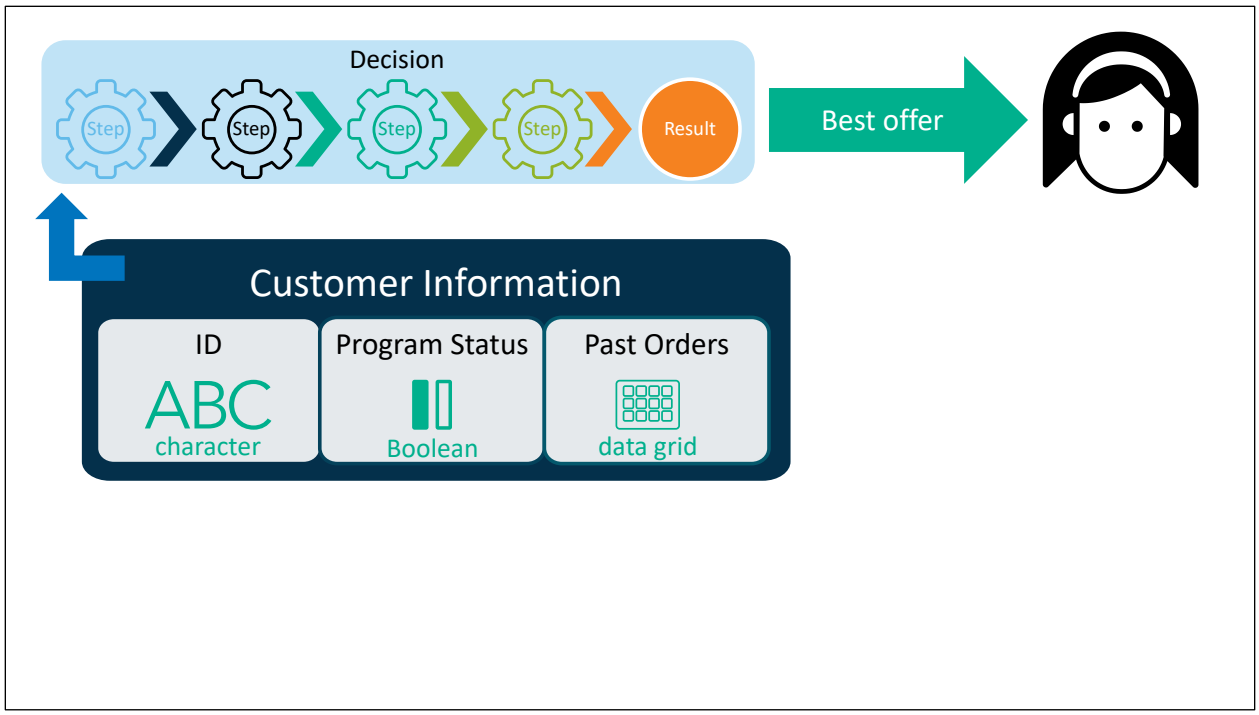


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

In SAS Intelligent Decisioning, data grids are represented as JSON strings with a specific format. The JSON string includes metadata that describes the columns in the data grid along with the values themselves. JSON (or Javascript Object Notation) is a format for storing and transporting data.

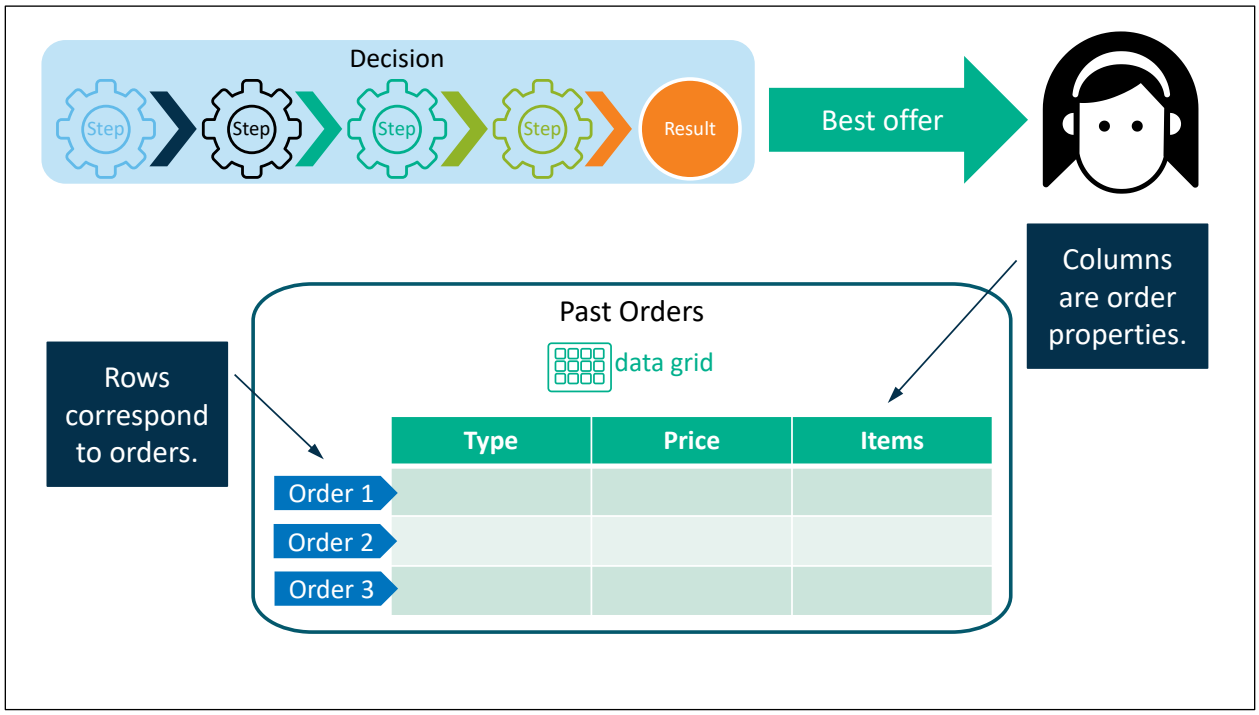


Suppose you are building a decision to determine the best offer to make to a customer and you need to use information about the customer to make the decision.

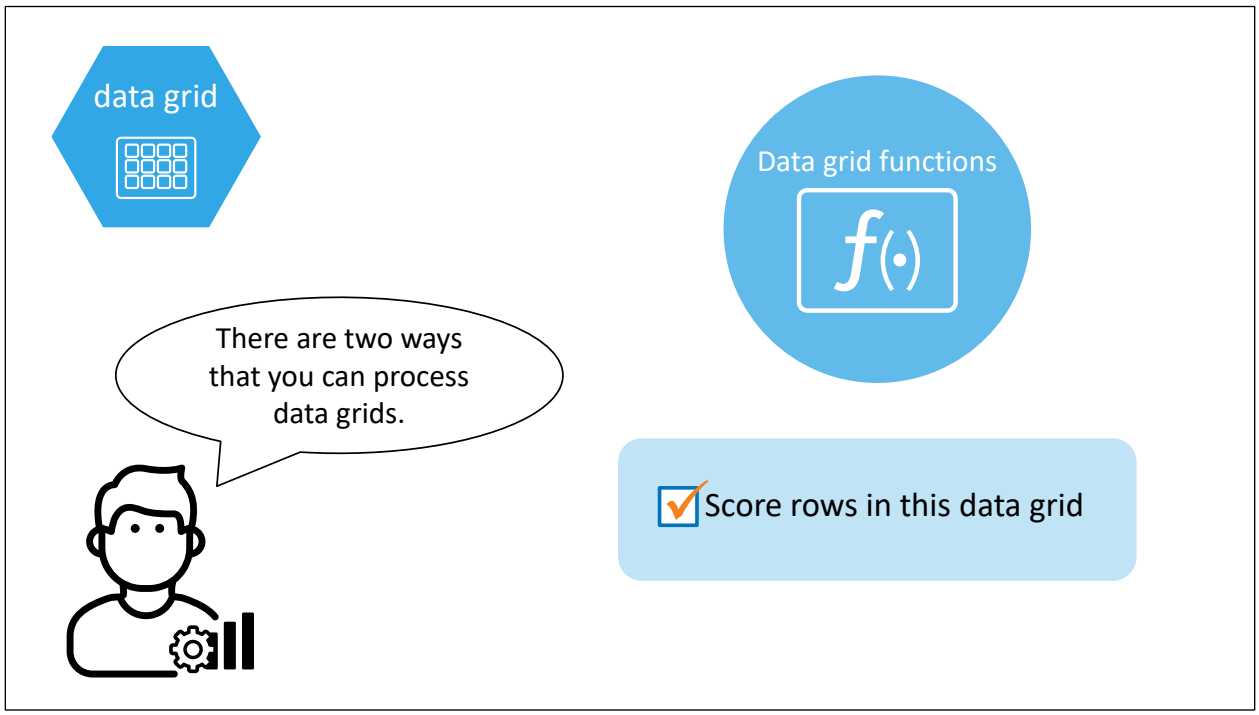


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The information that you need to use includes the customer's customer ID, loyalty program membership status, and details about each of the orders that they placed over the past year. The customer ID might be designated as a character variable, program membership as Boolean, and information about past orders could be represented by a data grid.

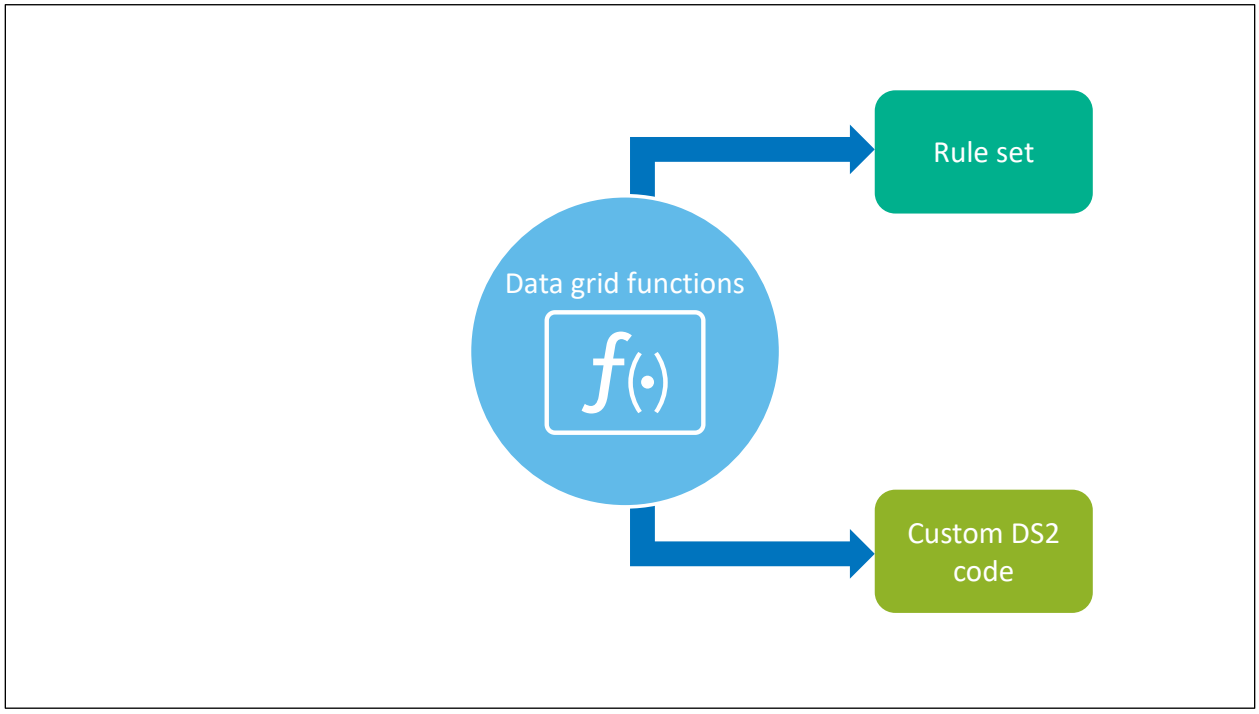


In the data grid, each row corresponds to an order, and columns correspond to properties of the order, such as order type, price, and number of items.



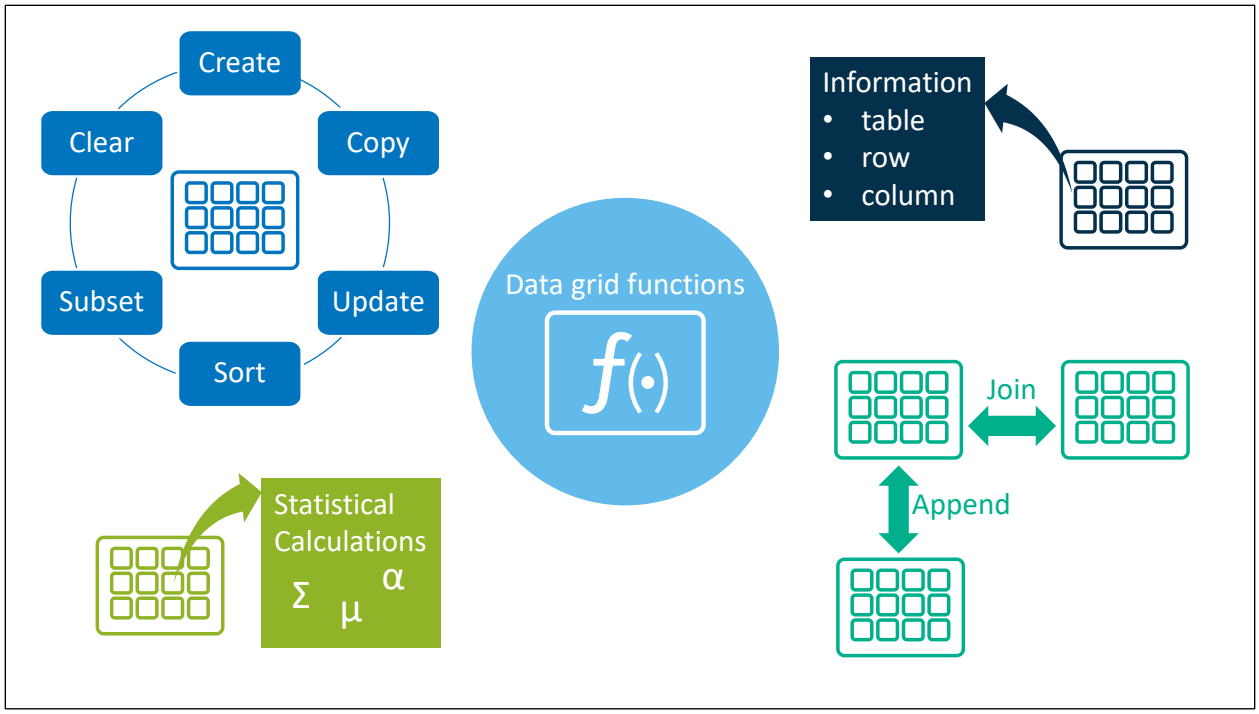
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

There are two ways that you can process data grids. In some situations, you can use data grid functions. And in some situations, you can choose to score rows in a data grid.

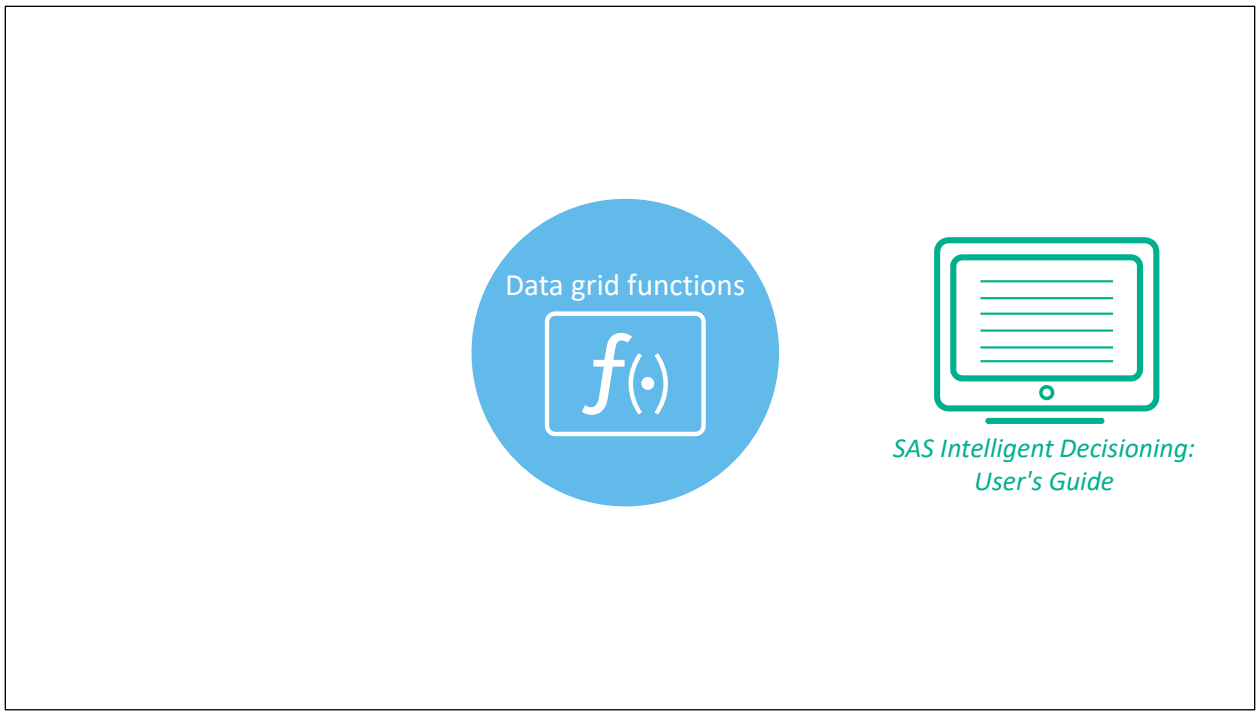


Let's begin with data grid functions. You can use data grid functions in rule sets and in custom code that uses the SAS DS2 language.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

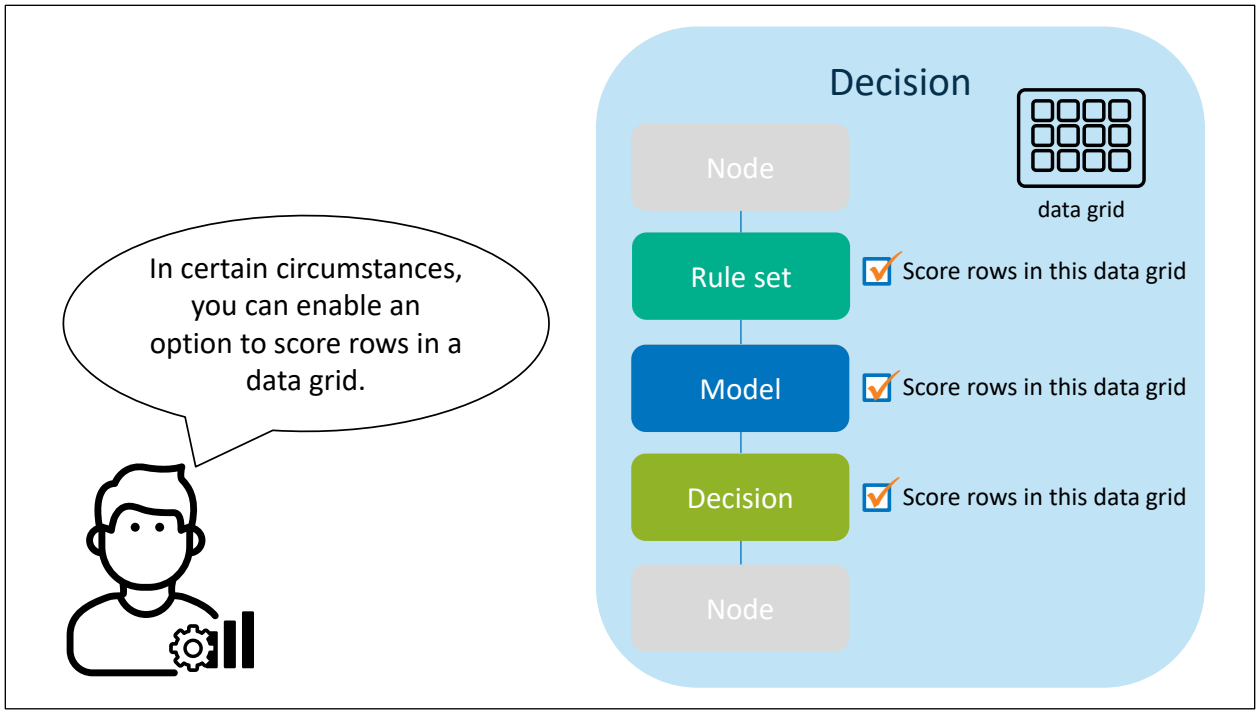


You can use data grid functions to perform a variety of types of operations. For example, you can create, copy, update, sort, subset, or clear the data in a data grid. You can return information about an entire data grid or about a column or row in the data grid. You can join or append two data grids. You can perform statistical calculations on values in the grid. Those are just some examples of the types of operations you can perform.

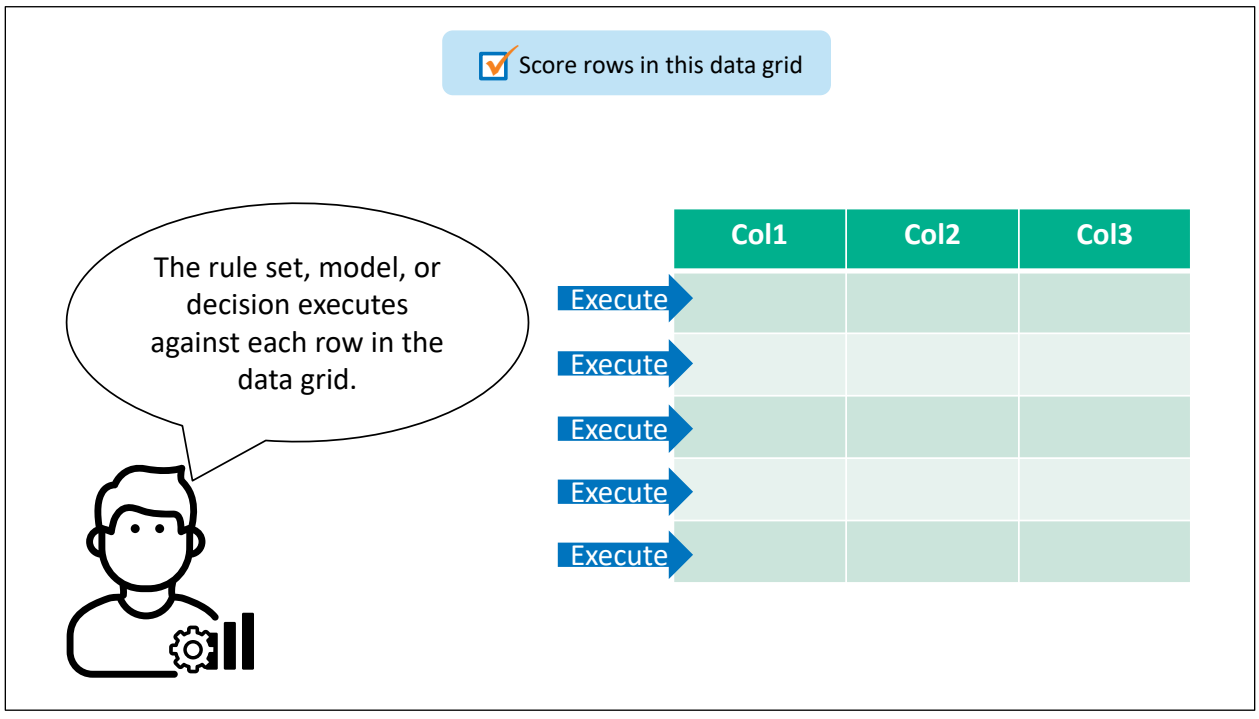


Refer to *SAS Intelligent Decisioning: User's Guide* to learn about the available data grid functions.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

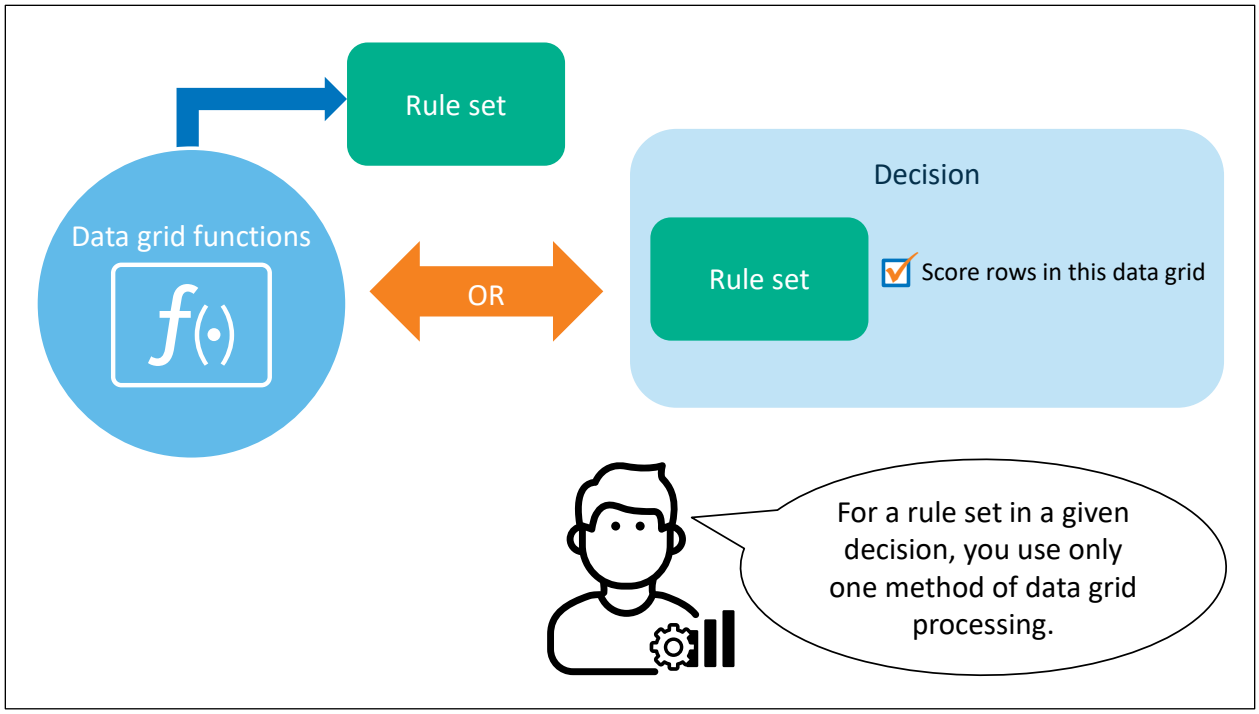


In a decision, when you configure a rule set, model, or another a decision and the decision includes at least one data grid variable, you can enable an option to score rows in the data grid.

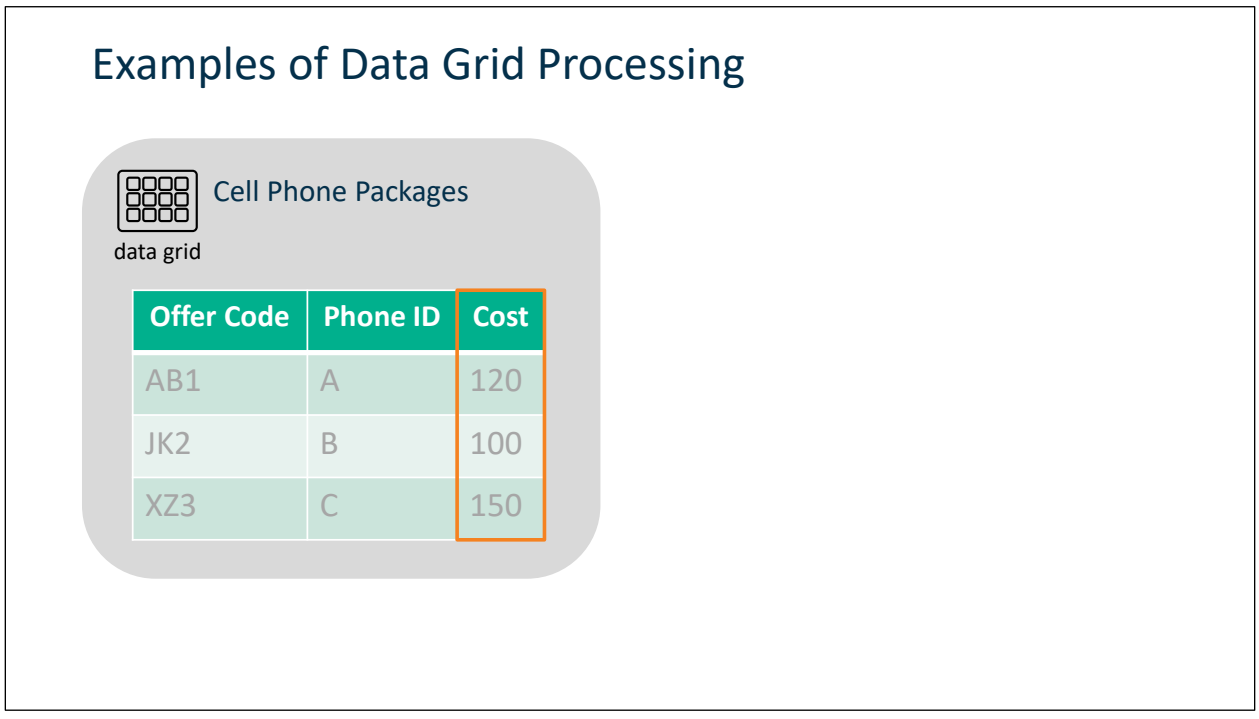


When you enable this option, the rule set, model, or decision executes against each row in the data grid.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

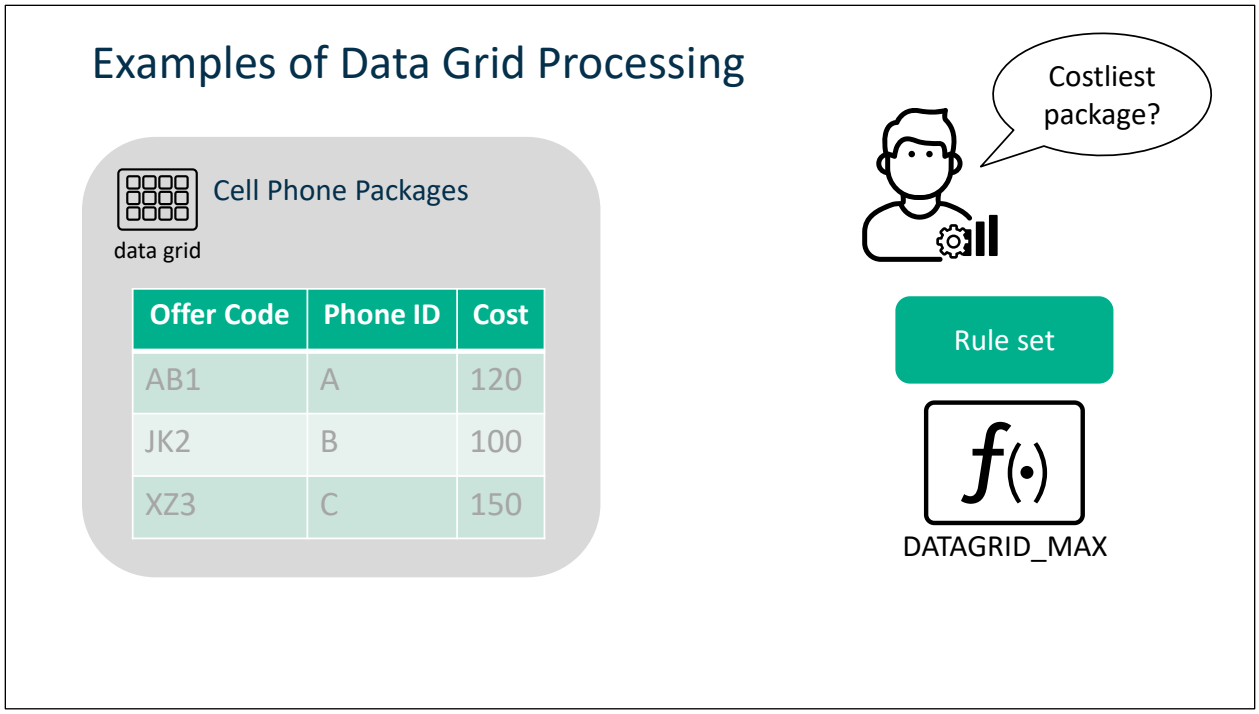


Although you can use both methods with rule sets, for a rule set in a given decision, you use only one or the other.

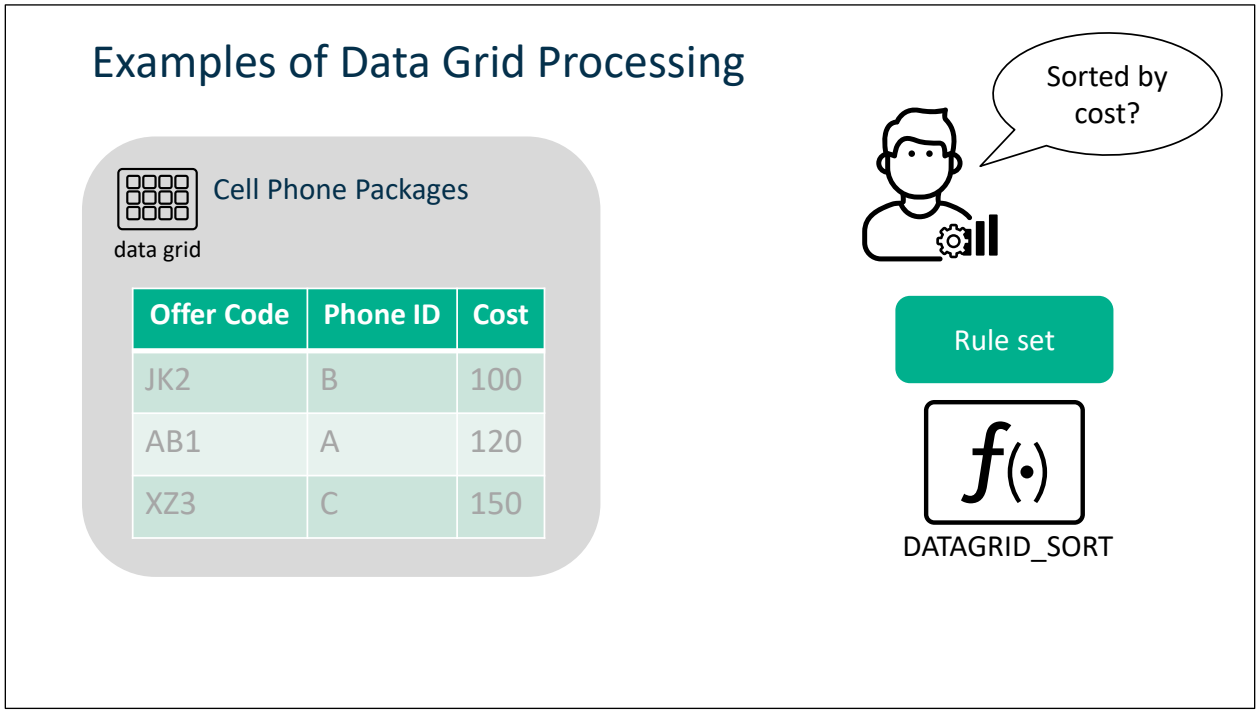


Let's look at some examples of data grid processing. Suppose you are working with a data grid that contains information about potential cell phone packages to offer to a customer. One of the columns is the cost of the package.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

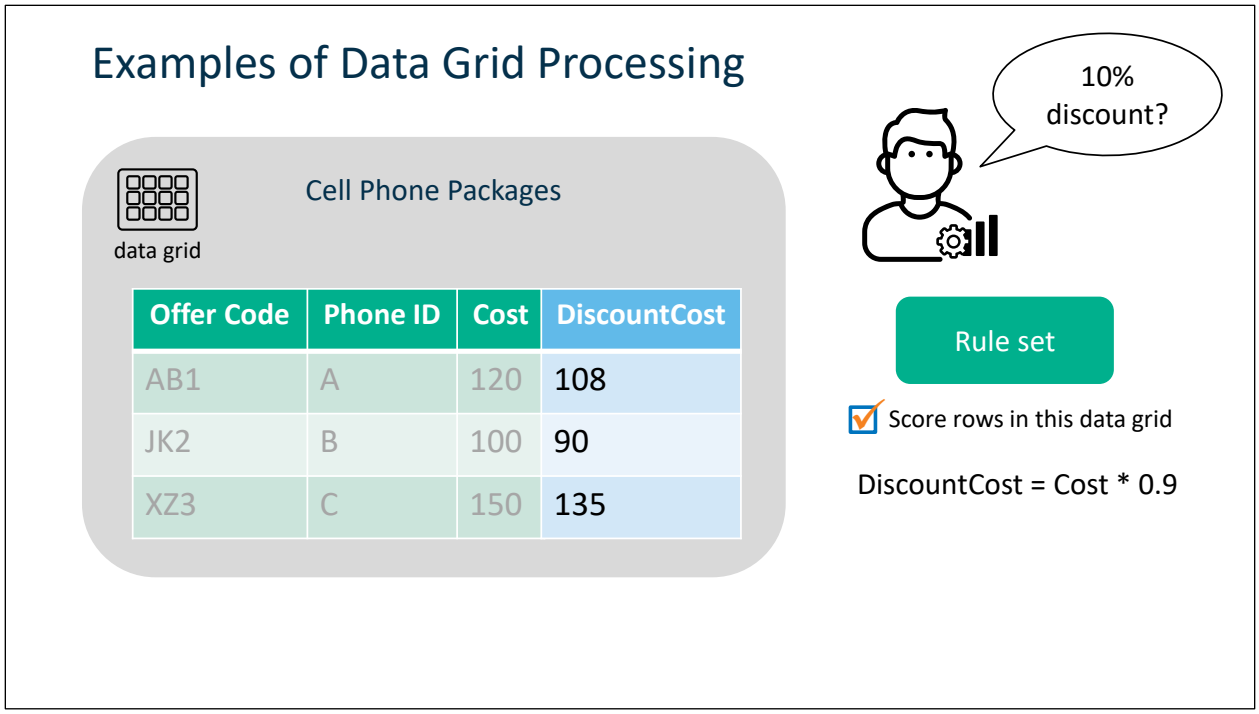


If you want to determine the price of the costliest package, you could use the data grid function DATAGRID_MAX in a rule set. This function returns a result with a data type of double, which you can assign to a variable of the same type.

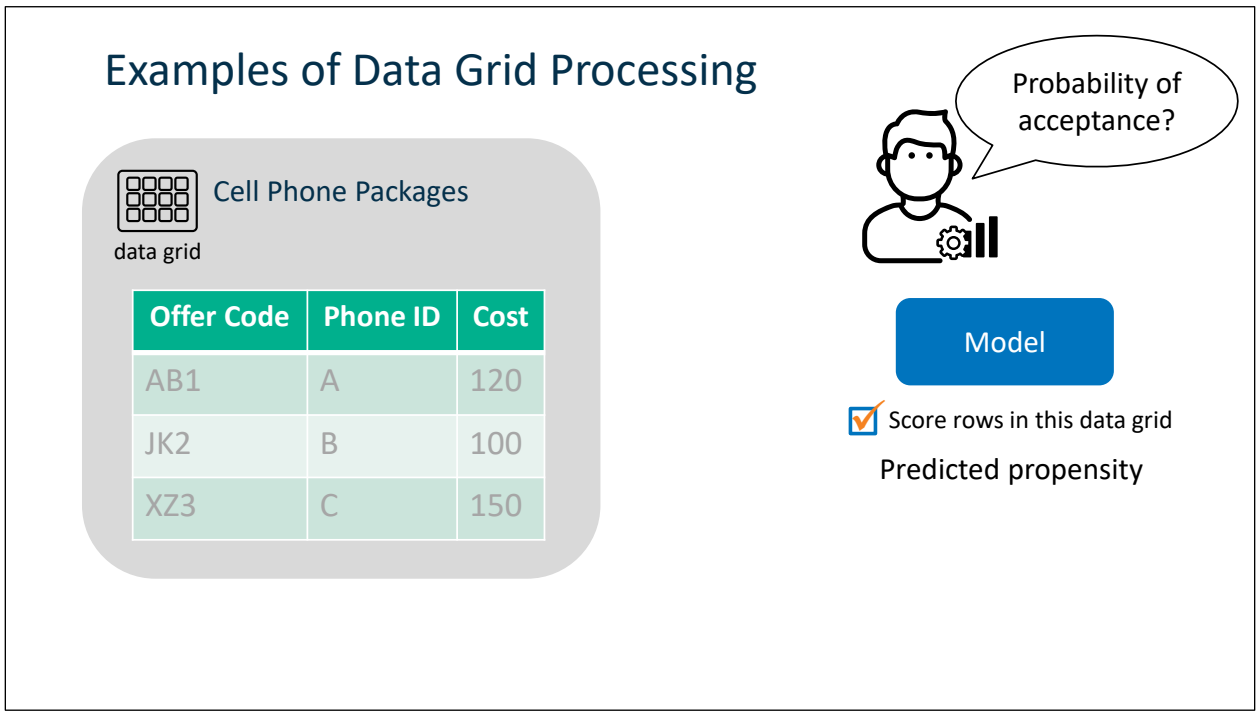


If you want to sort the offers by cost, you could use the data grid function DATAGRID_SORT in a rule set. This function writes the sorted result to a data grid variable that you specify.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

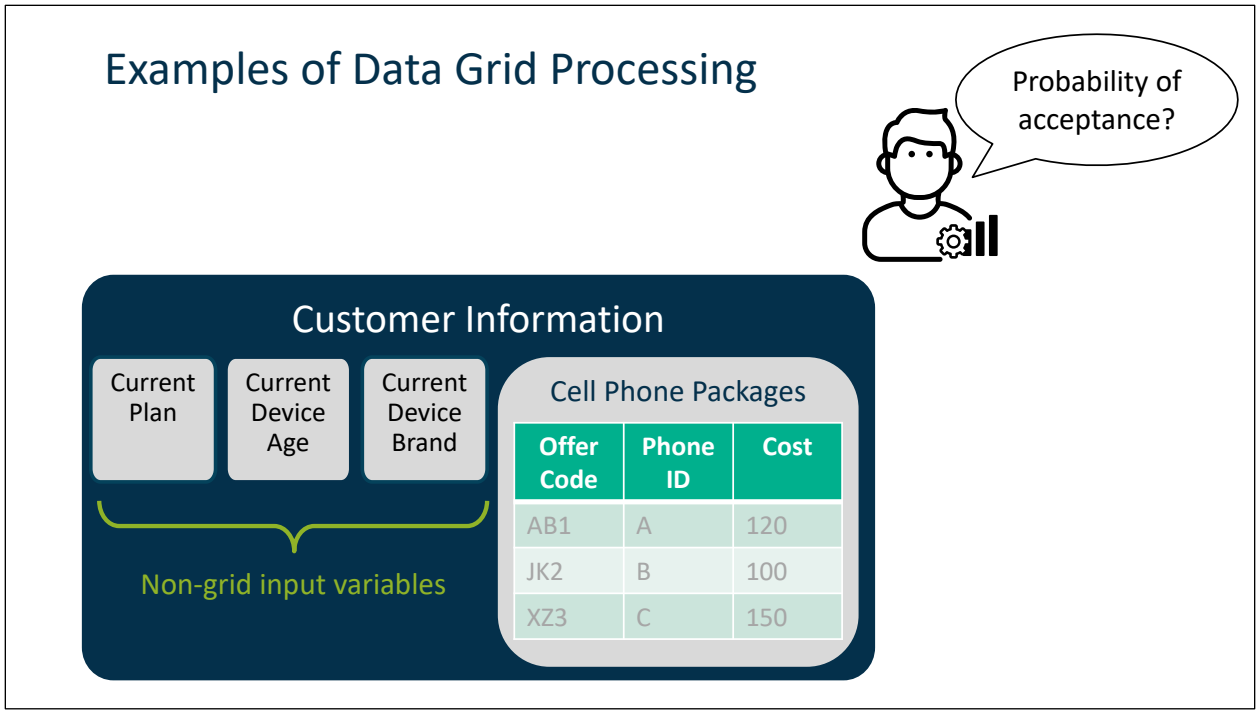


If you want to apply a percentage discount to the cost of each package, you could create a rule set that includes an expression to calculate the discount and score rows in the data grid.

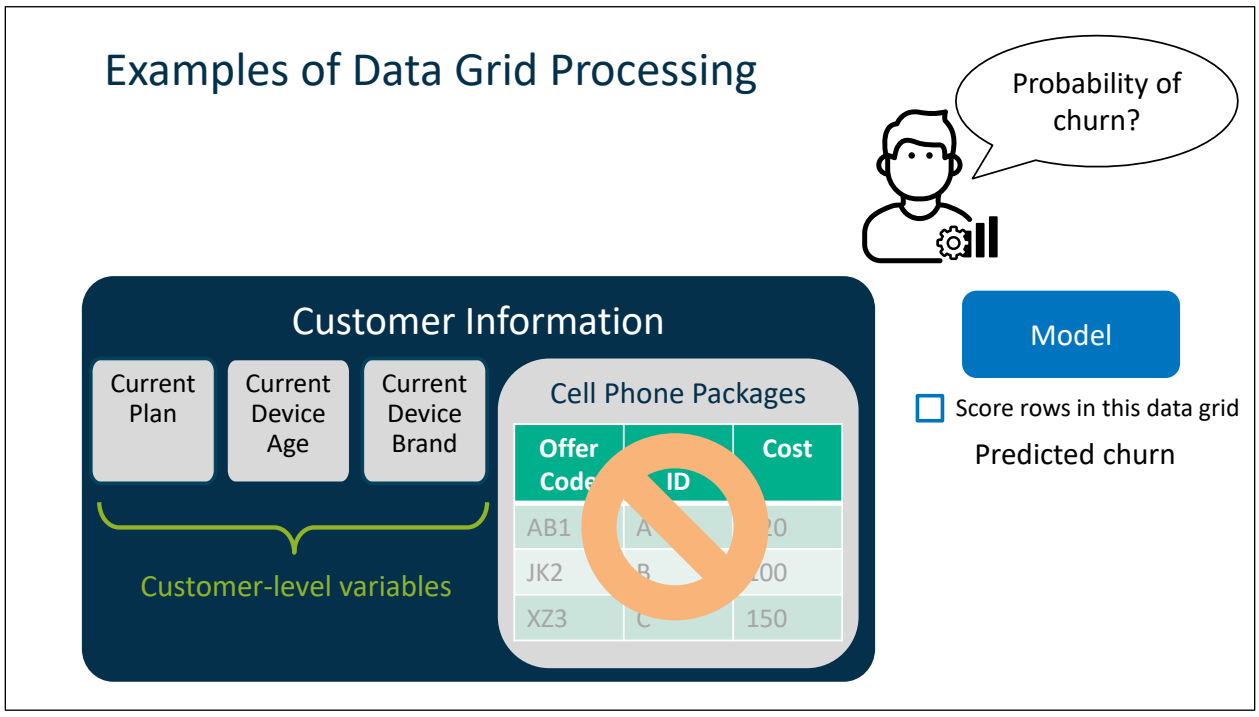


If you want to predict the propensity of acceptance of each package, you could use a model developed for this purpose and score rows in the data grid.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

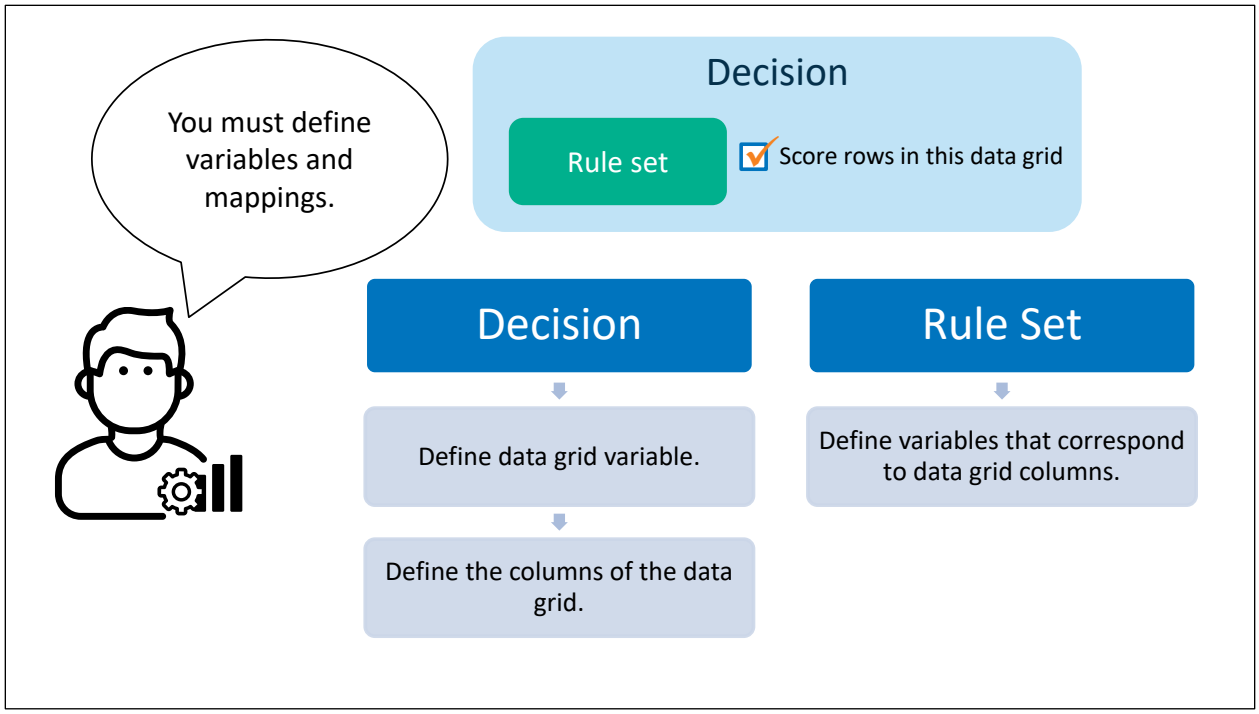


The input variables for the model could include customer-level variables that are not part of the data grid.

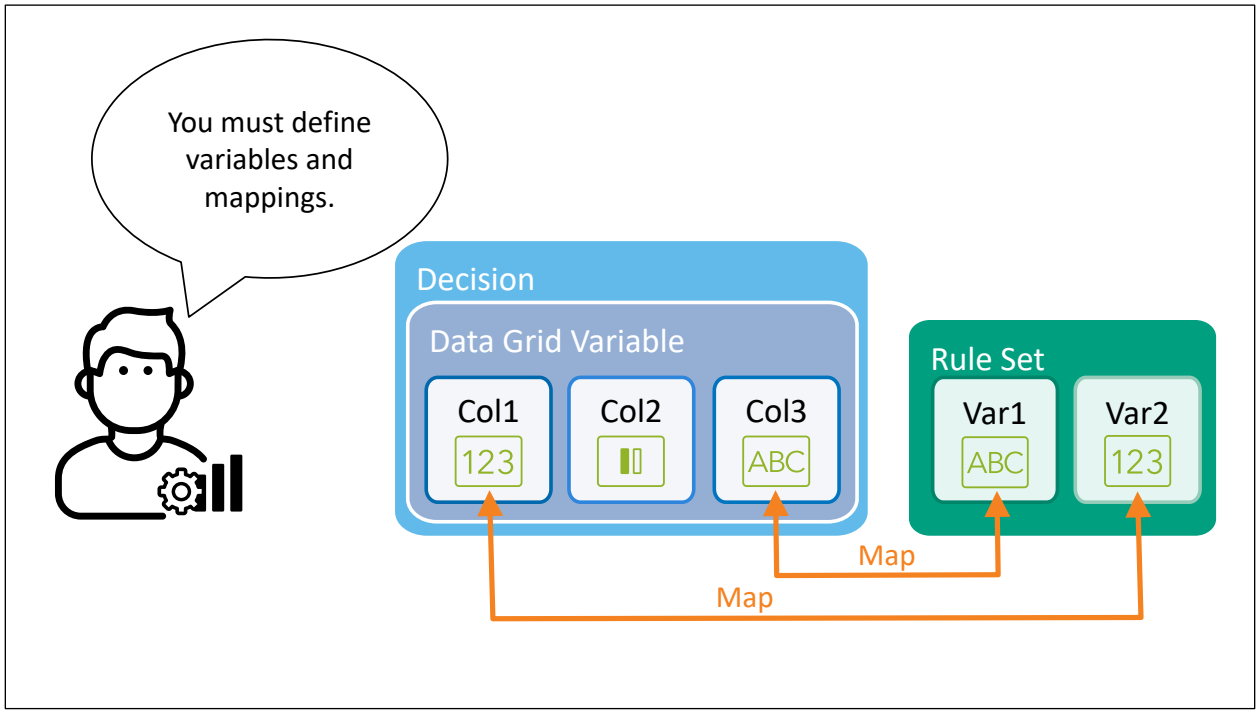


Alternatively, suppose that you want to predict the churn probability for a customer. A model that was built for this purpose uses customer-level variables only as input, and does not include any variables from the cell phone packages data grid. Therefore, you would not choose to score rows in the data grid.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

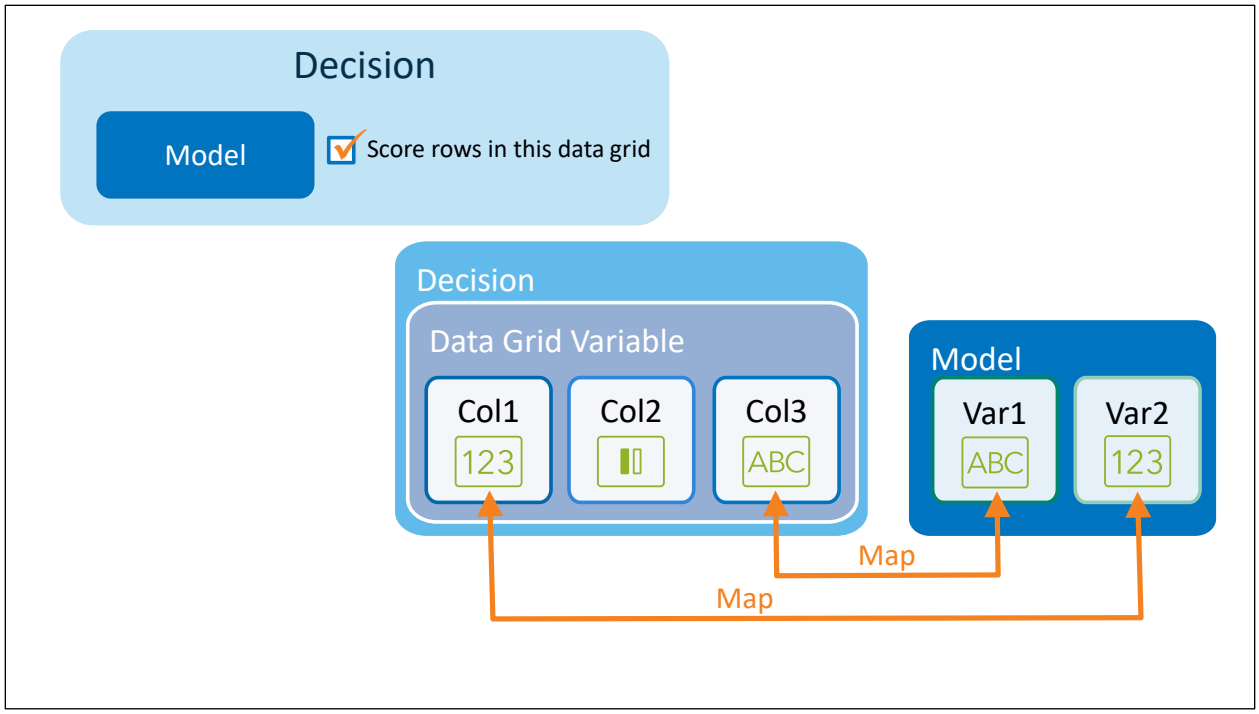


When you enable the option to score rows in a data grid with a rule set, you must define the variables and mappings. In the decision, you define the data grid variable, and you specify the data grid columns as properties of the data grid variable. In the rule set, you define variables that correspond to the data grid columns that you want to process.

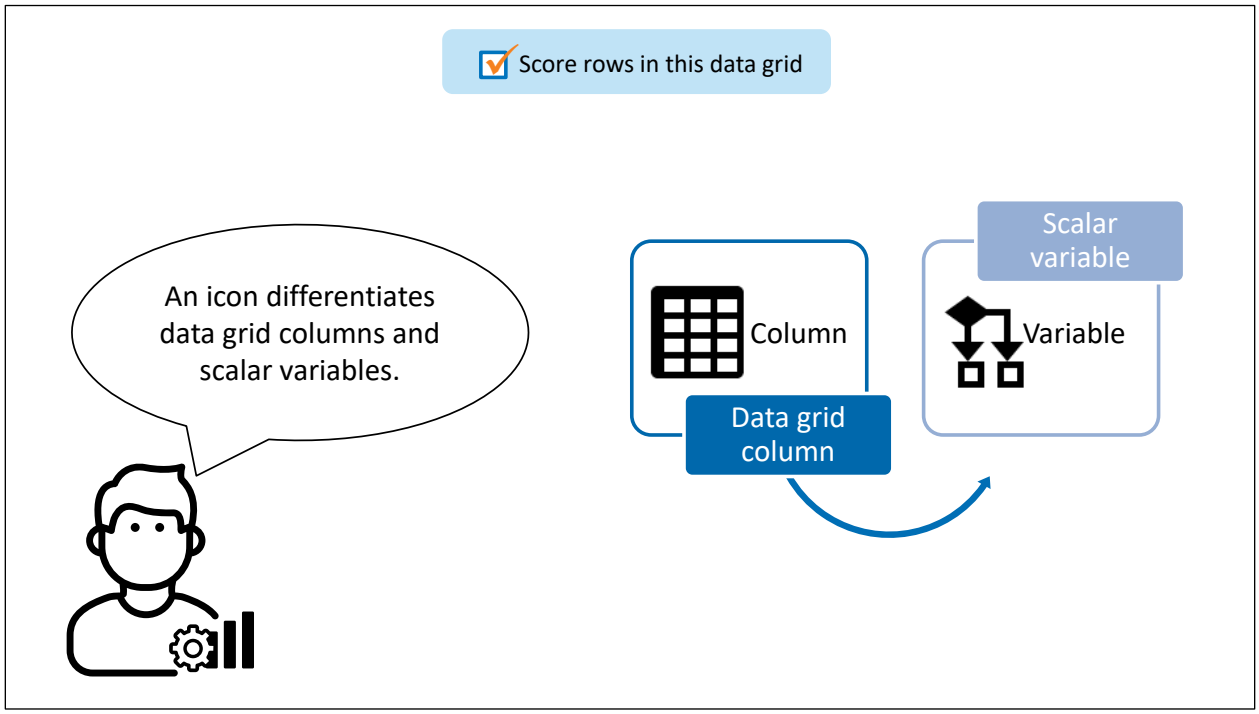


Then you map the columns of the data grid to variables in the rule set. The columns must be of the same data type as the variables. You do not necessarily need to define the data grid variable itself in the rule set.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

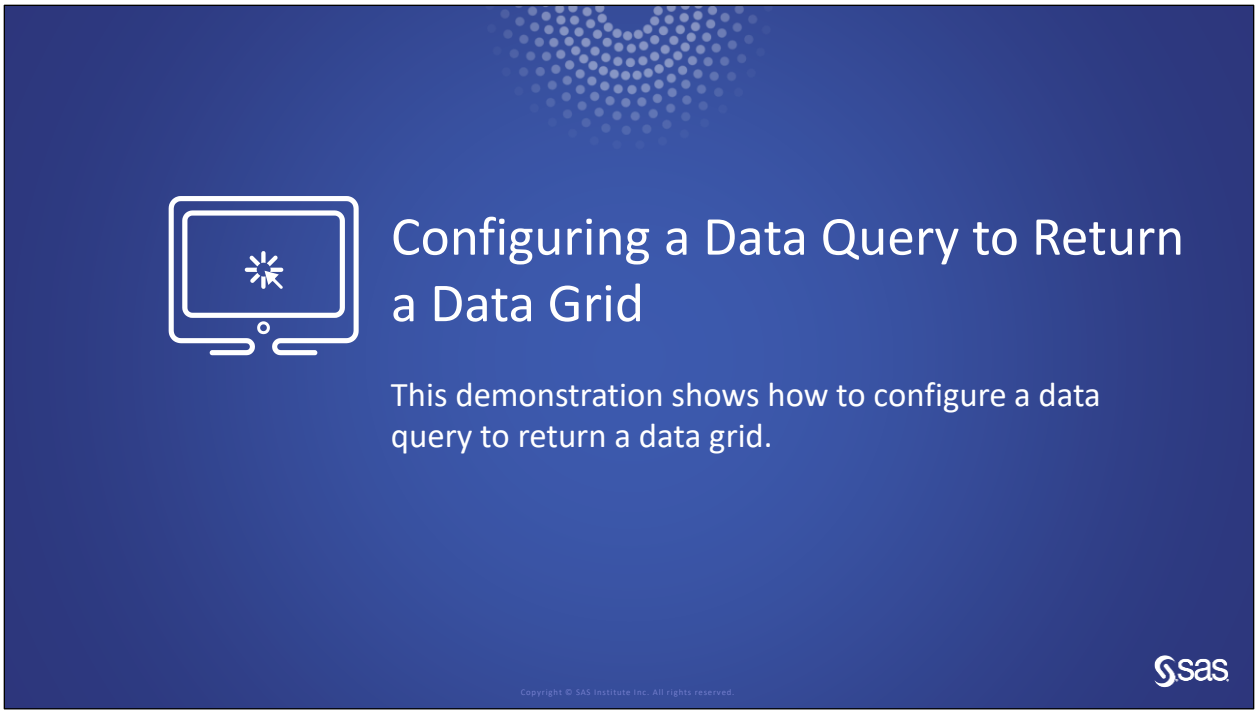


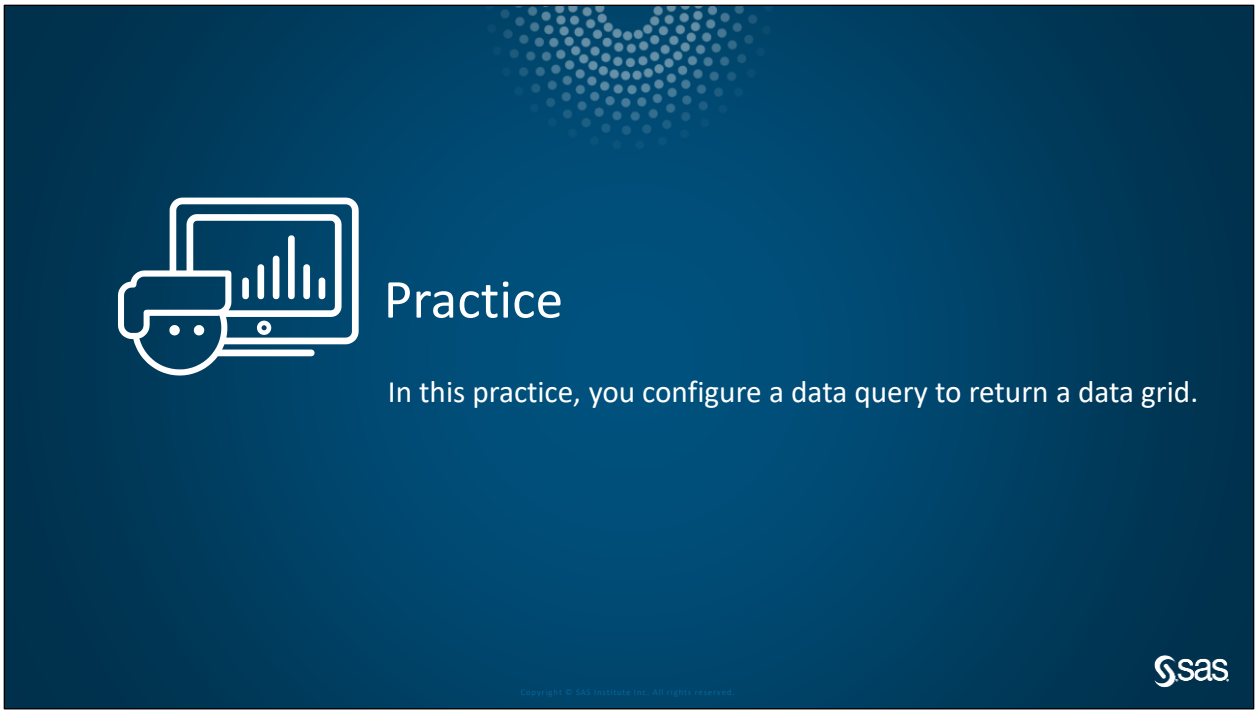
You can use the same approach with models. Models cannot include data grid variables.



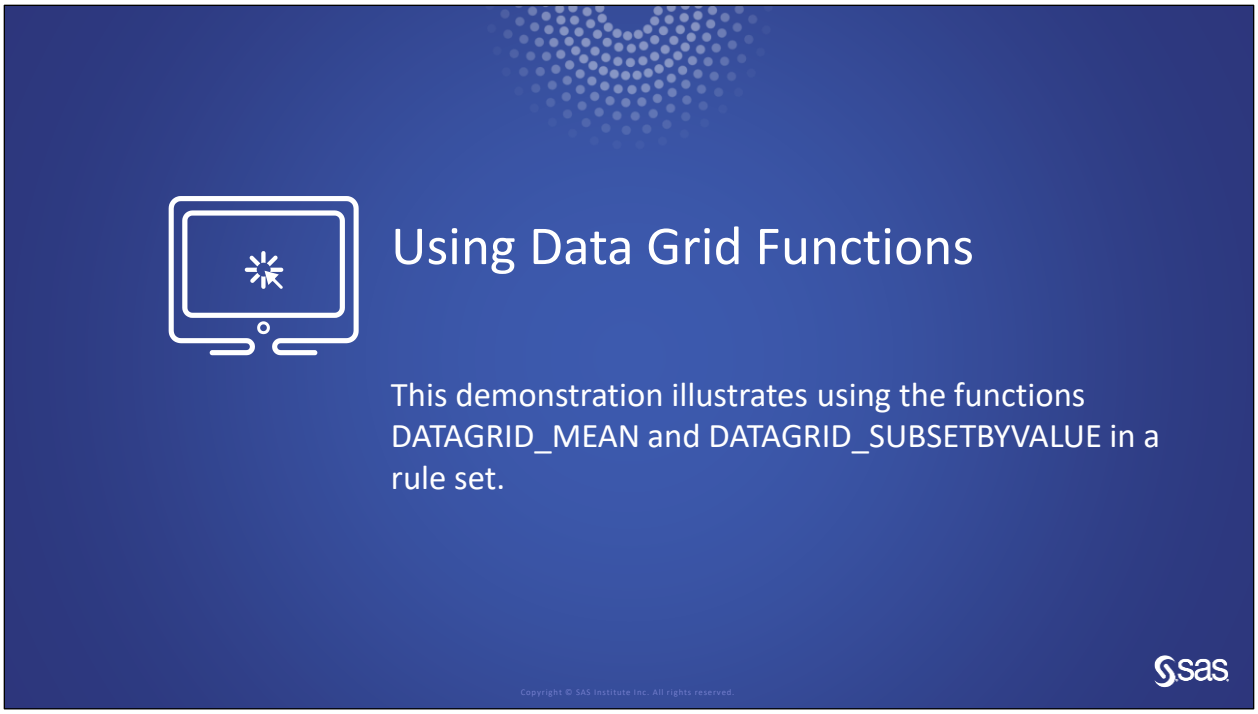
When you map data grid columns and scalar variables, an icon next to the variable name in the properties pane for the object enables you to differentiate between the two.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



3.05 Activity

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the **L** at the top right of the interface and select **Help Center**. Locate the documentation for the DATAGRID_SORT function. What value do you specify for the sort_order argument to sort a data grid in descending order?

Hint: Search for the text **datagrid_sort** and click the link to the documentation for the function.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.05 Activity – Correct Answer

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the **L** at the top right of the interface and select **Help Center**. Locate the documentation for the DATAGRID_SORT function. What value do you specify for the sort_order argument to sort a data grid in descending order?

Hint: Search for the text **datagrid_sort** and click the link to the documentation for the function.

Answer: You specify D for the sort order argument to sort in descending order.

Practice

In this practice, you use the data grid functions DATAGRID_MAX to calculate the maximum value of a data grid column and DATAGRID_SORT create a sorted copy of a data grid.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Scoring Rows in a Data Grid

This demonstration illustrates using a rule set to score rows in a data grid and calculate a discount price for products offered by vendors with an average price of more than 150.

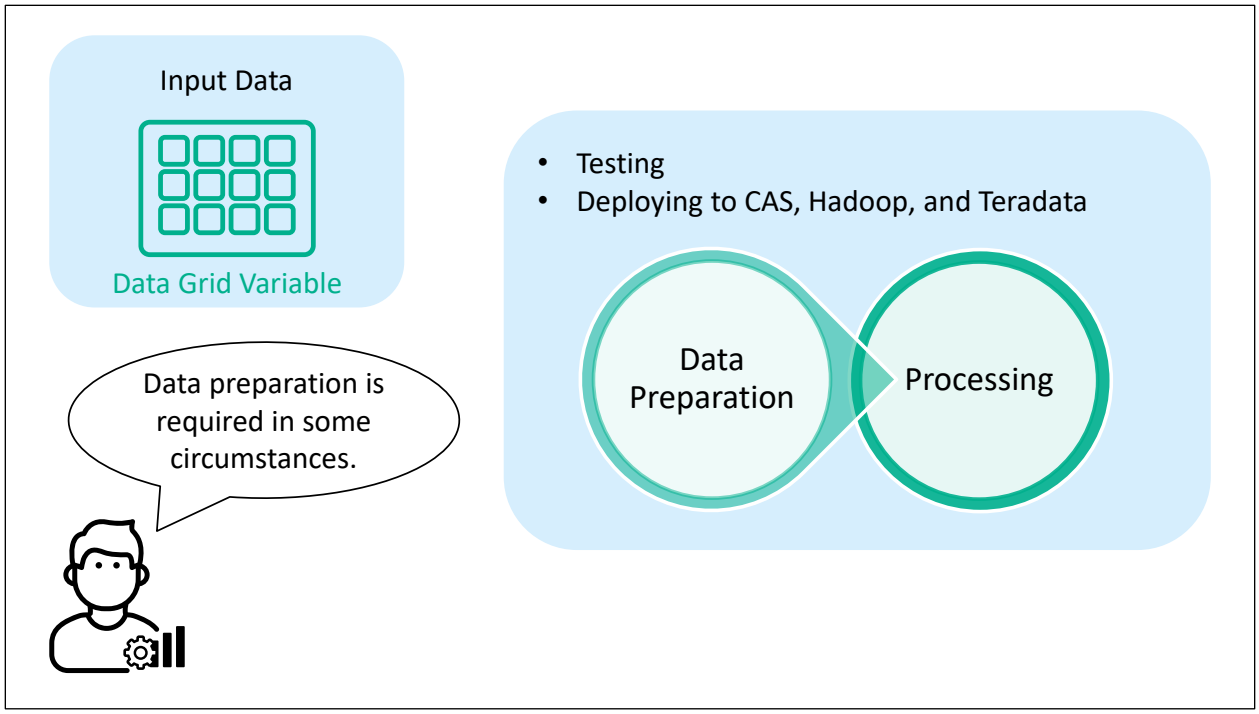
Copyright © SAS Institute Inc. All rights reserved.

Practice

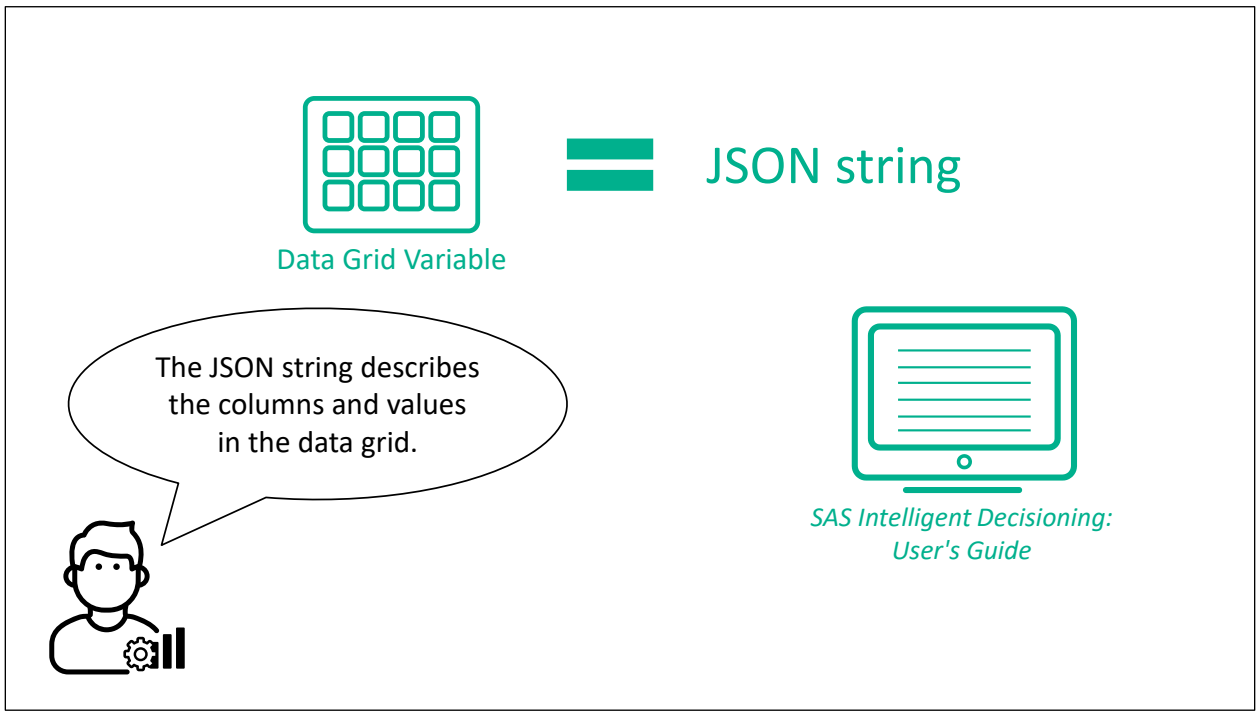
In this practice, you create a rule set to score rows in a data grid to calculate a commission for products with a price of more than 200.

Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

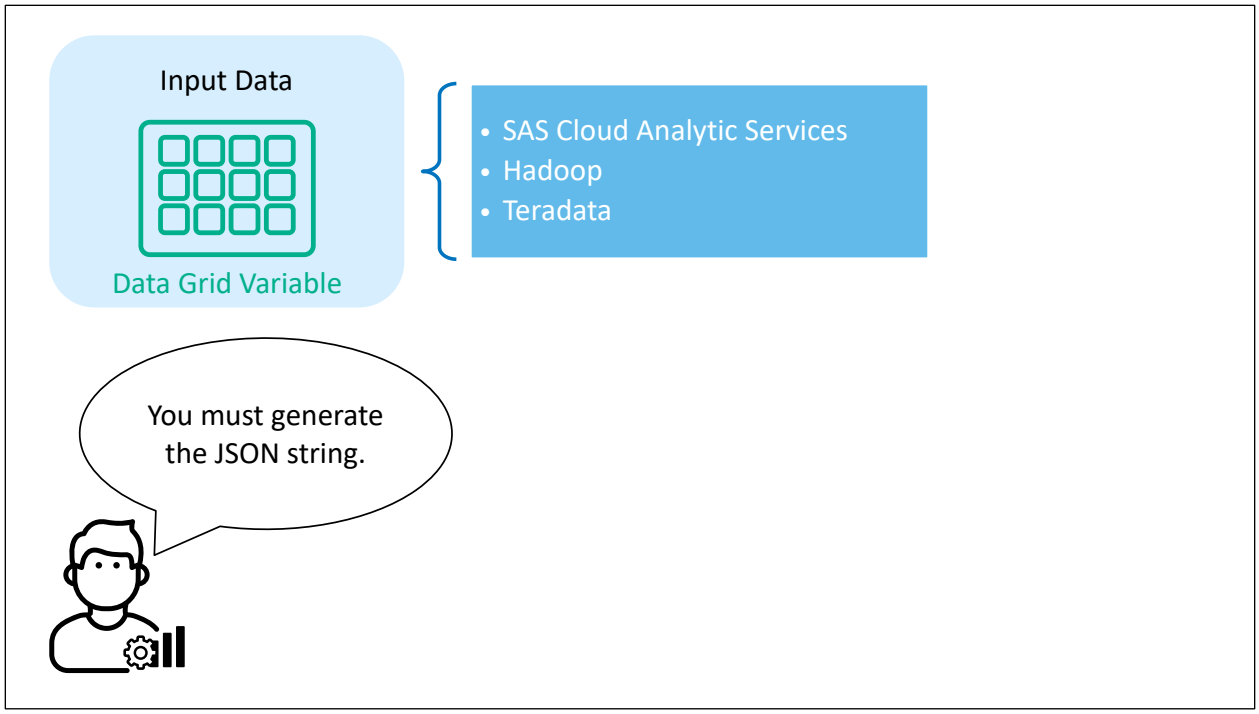


When the input data to your decision or rule set includes data grids, some steps to prepare for processing are required in some circumstances.

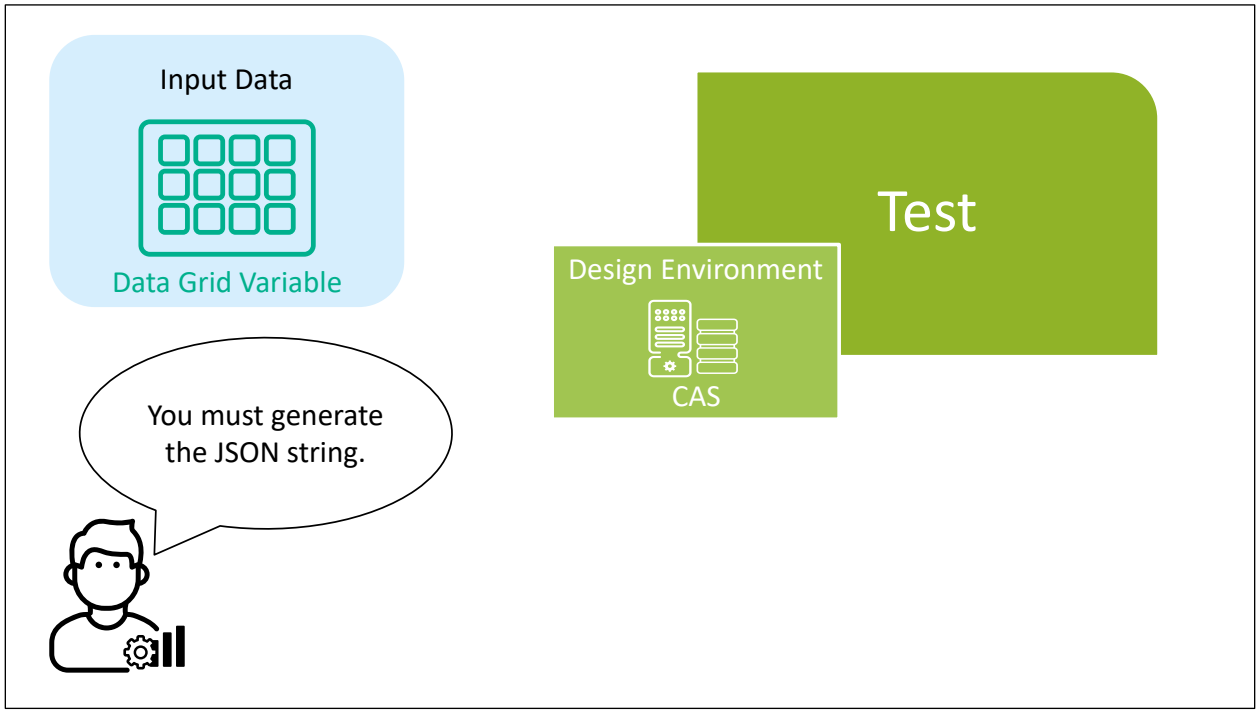


Data grids are represented as JSON strings. The JSON string includes metadata that describes the columns in the data grid along with the values themselves. JSON (or Javascript Object Notation) is a format for storing and transporting data. Refer to *SAS Intelligent Decisioning: User's Guide* for specifics about the formatting of data grid JSON strings.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

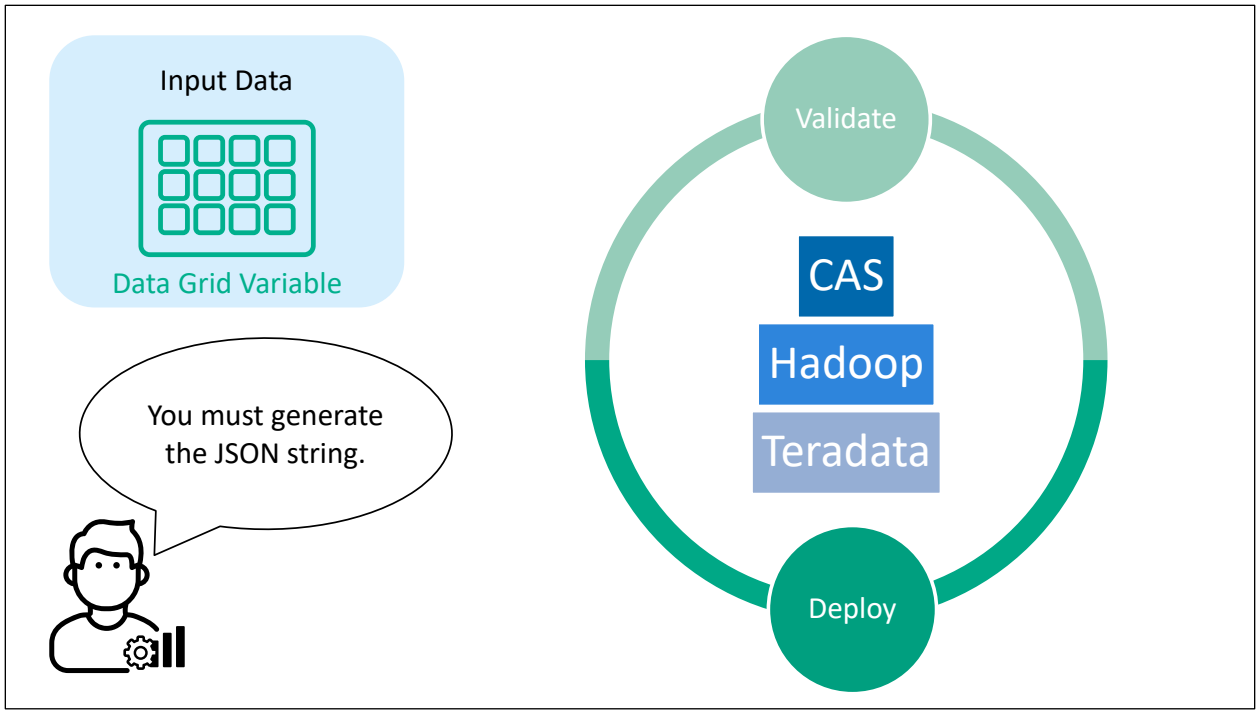


When you are working with decisions and rule sets where the input data includes data grids, you need to generate JSON strings for data grid variables anytime that you are working with CAS, Hadoop, or Teradata.

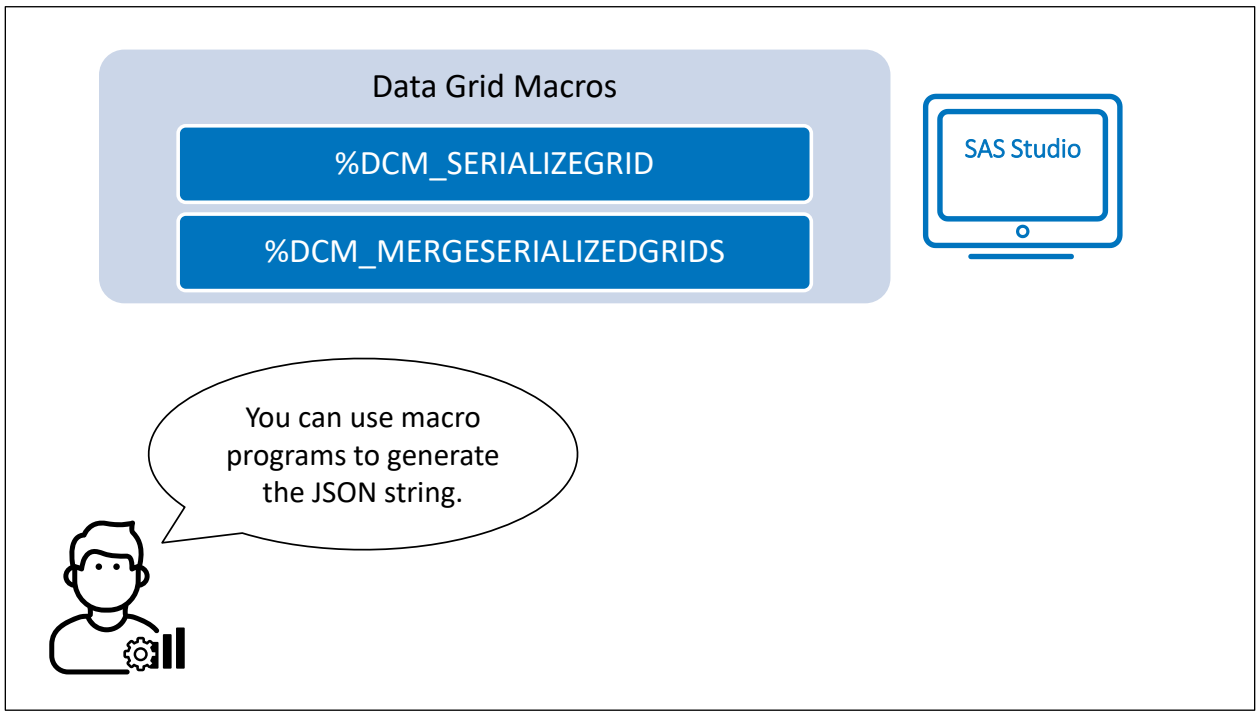


Recall that testing is always performed in CAS. Therefore, you must generate JSON strings for any data grid variables in input data when testing decisions or rule sets.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

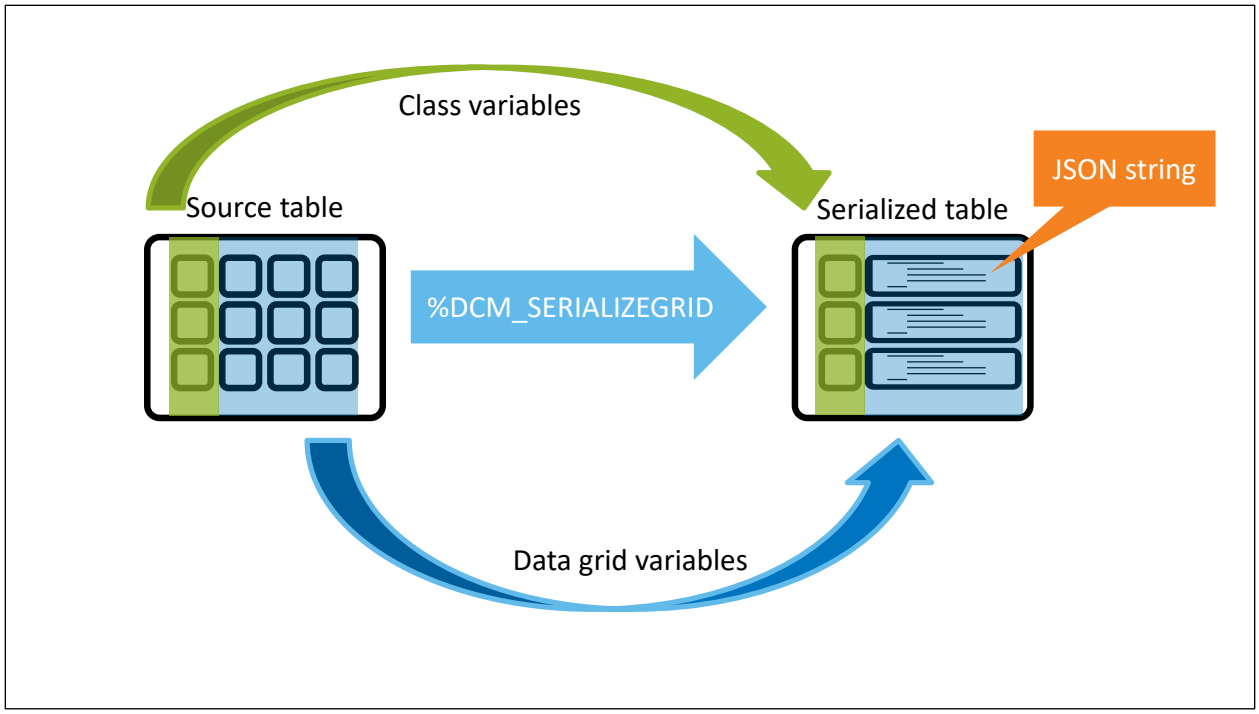


You must also generate JSON strings for any input data grid variables when you want to validate or deploy decisions and rule sets using CAS, Hadoop, or Teradata.

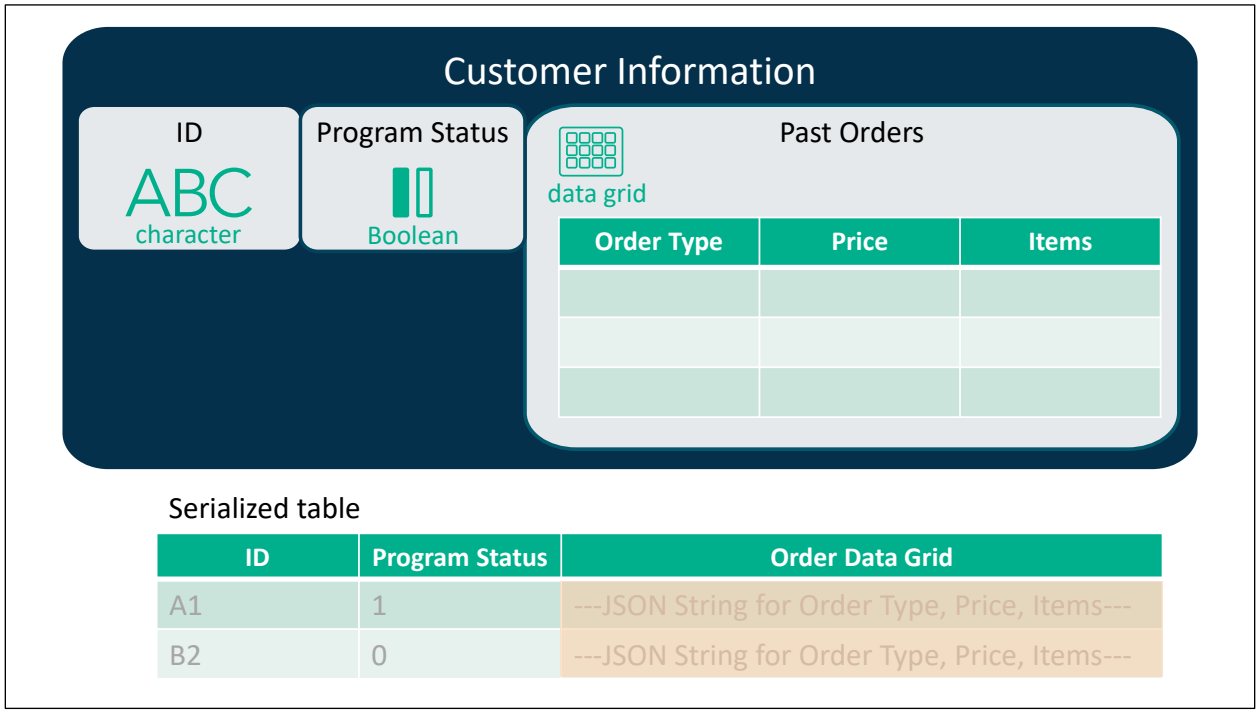


You can use two macro programs provided with SAS Intelligent Decisioning to generate JSON strings for data grid variables. You can use SAS Studio to run these programs.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

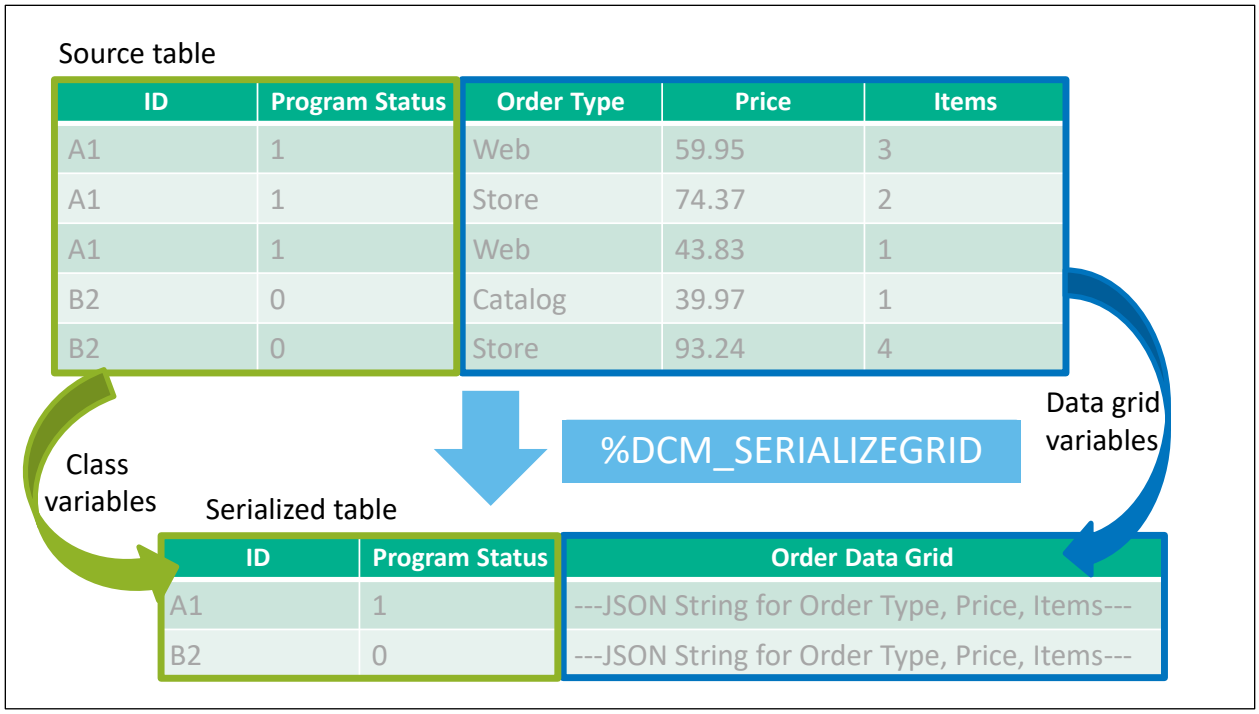


The first of these macros creates a table in which one of the columns contains a JSON string that represents a data grid. You specify the variables to be serialized into the data grid, along with any class variables to be preserved as separate columns.

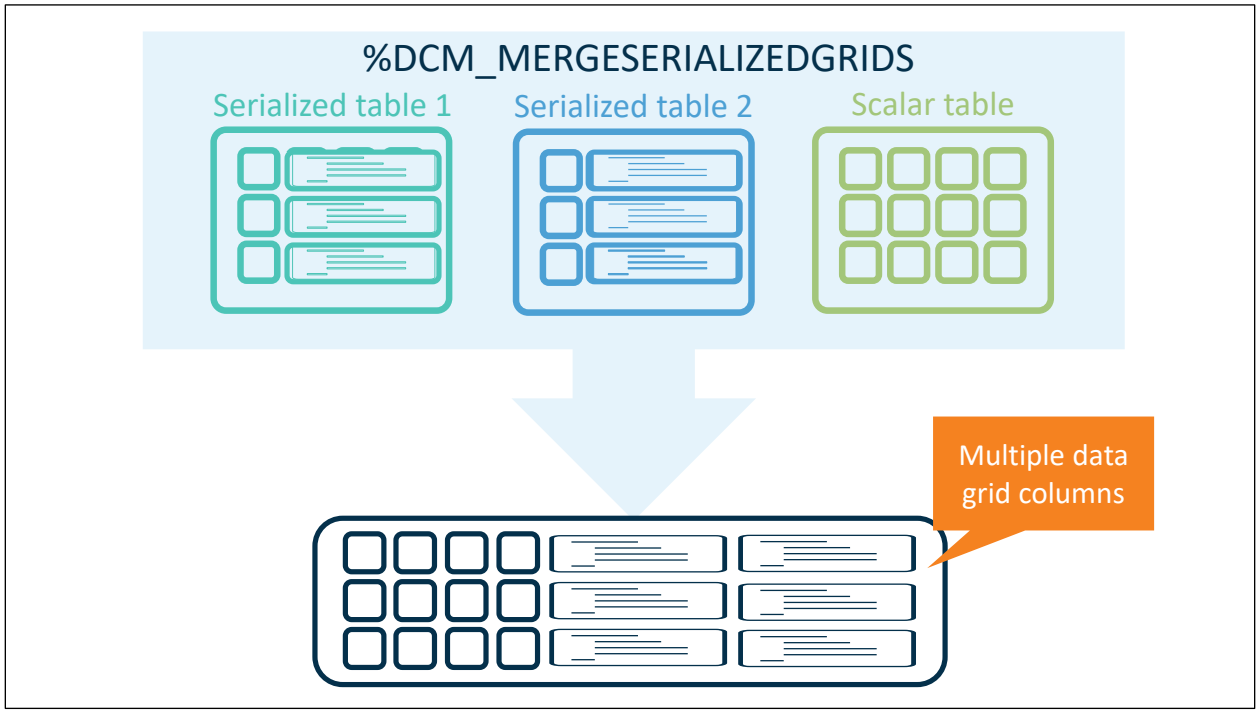


For example, consider the customer information table that we discussed earlier that includes ID and status variables along with a data grid for past orders. For processing purposes, the data grid is represented as a JSON string in a serialized table.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

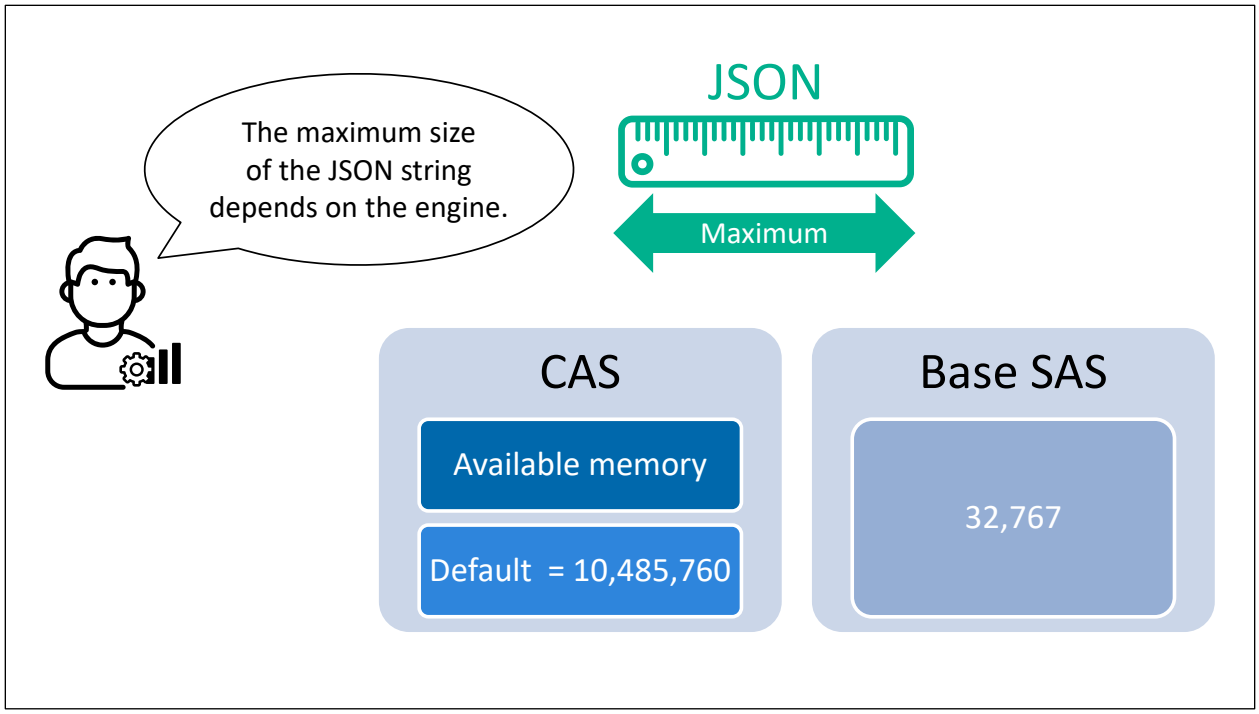


You could create this serialized table from a source table that contains all of the contributing columns using the macro. You specify that the **Order Type**, **Price**, and **Items** variables be serialized into a data grid, and **ID** and **Program Status** be included as class variables.

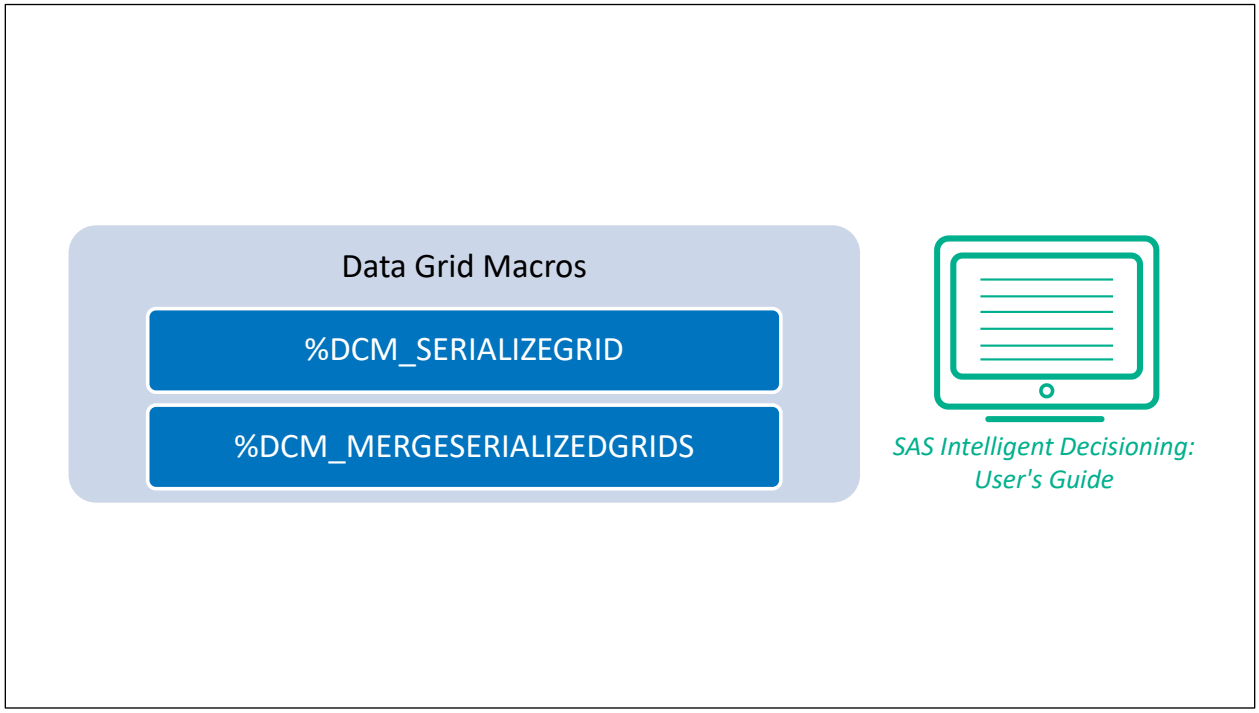


You can use the second macro when you need to create input data with multiple data grid columns. It merges serialized tables that have been created using the first macro with a table that contains scalar data to create an output table with multiple data grids.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

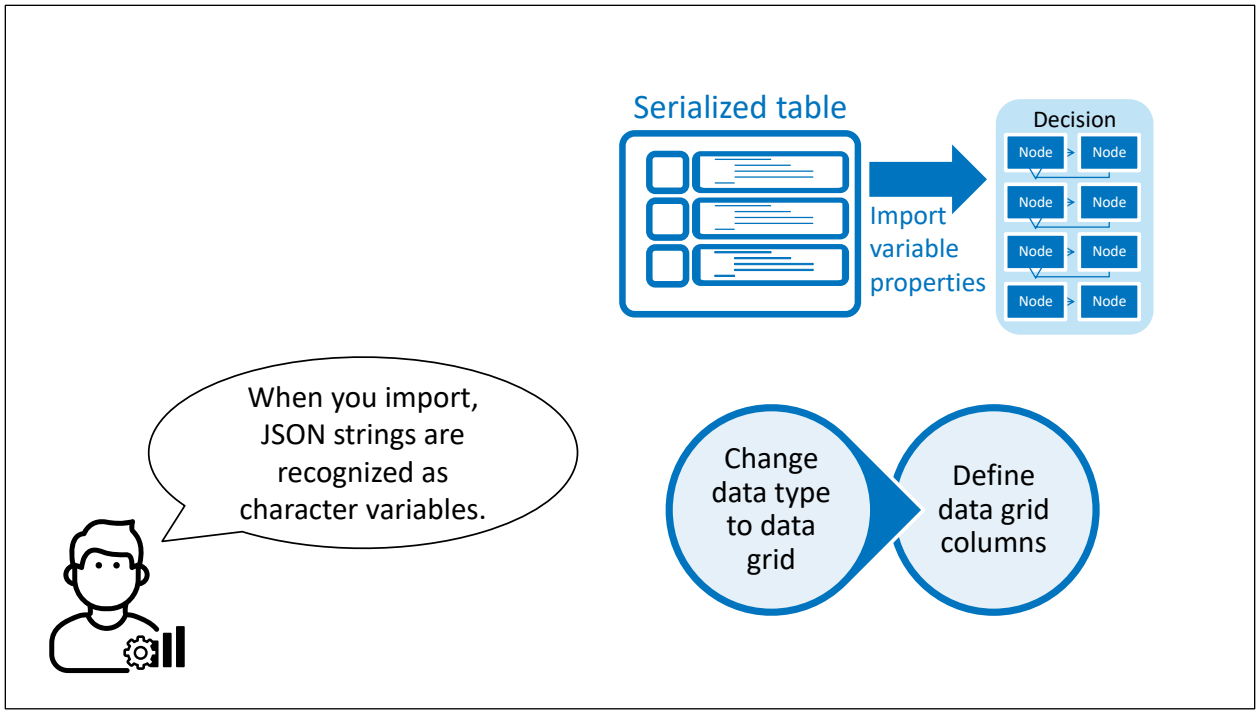


When you use the macros, the maximum size of the JSON string depends on the engine that writes the table. If you write the serialized data grid variable to a CAS table, the maximum size of the JSON string is determined by the amount of available memory. The default size is 10,485,760. For the Base SAS engine, the limit is 32,767. You can publish rule sets and decisions that use data grids to any destination. However, it is unlikely that objects that use data grids can be published to Teradata because of limitations on row sizes.

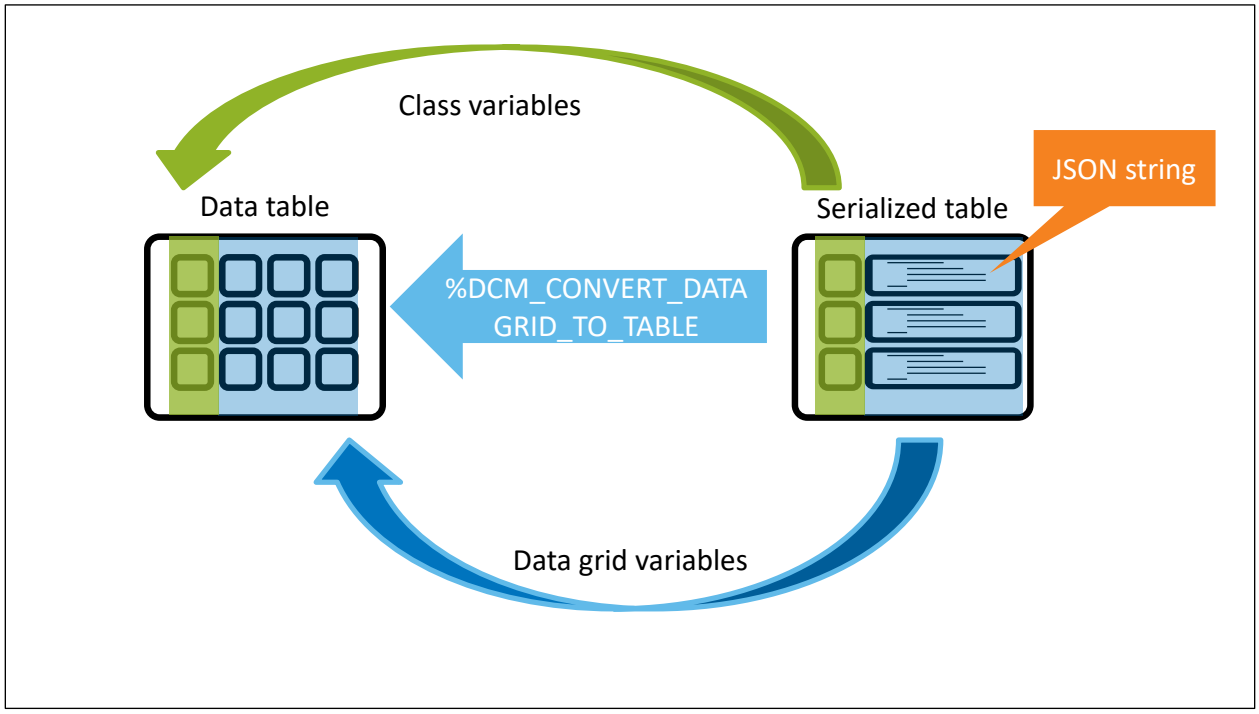


Refer to *SAS Intelligent Decisioning: User's Guide* for details about the syntax for these macros.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



When you import variable properties from a data source that was prepared using the data grid macros, the JSON strings are recognized as text and are assigned a data type of character. You must change the data type to data grid and define the columns of the data grid manually.



If you have the JSON strings in a table, you can reverse the direction and convert it into an original table format. The %DCM_CONVERT_DATAGRID_TO_TABLE macro converts each row in a data grid to a separate table. Working with the content of a data grid as tables enables you to more easily validate the content of the data grid and test custom code that uses the data grid.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.07 Activity

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the **L** at the top right of the interface and select **Help Center.** Locate the documentation for the %DCM_SERALIZEGRID macro. What arguments are required?

Hint: Search for the text **serialize** and click the link to the documentation for the macro.

3.07 Activity – Correct Answer

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the **L** at the top right of the interface and select **Help Center.** Locate the documentation for the %DCM_SERALIZEGRID macro. What arguments are required?

Hint: Search for the text **serialize** and click the link to the documentation for the macro.

Answer: The arguments GRIDCOLNAME, GRIDSOURCETABLE, and OUTPUTTABLE are required.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

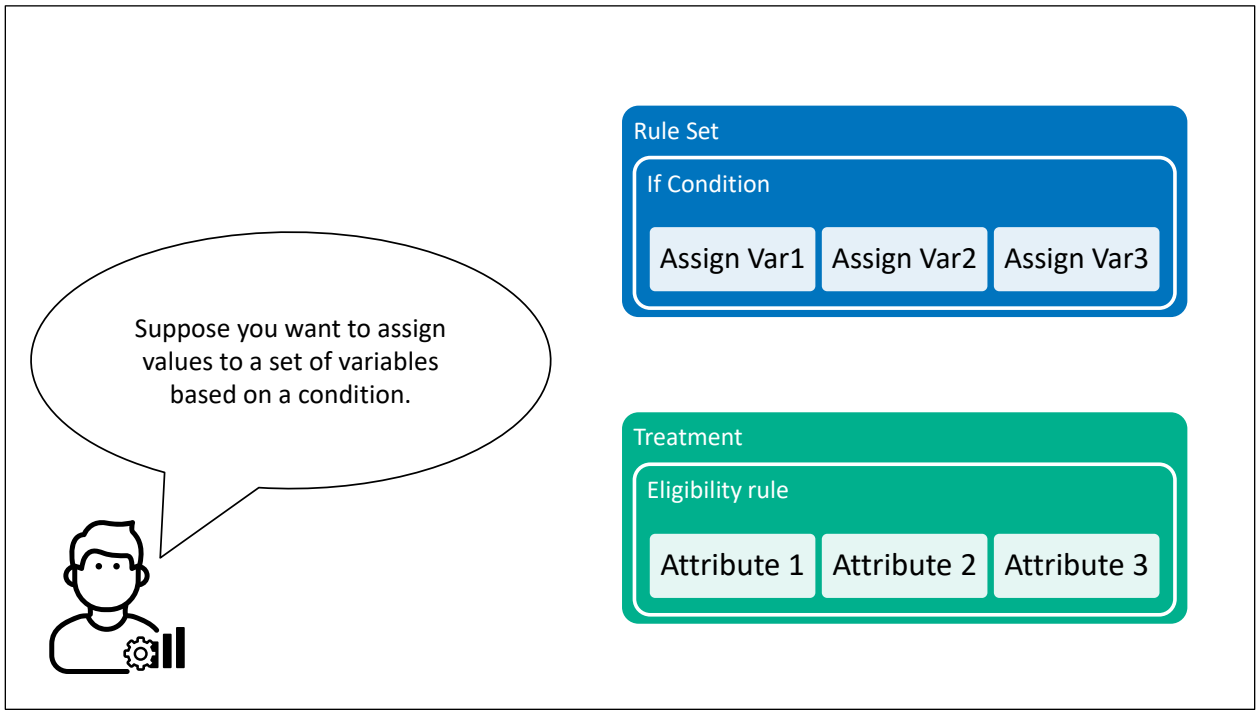
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Lesson 4 Treatments, Treatment Groups, and Arbitration

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

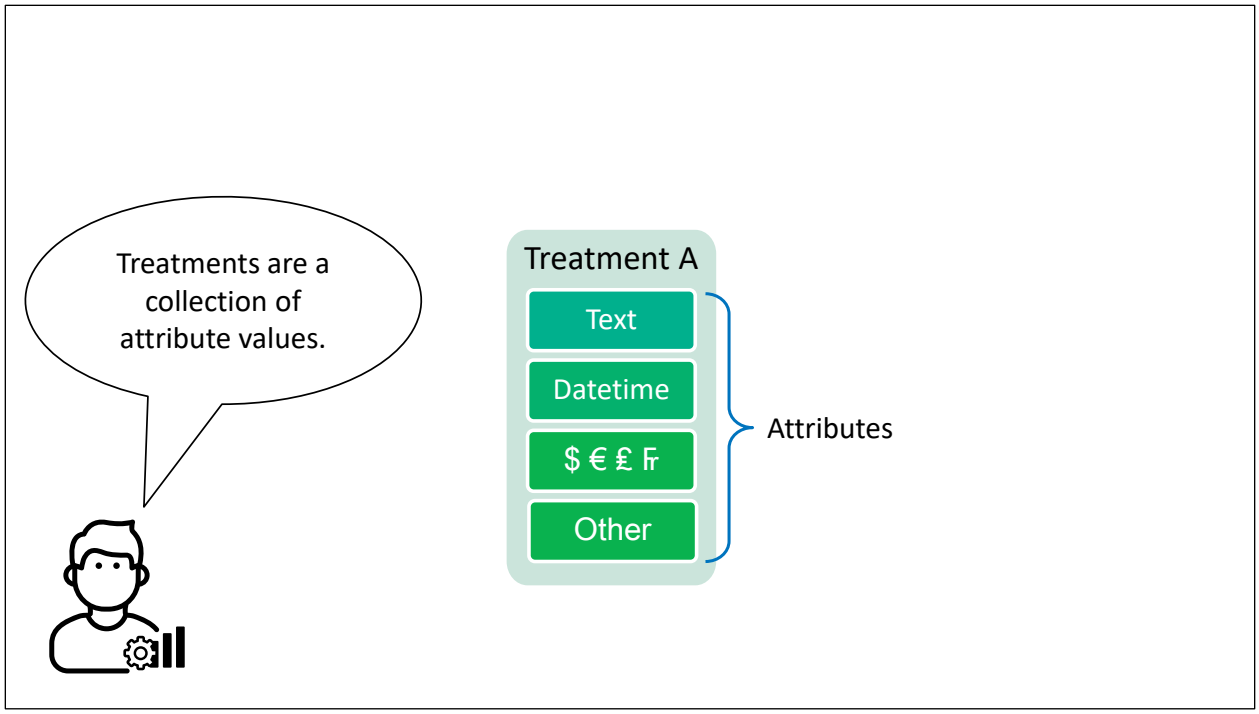
4.1 Treatments and Treatment Groups

The Big Picture

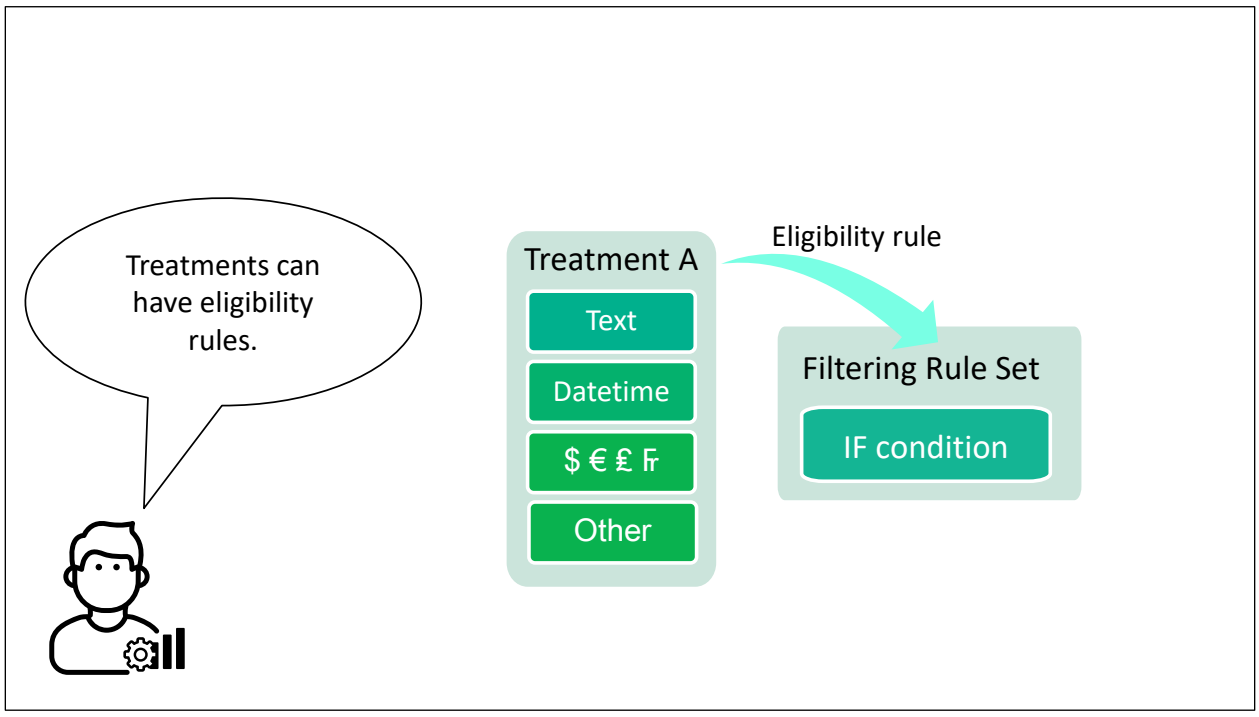


Suppose you want to assign values to a set of variables based on a condition. You could do so using a rule set with a condition and several assignments. You could also do so using treatments.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

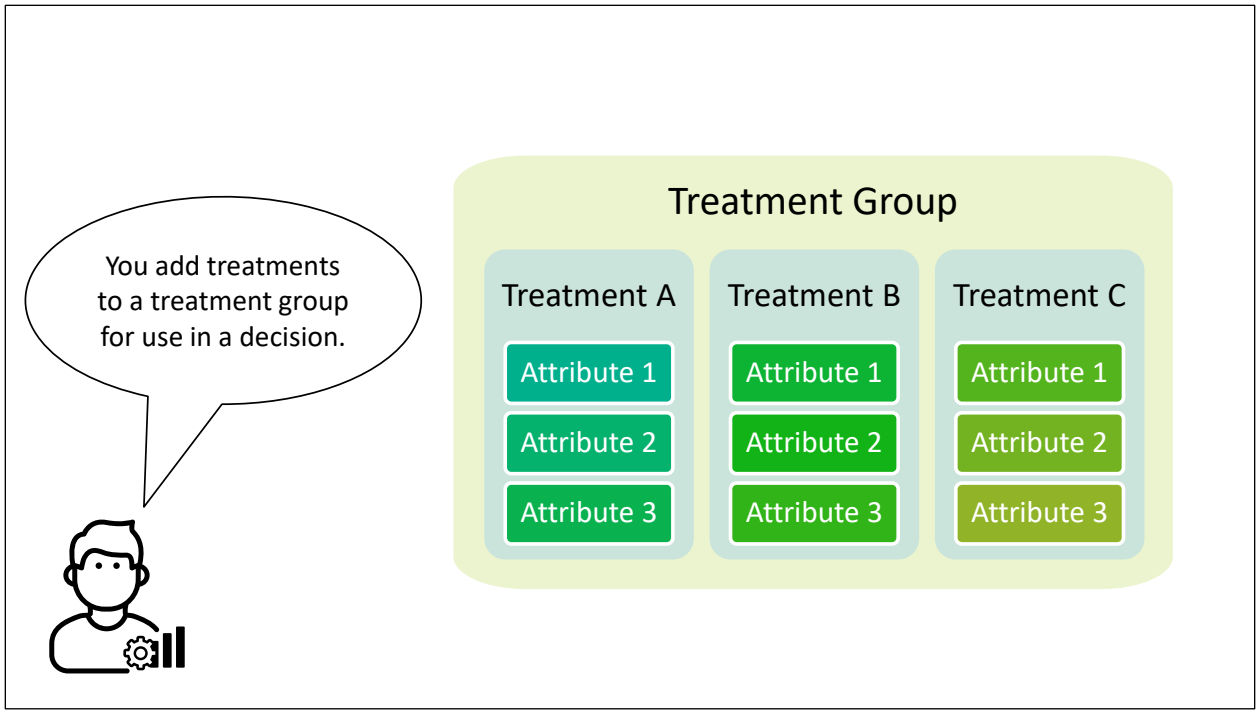


Treatments are a collection of attribute values. Attributes can contain different types of data such as descriptive text, datetime values, or monetary amounts.

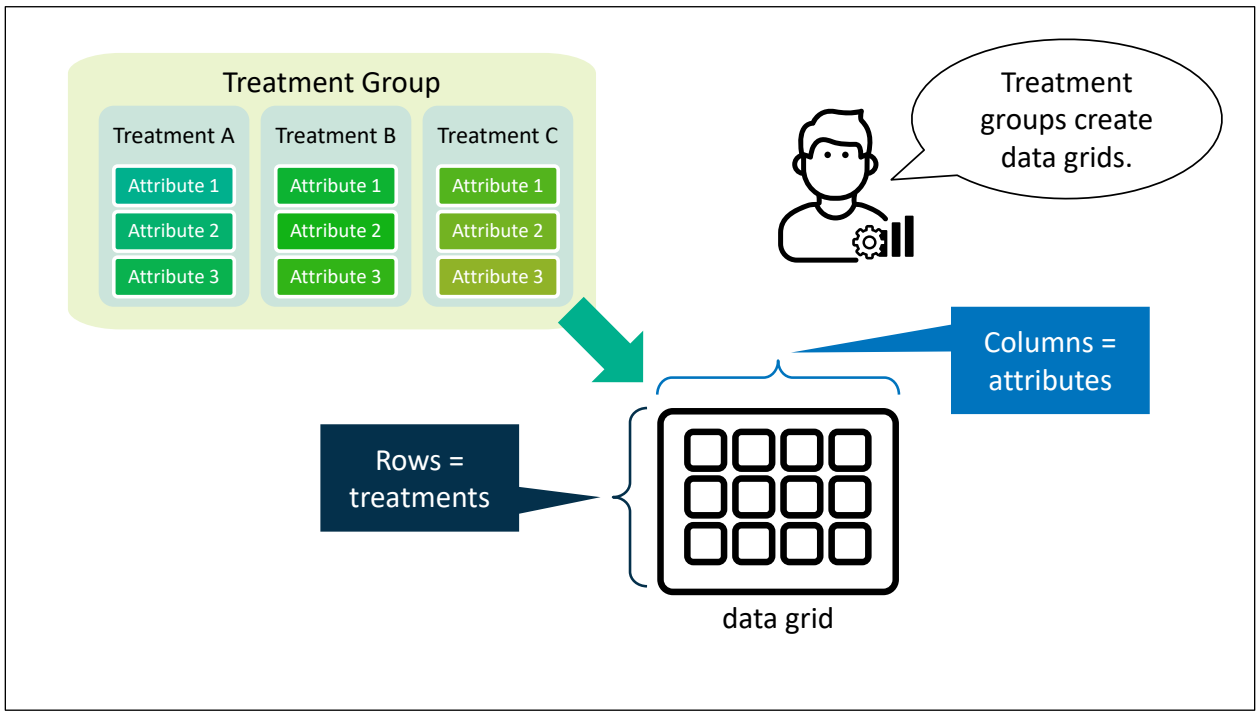


Treatments can have eligibility rules. Treatment eligibility rules are set using IF conditions in a filtering rule set.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

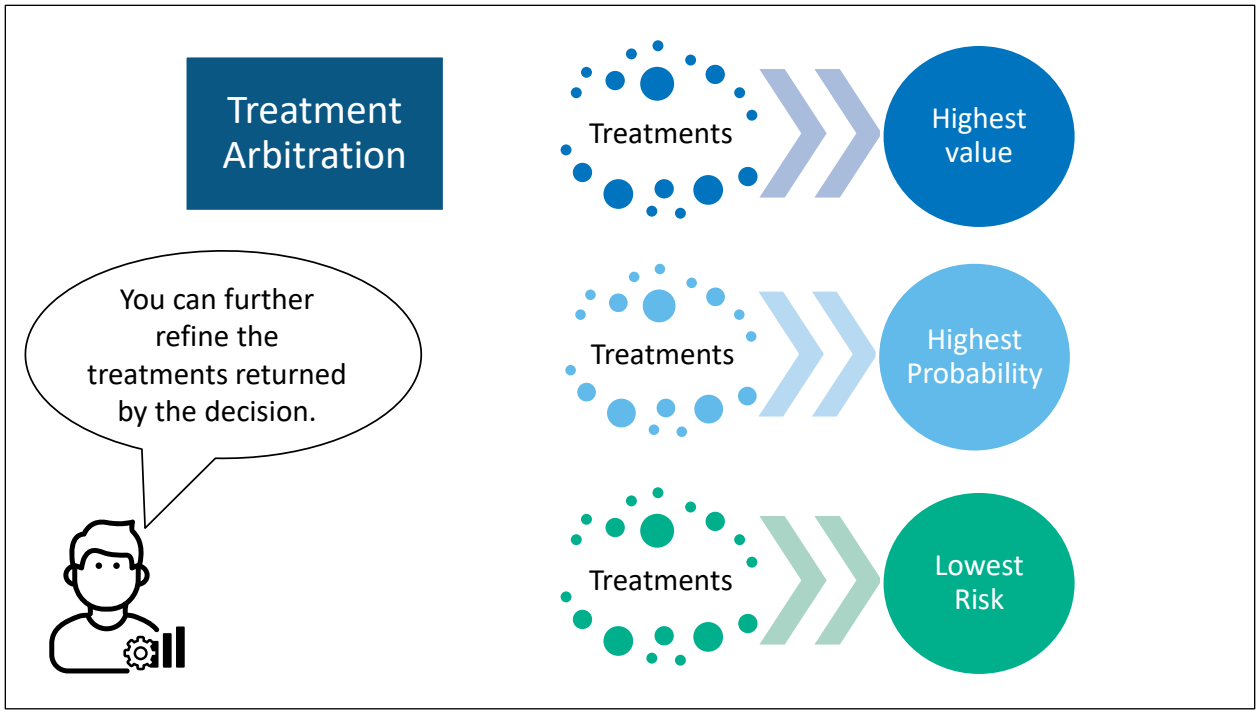


In order to use treatments in a decision, you add treatments to treatment group. A treatment group can contain one or more treatments.

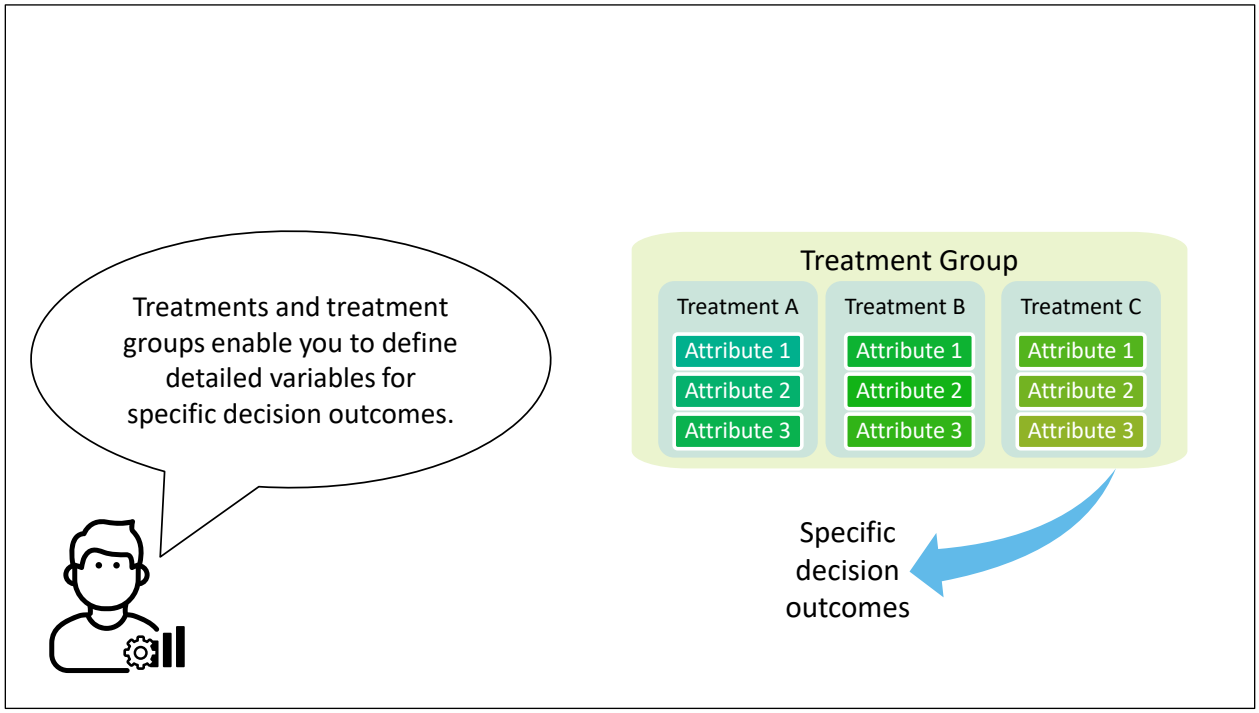


A treatment group in a decision creates a data grid variable in the decision. The data grid contains one row for each treatment in the group for which any eligibility rules are met. The columns of the data grid include the attributes that were defined for the treatment.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

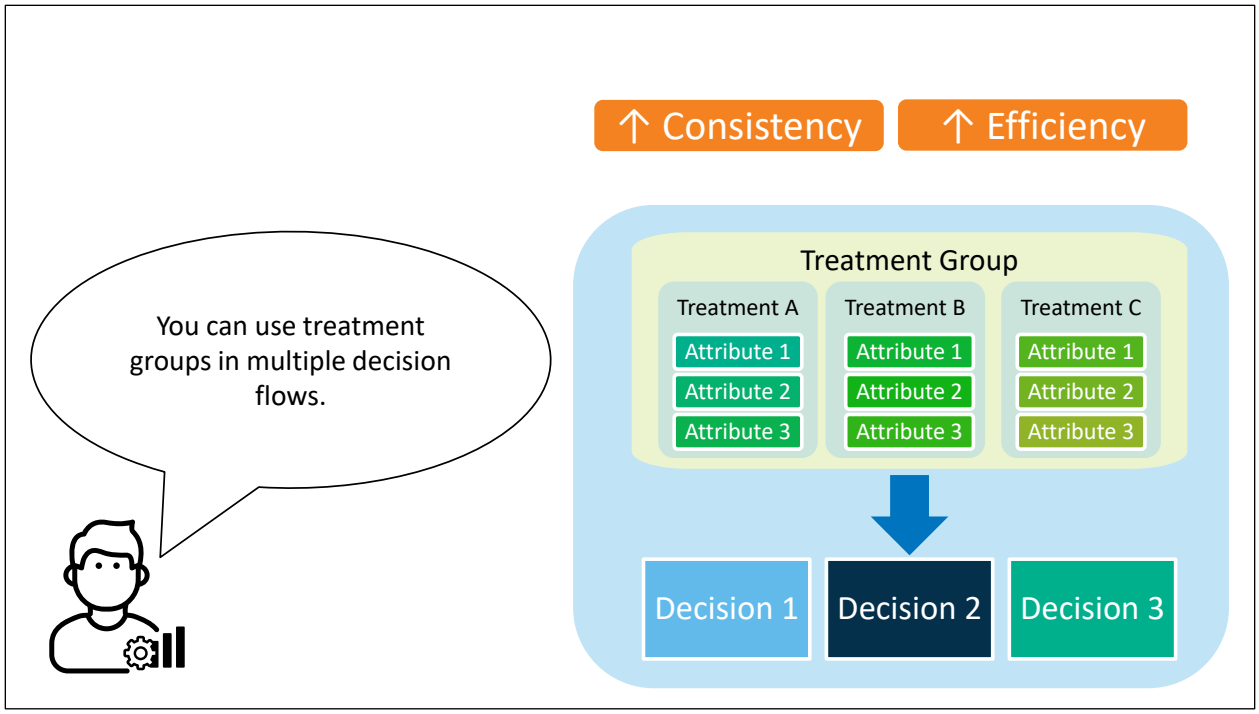


You can further refine the list of treatments returned by the decision. This process is known as treatment arbitration. For example, you could select a subset of treatments with the highest value or probability, or the lowest risk.



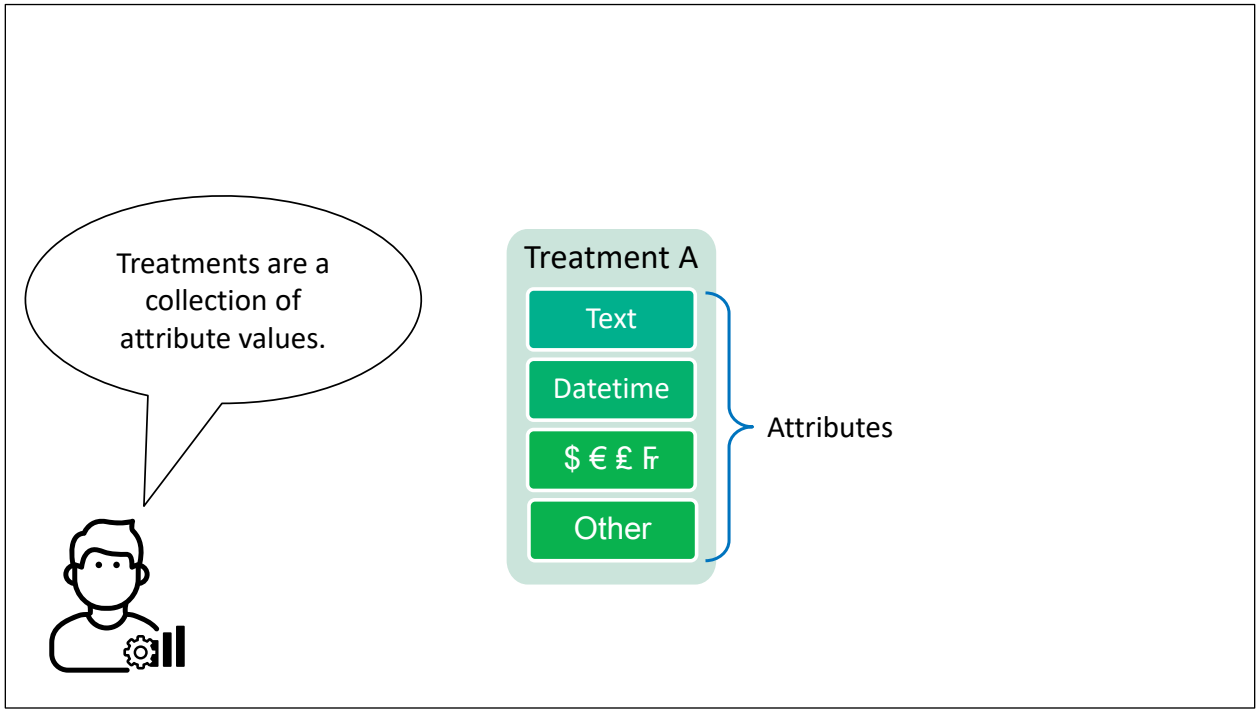
In summary, treatments and treatment groups enable you to define detailed variables for specific decision outcomes.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



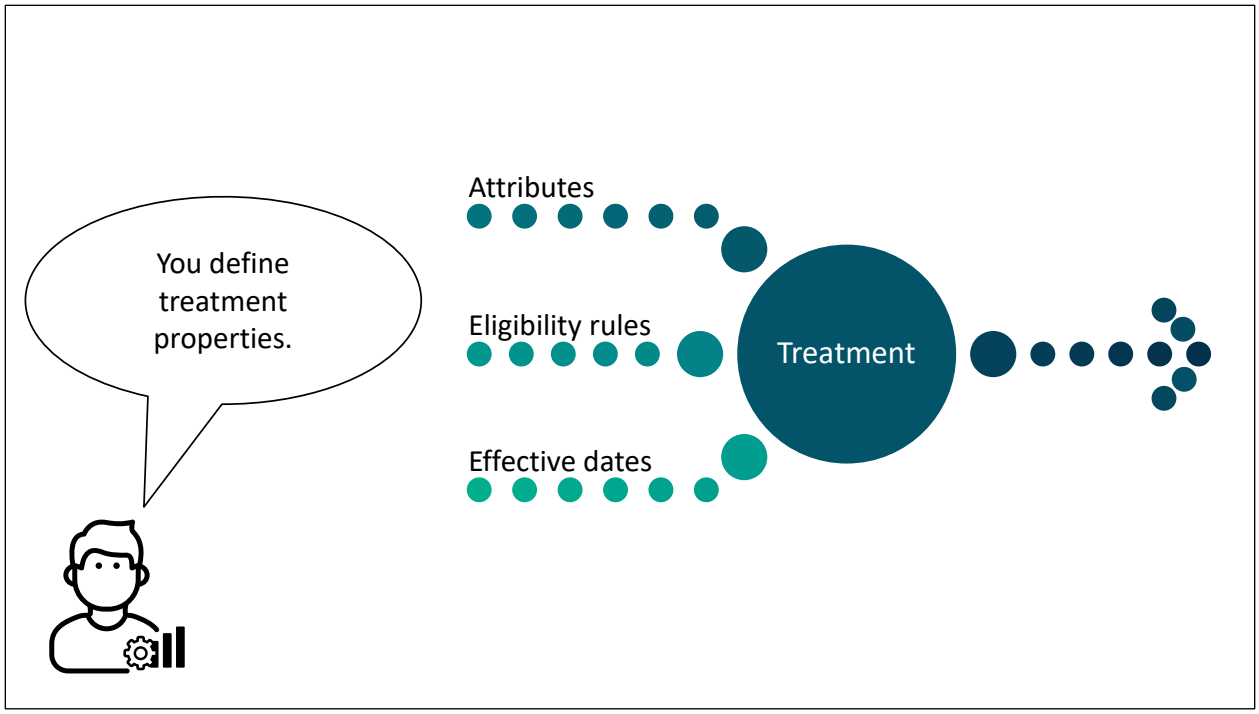
These can be used across multiple decision flows, promoting consistency and efficiency.

Treatments

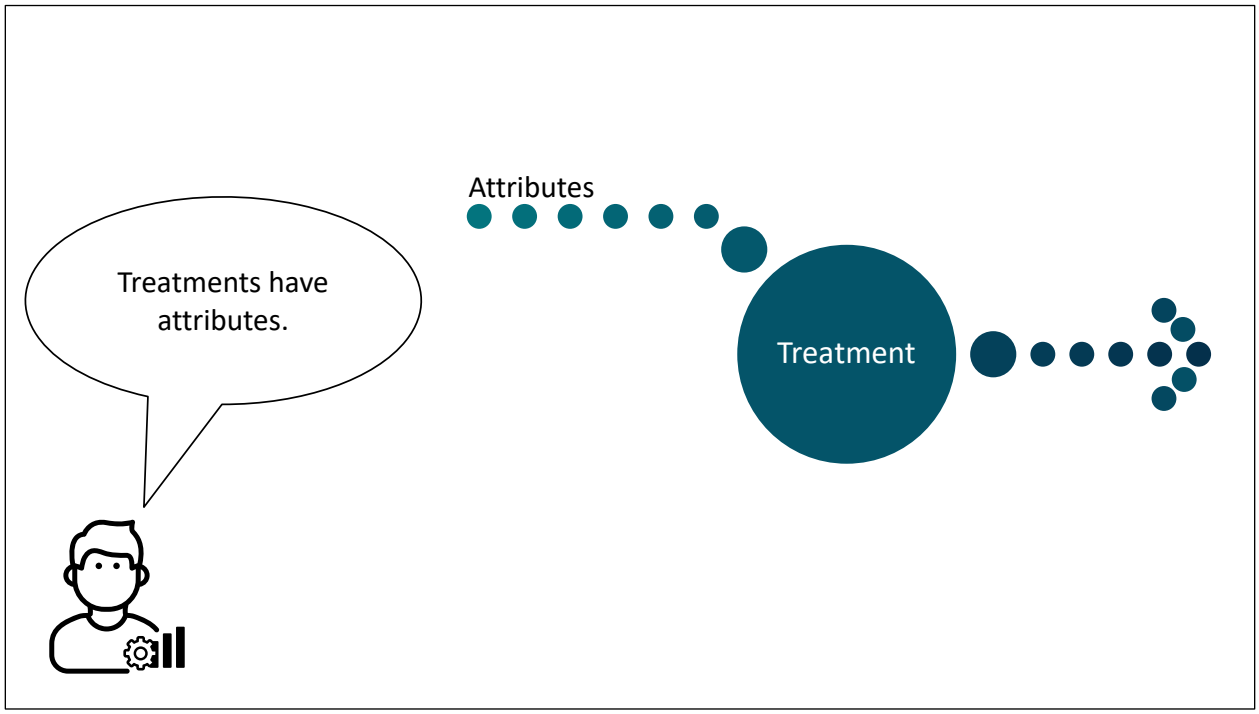


Recall that treatments are a collection of attribute values that correspond to possible decision outcomes.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

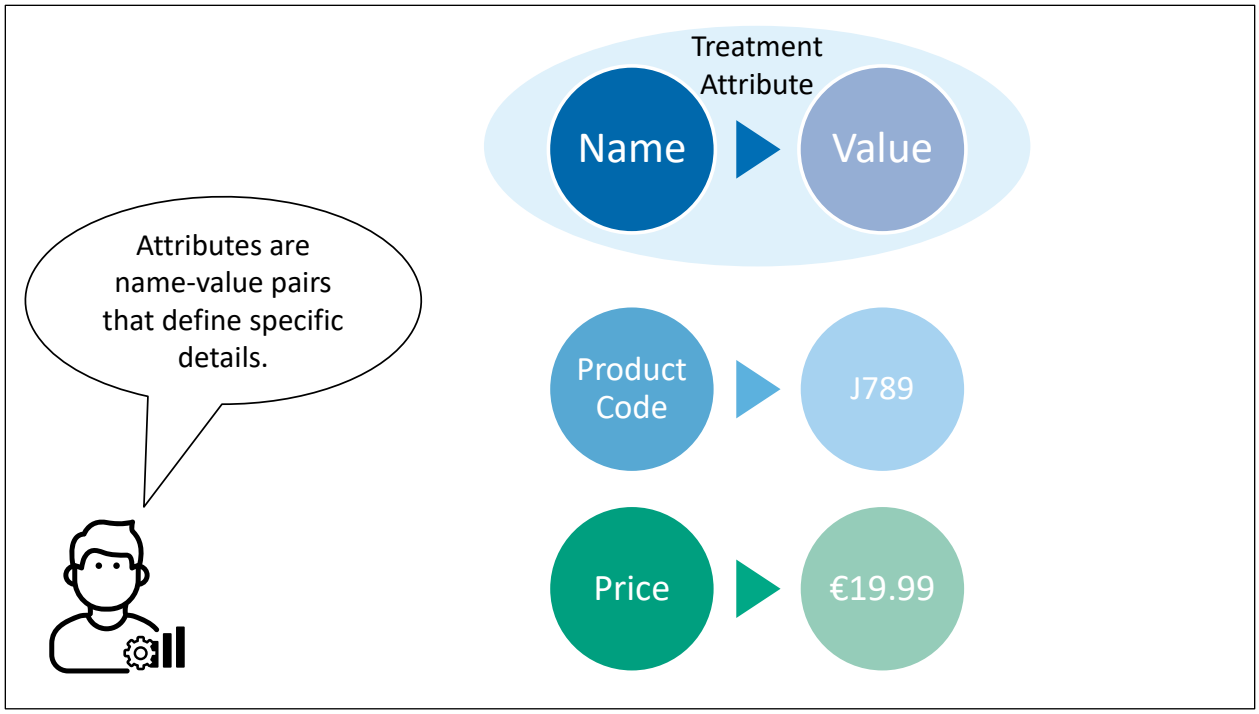


When you define a treatment, you can specify attributes, eligibility rules, and effective dates.

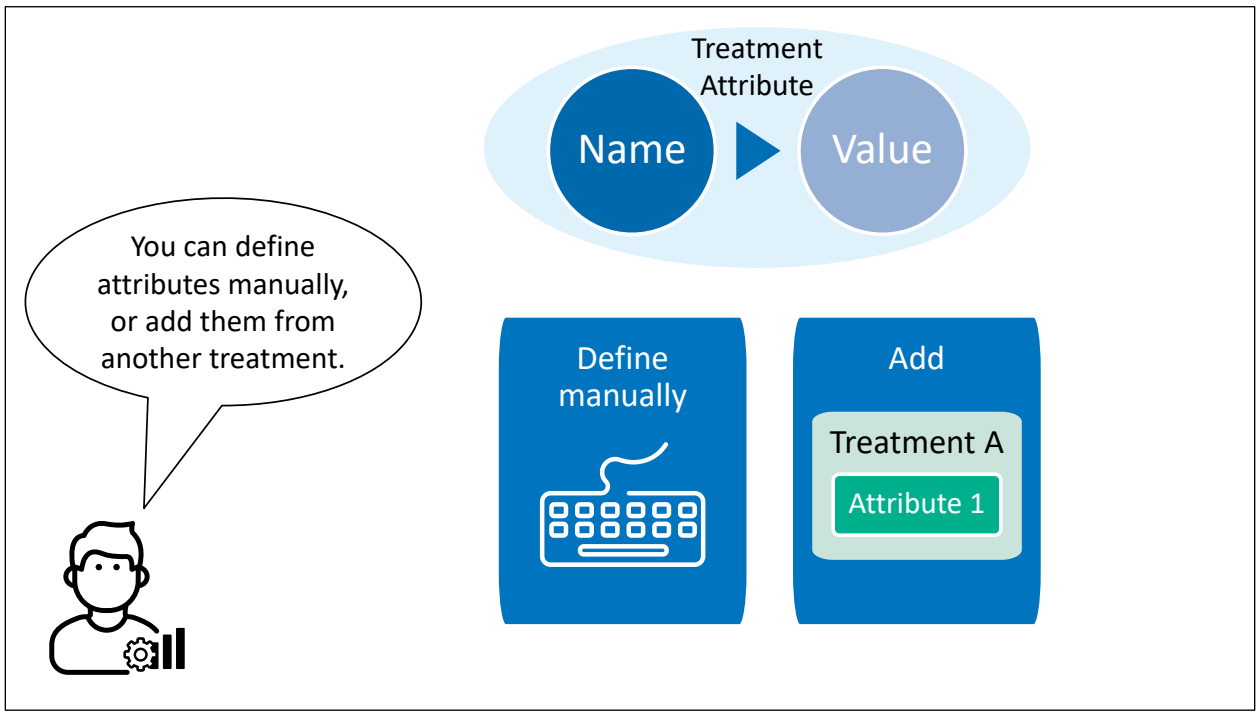


Let's begin with attributes.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

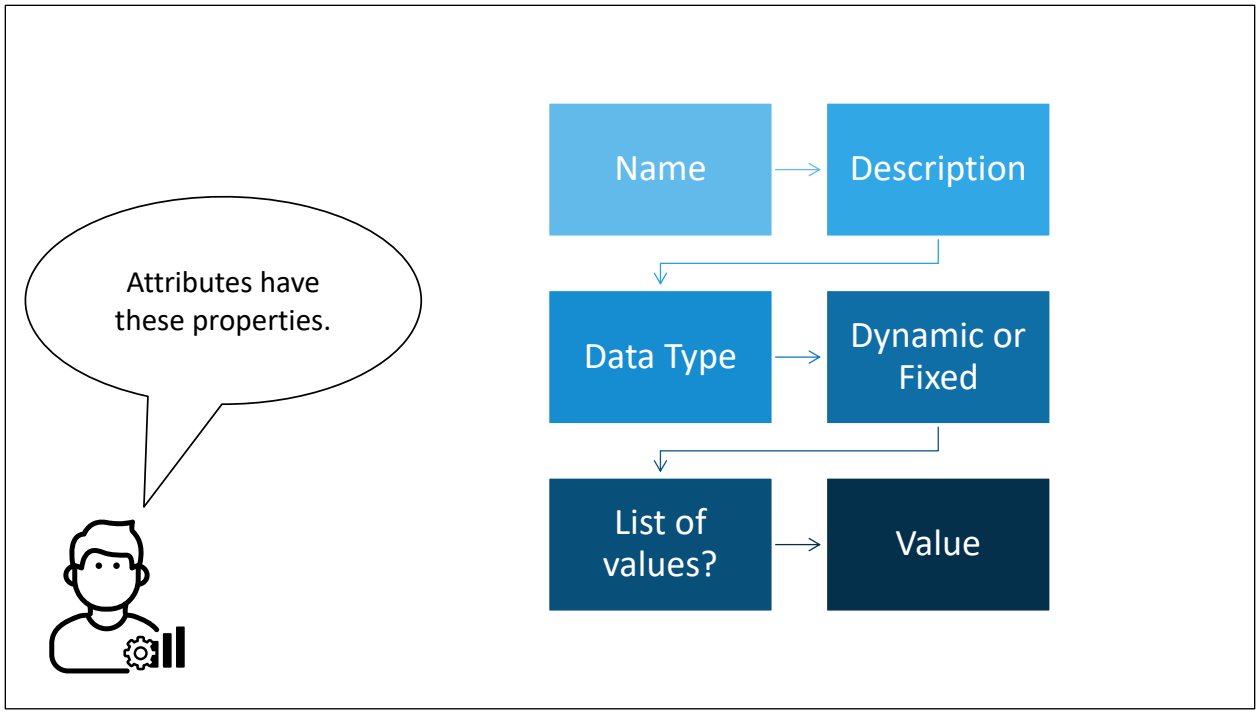


Attributes are name-value pairs. In the context of a marketing campaign, you could define attributes that specify the details of the offer that you present to a subject, such as a product code or a price.

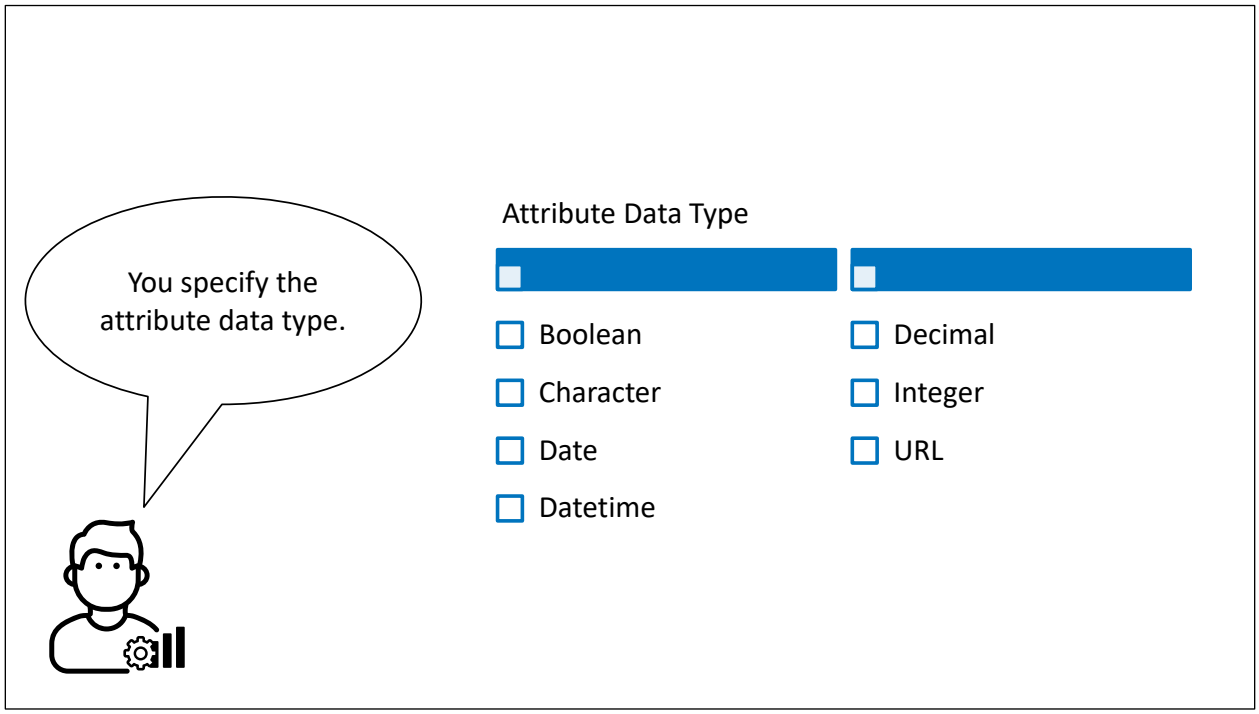


Attributes can be defined manually or added from another treatment.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

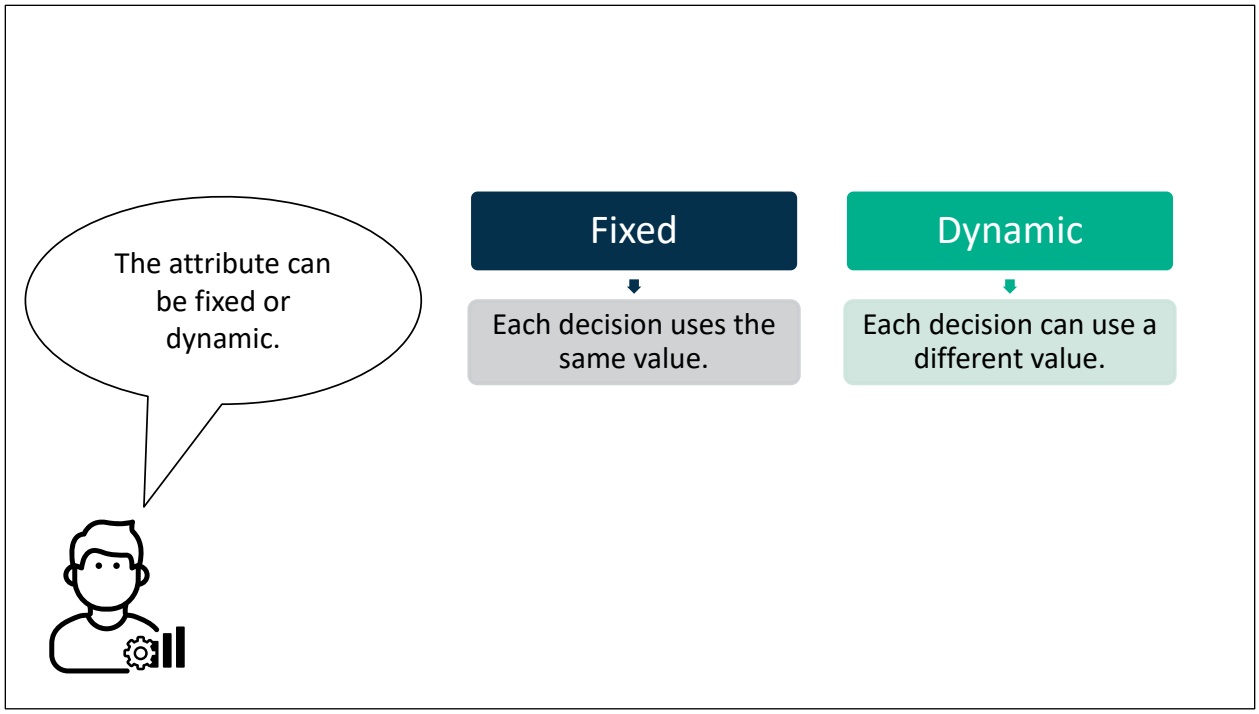


When you define an attribute manually, you specify these properties.

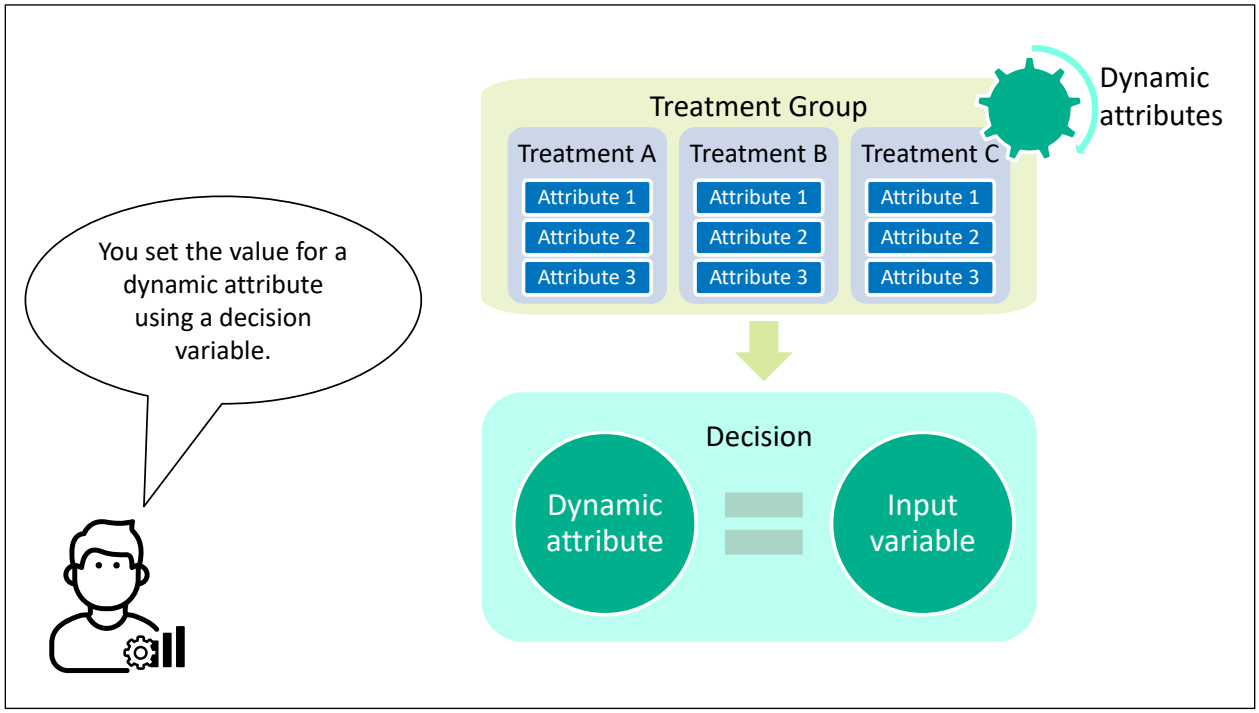


When you define a treatment attribute, you specify the data type. Your choices are similar to those for variables, but the choices do not include data grids or binary variables, and they do include a new type of URL.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

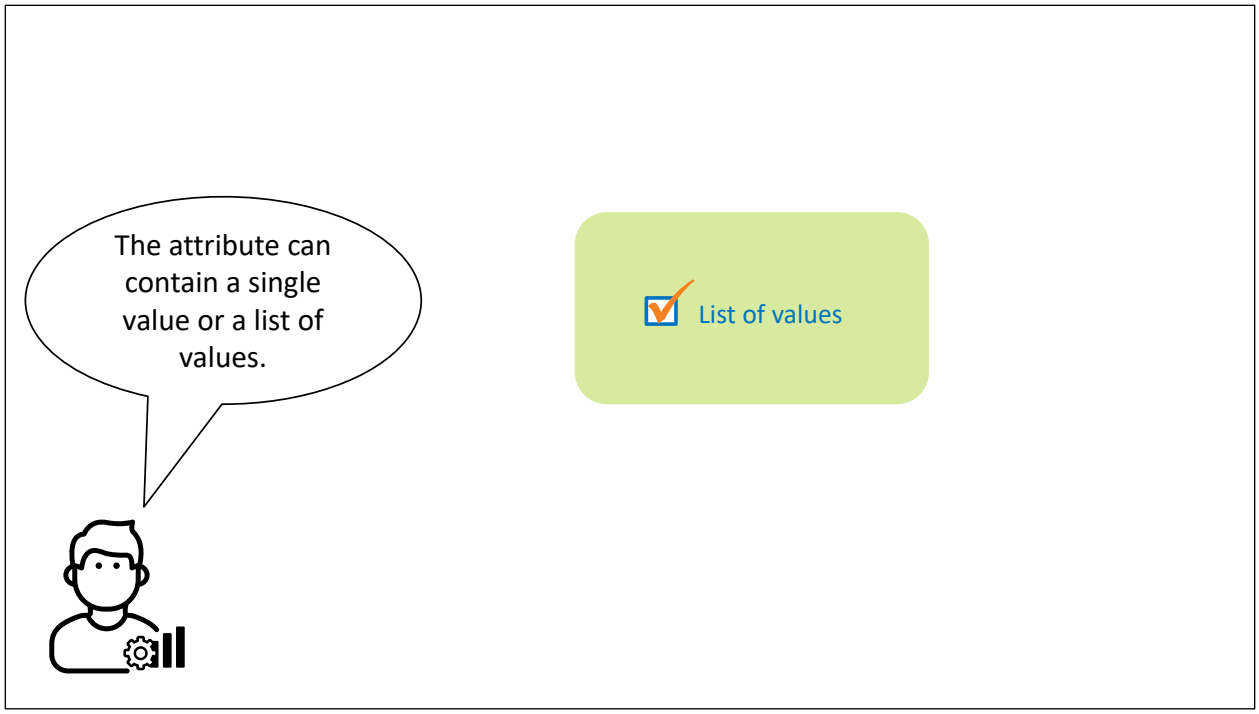


You can also specify whether the value type is fixed or dynamic. When you choose fixed, you set the value when you create the treatment, and each decision uses the same value. When you choose dynamic, each decision can use a different value.

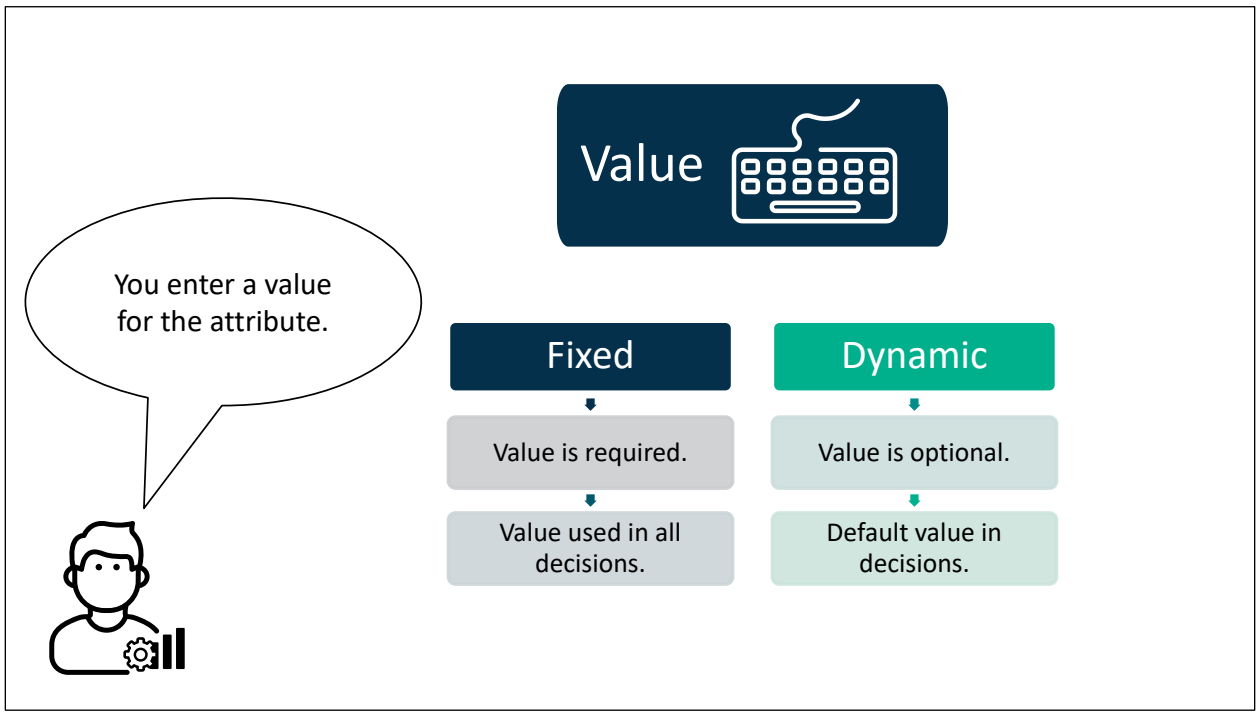


When you add a treatment group that contains dynamic treatment attributes to a decision, SAS Intelligent Decisioning creates an input variable that corresponds to each dynamic attribute. You set the value for the attribute by mapping a decision variable to the input variable.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

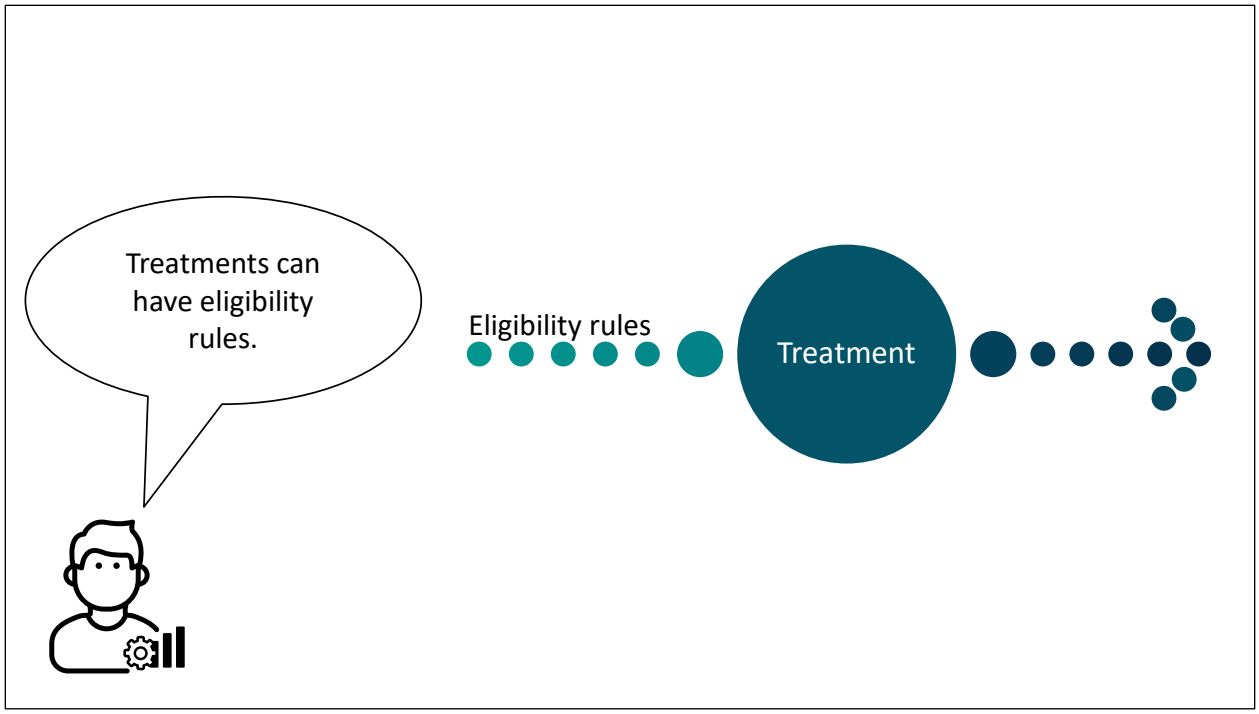


You can enable an option to specify that the attribute contains a list of values instead of a single value.

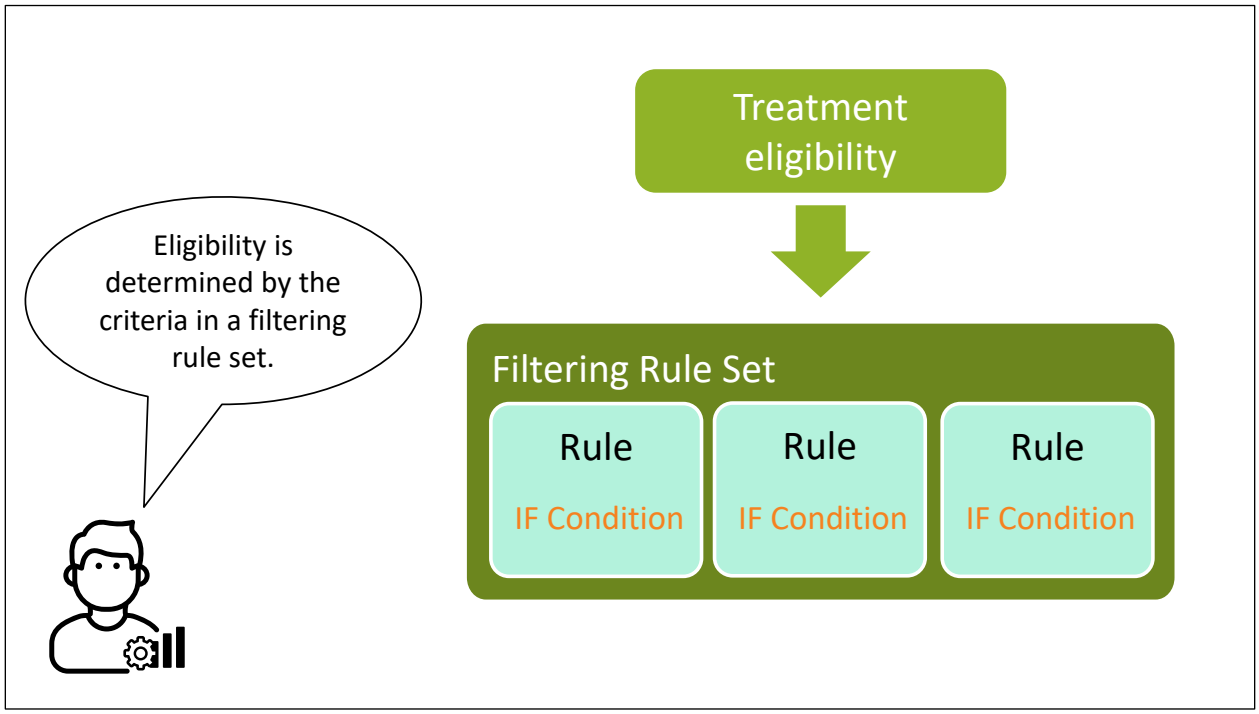


Finally, you enter a value for the attribute. If you set the attribute to be fixed, this is the value of the attribute that is used in all decisions that use the treatment. If you set it to be dynamic, it is the default value.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

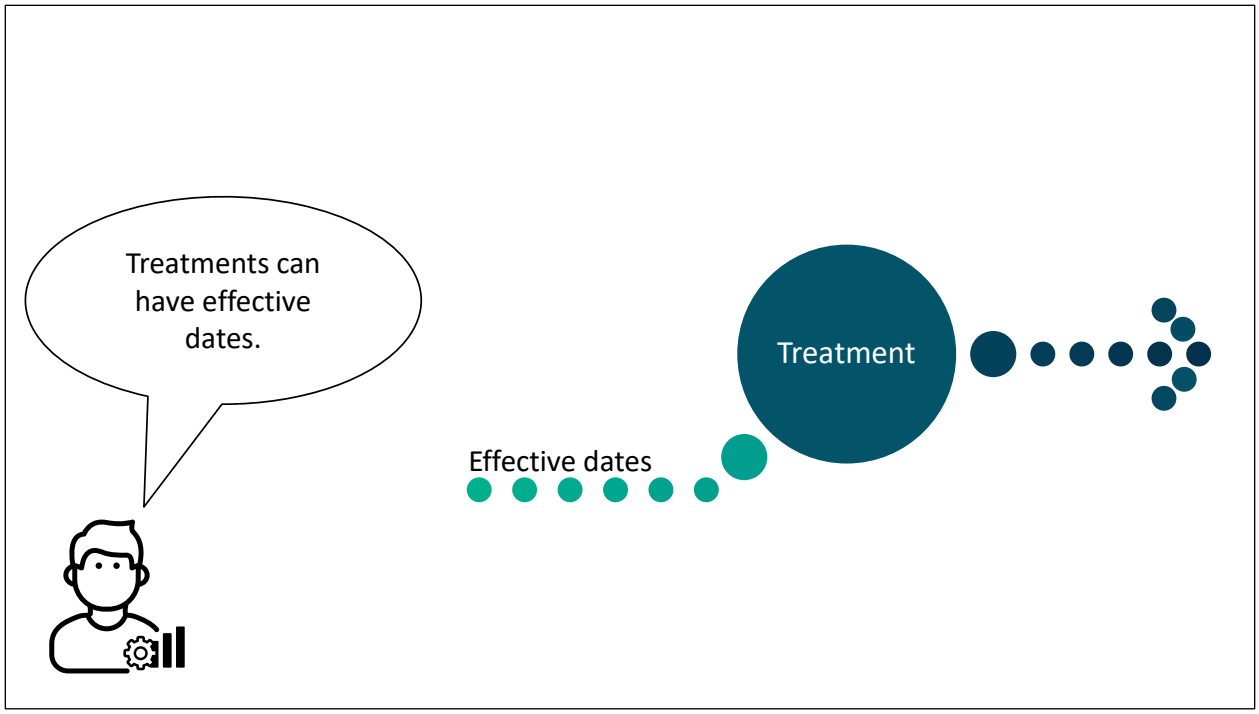


Treatments can have eligibility rules.

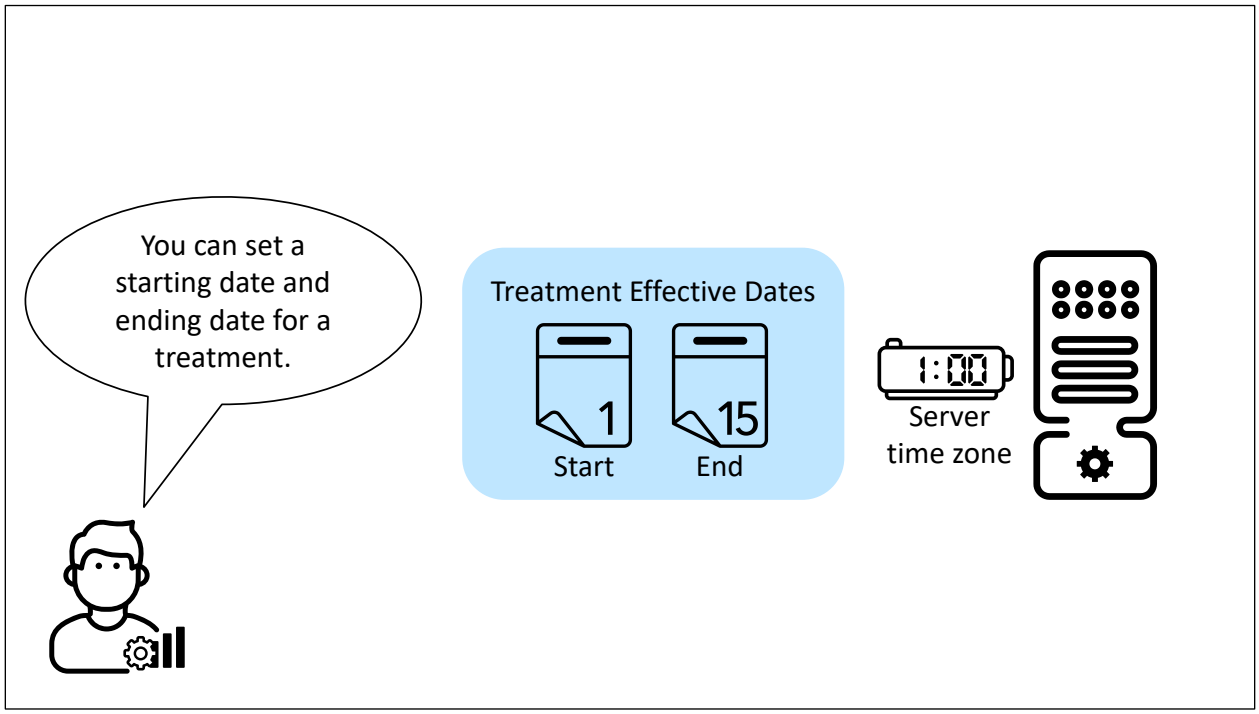


You can specify eligibility rules for the treatment by selecting a filtering rule set. The treatment is returned by a decision only when the IF conditions in the filtering rule set are true.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

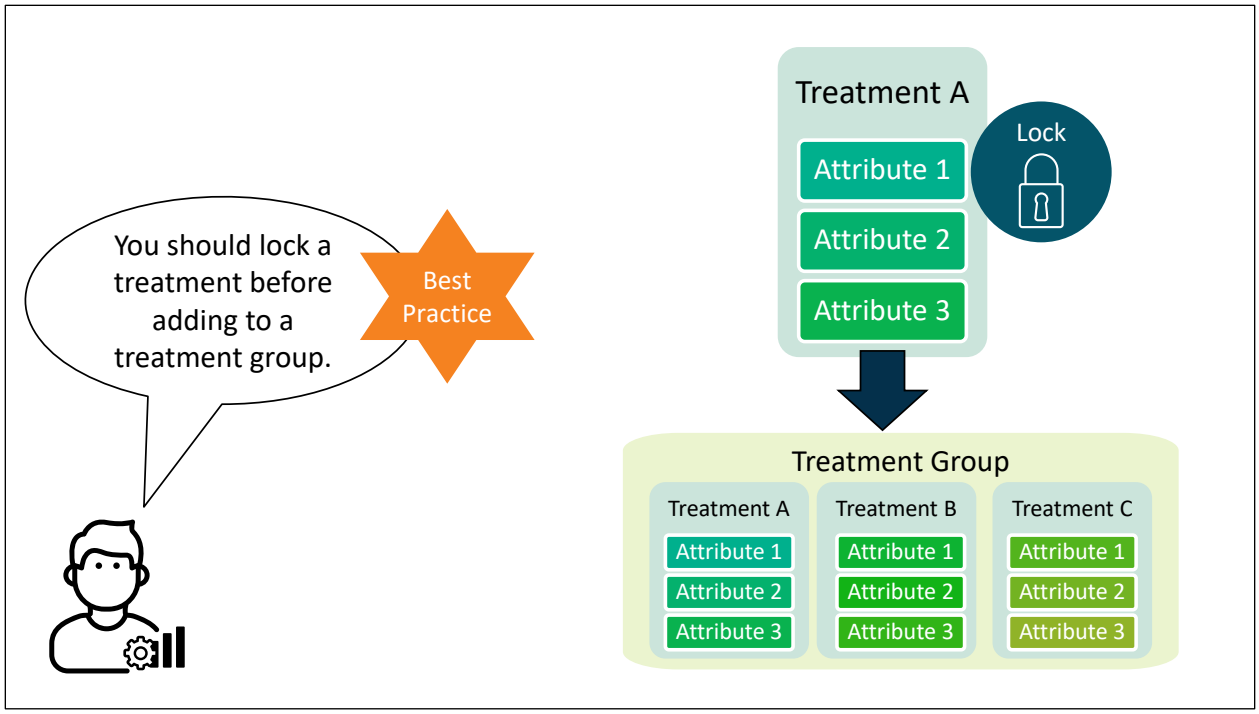


Treatments can have effective dates.

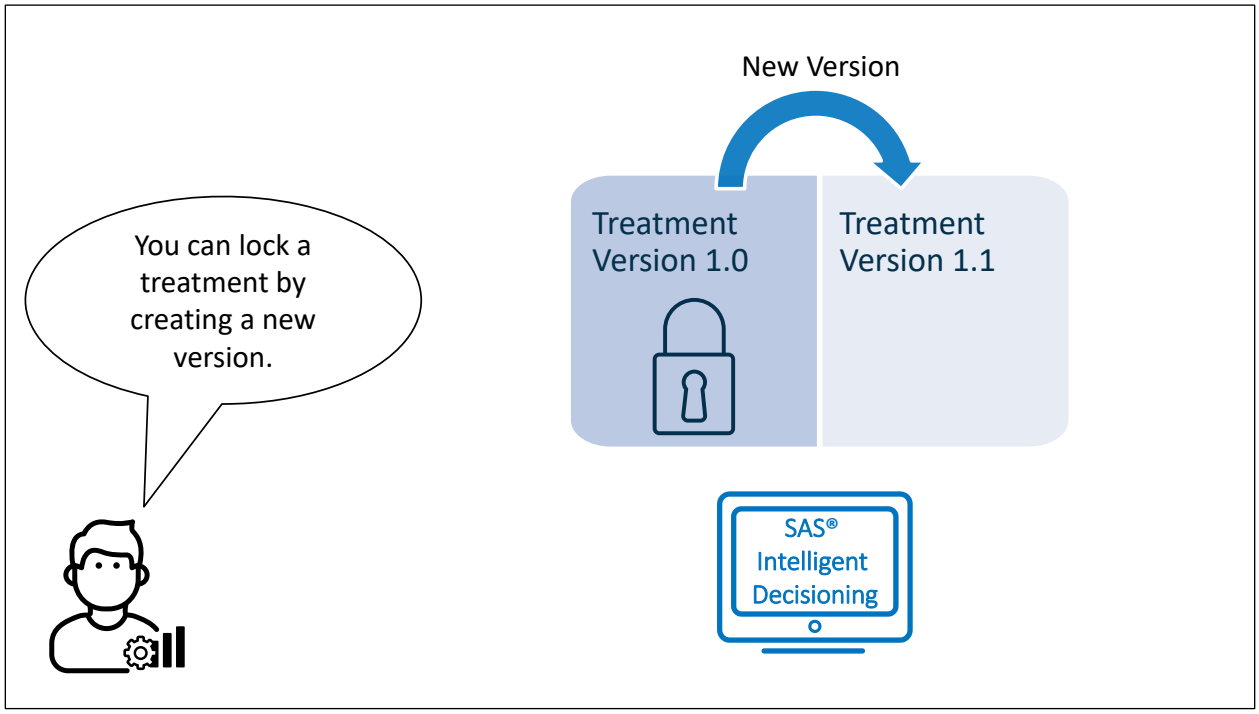


You can set a starting date and ending date for a treatment. The treatment will not be returned by a decision outside this date range. The dates are always based on the time zone of the server where the decision is executed.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

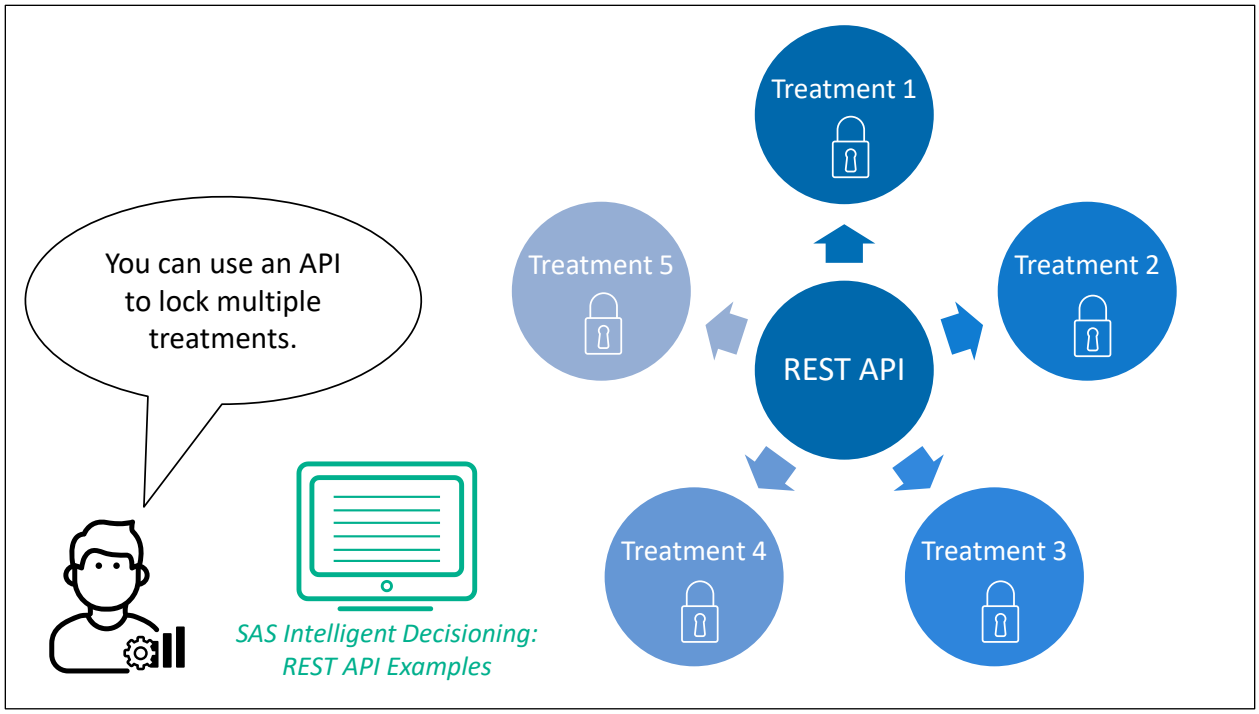


When you are finishing defining treatment properties, it's a best practice to lock the treatment before adding it to a treatment group. The locked version cannot be edited. This best practice enables you to determine the treatment properties that were used in the group for auditing and governance purposes.

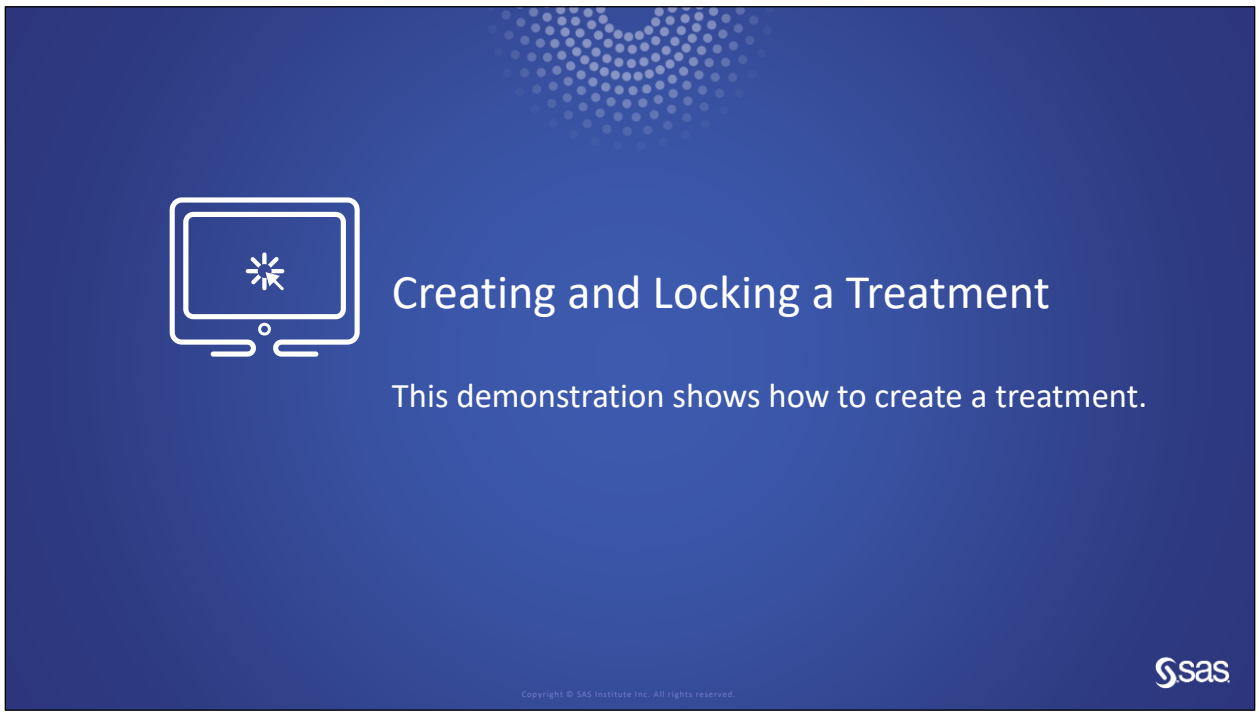


You can lock a treatment by creating a new version in the Intelligent Decisioning interface.

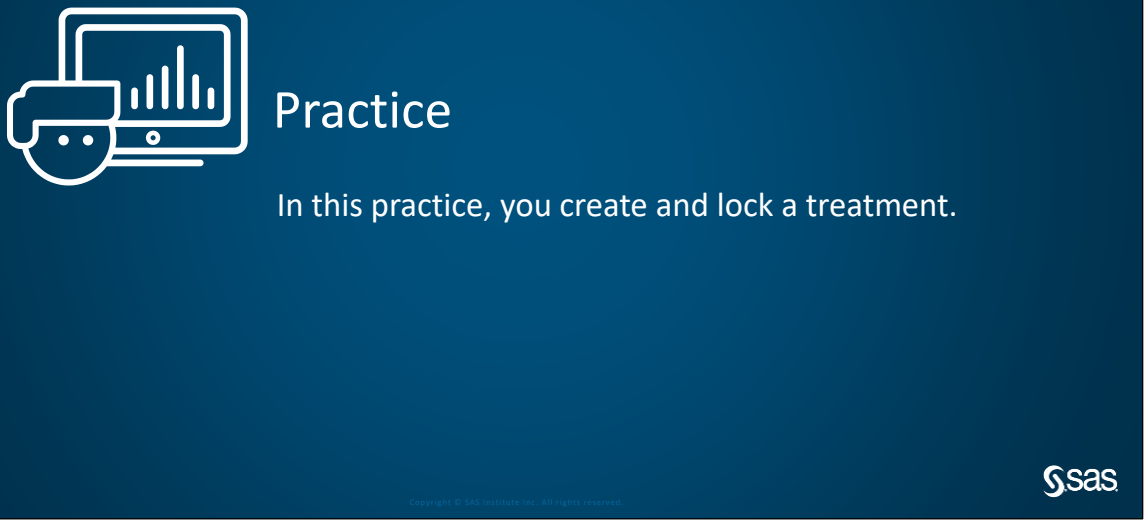
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

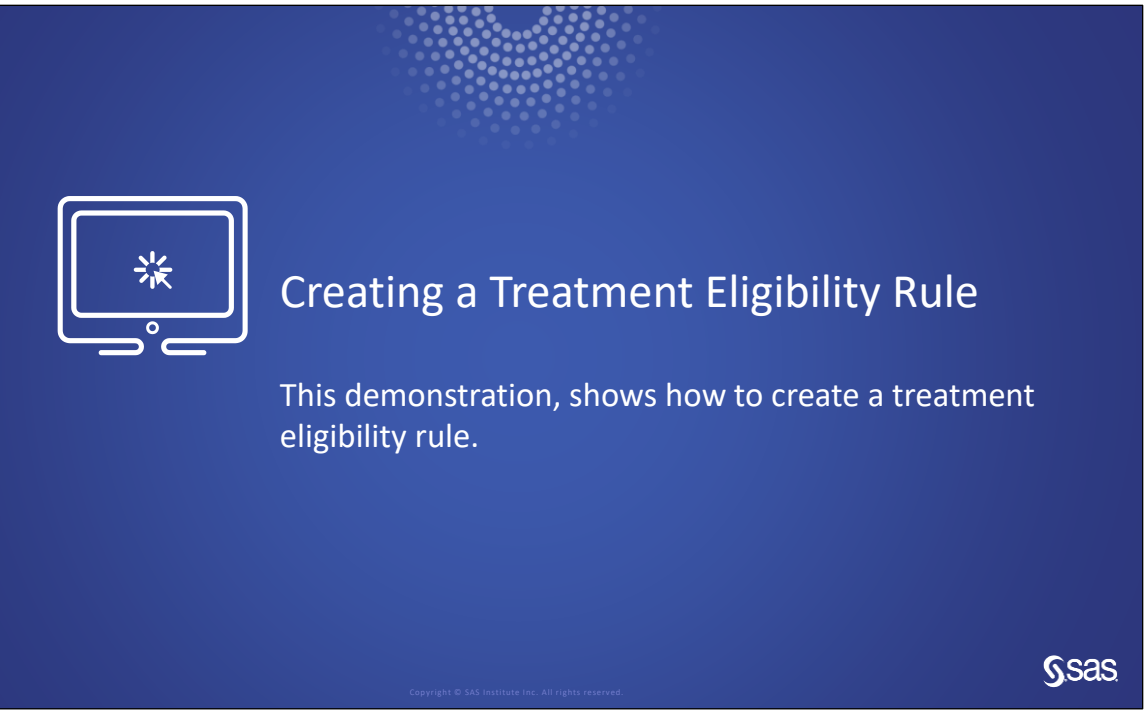


You can use a REST API to lock multiple treatments. Refer to *SAS Intelligent Decisioning: REST API Examples* for more information.

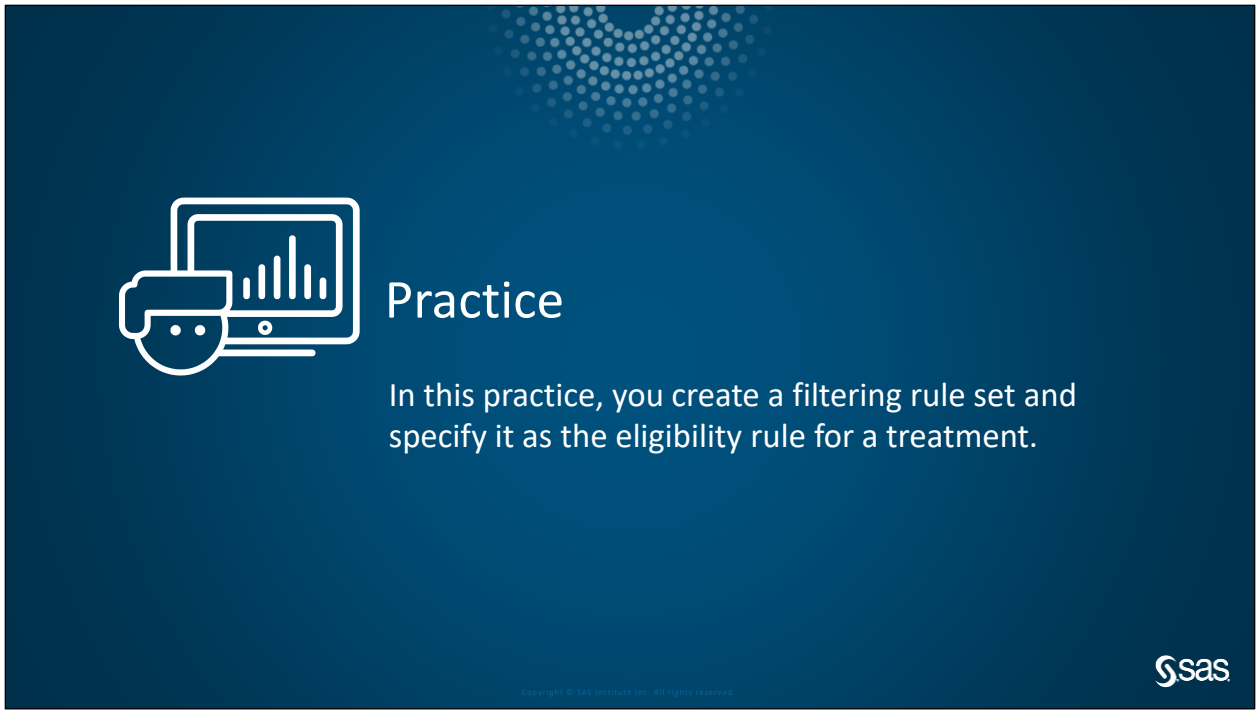


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

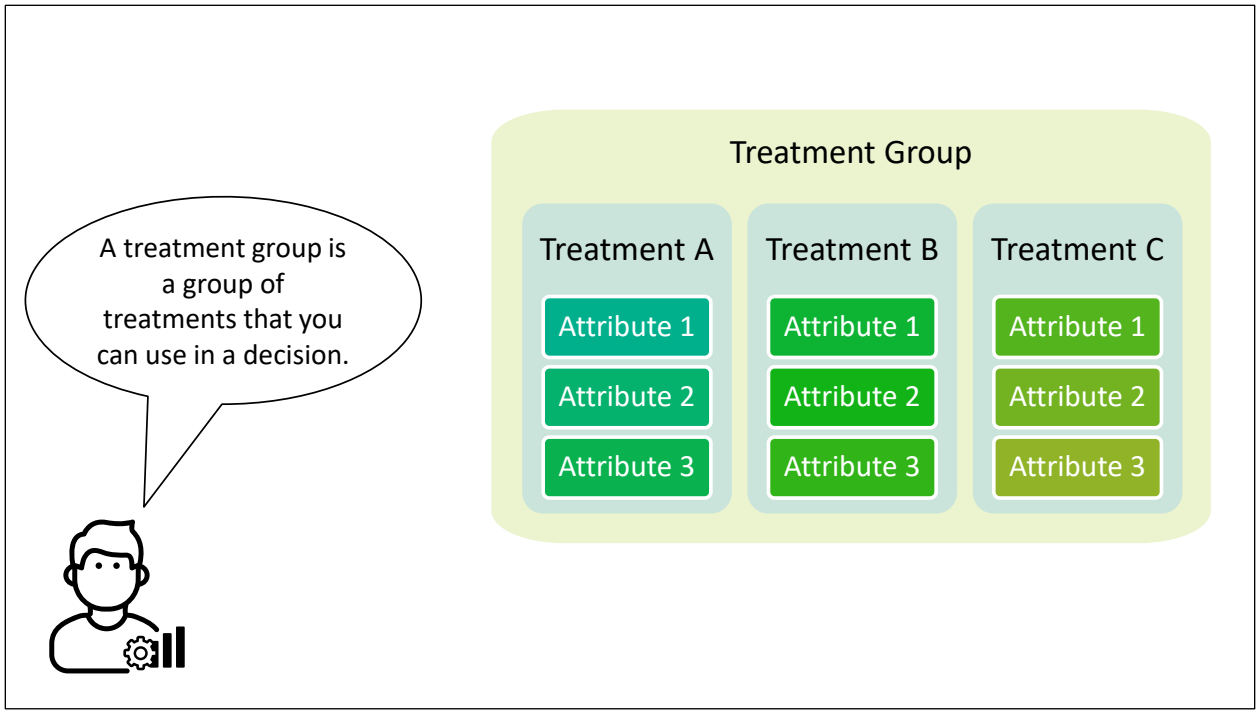




Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

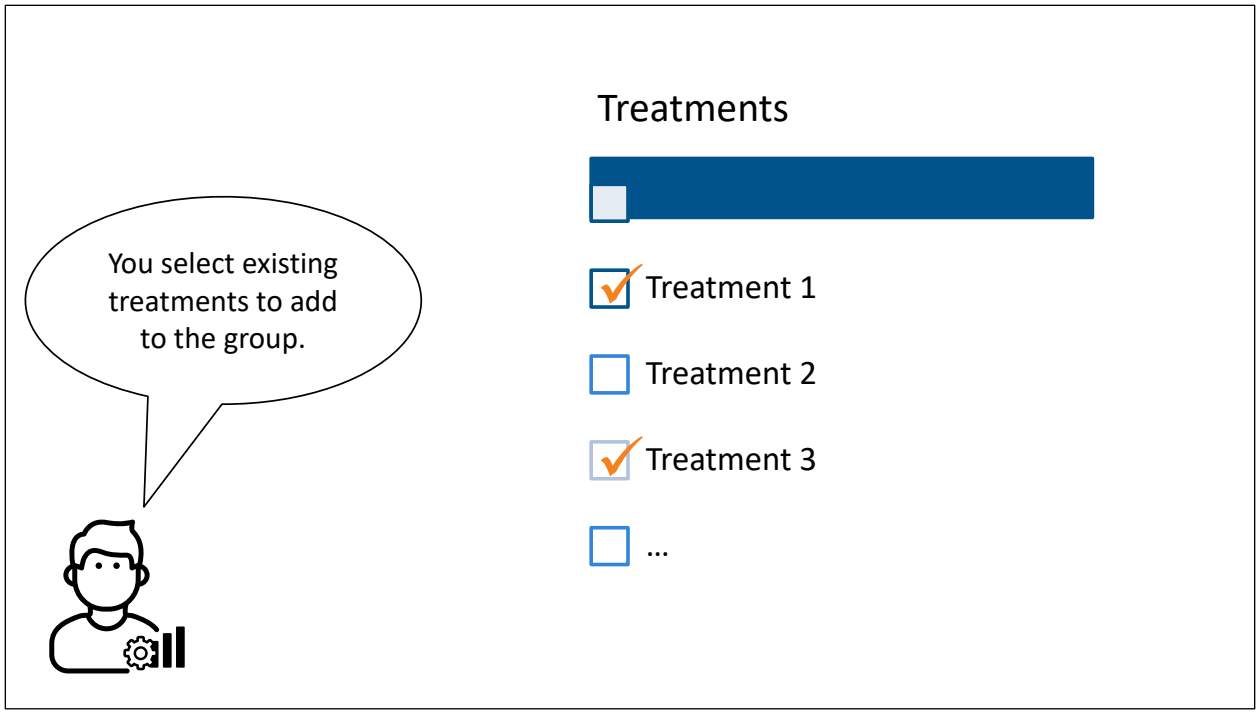


Treatment Groups

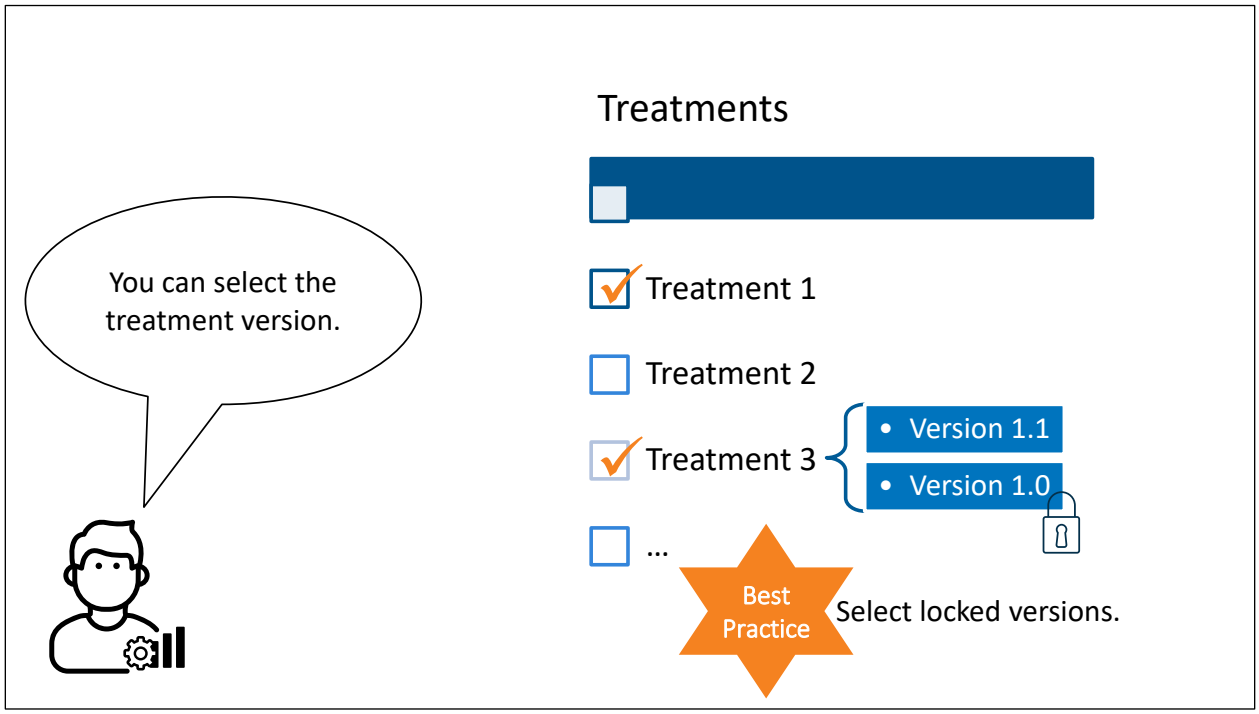


Recall that a treatment group is a group of treatments that you can use in a decision.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

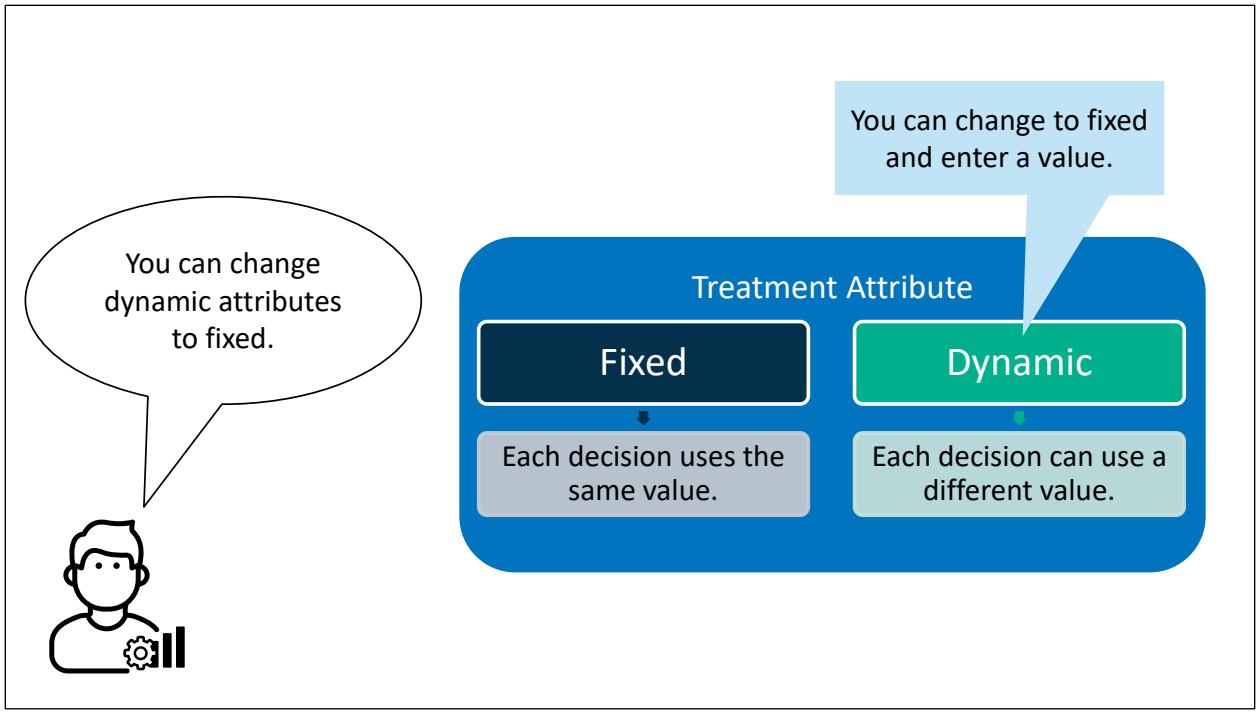


When you define a treatment group, you select existing treatments to add to the group.

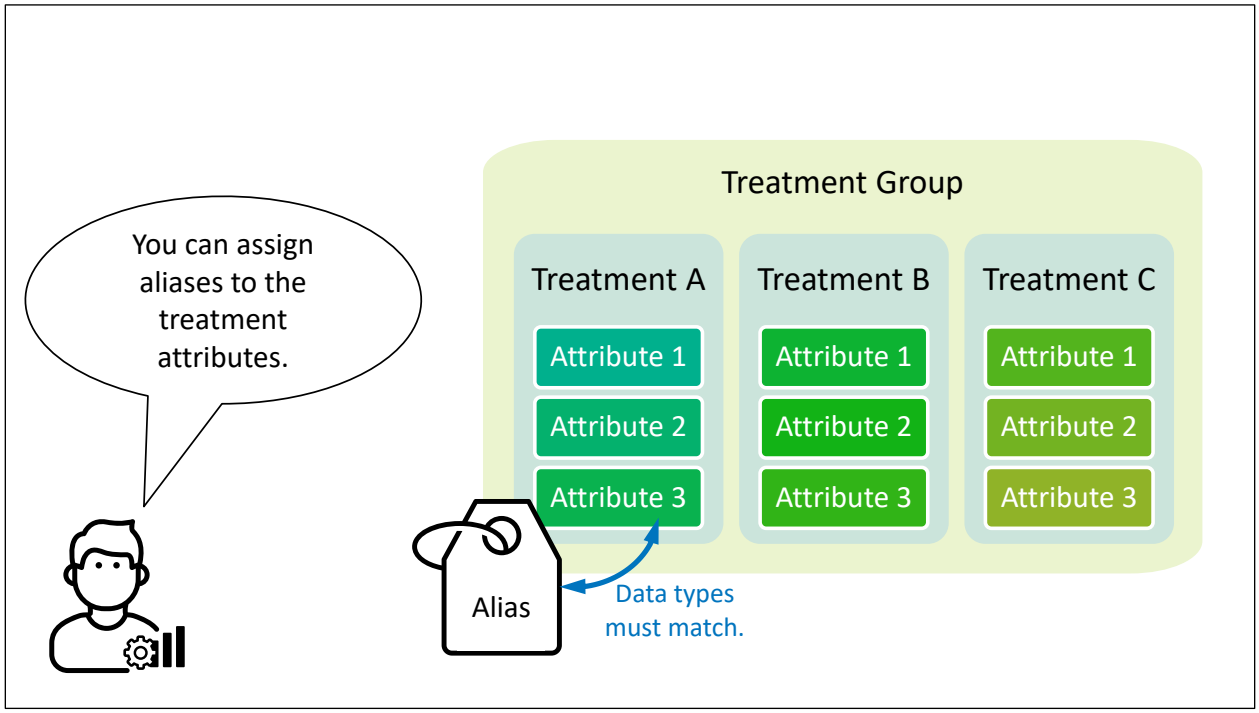


If a treatment has multiple versions, you can choose the specific version of the treatment that you want to use after adding it. Recall that it's a best practice to select only locked versions of treatments.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

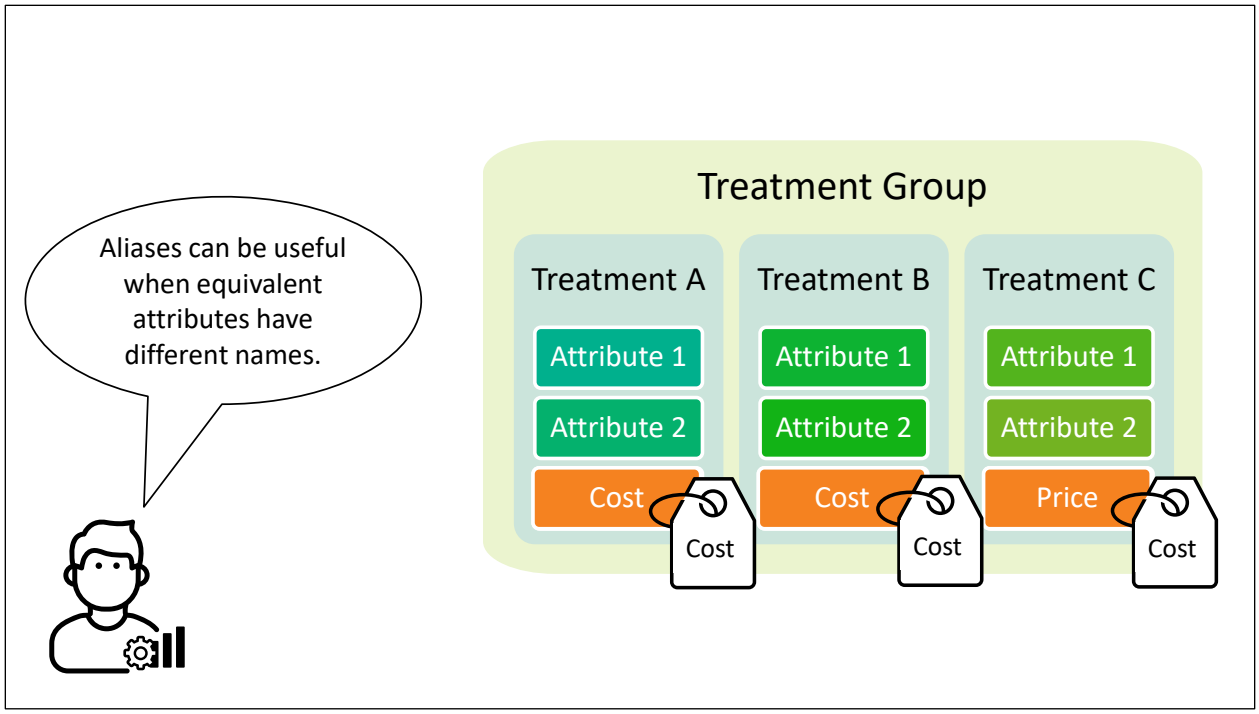


Recall that treatment attributes can be defined as fixed or dynamic. If any of the treatments include dynamic attributes, you can change them to fixed and enter a value. When you do so, the attributes will no longer get their value set in the decision.

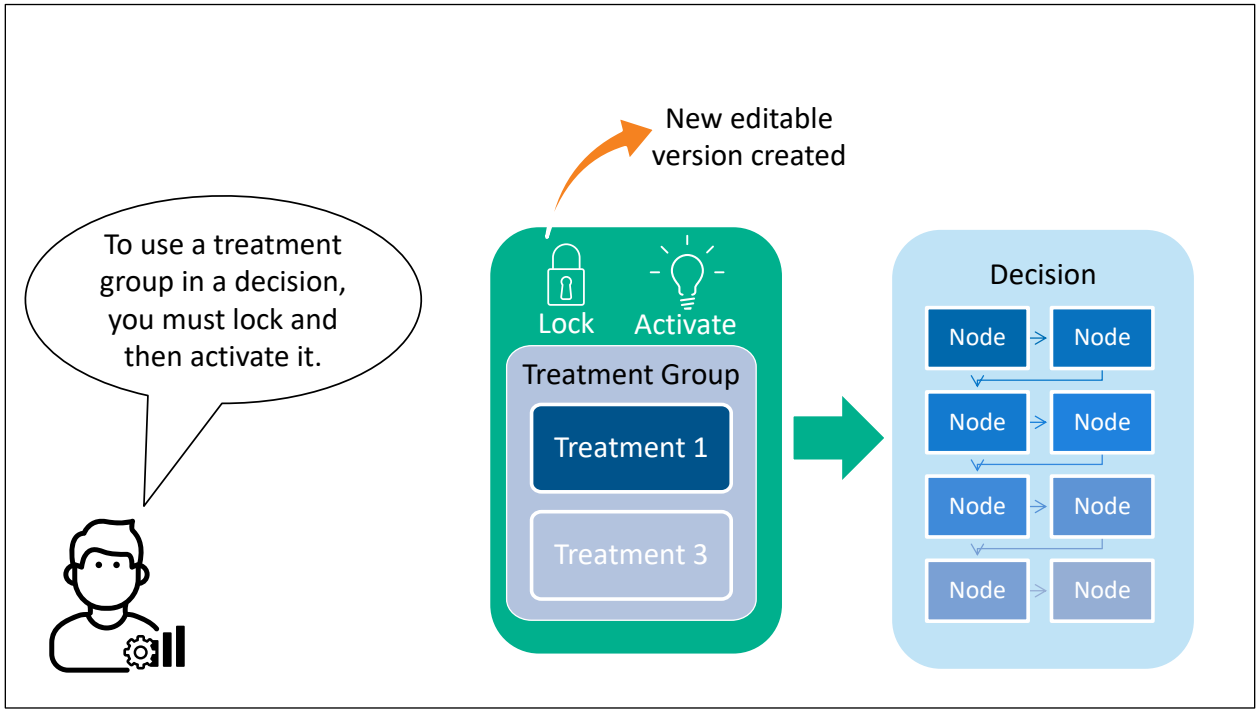


After you add treatments to the group, you can assign aliases to the treatment attributes. An alias is an alternative name for the attribute. The data type of the alias must match that of the attribute.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

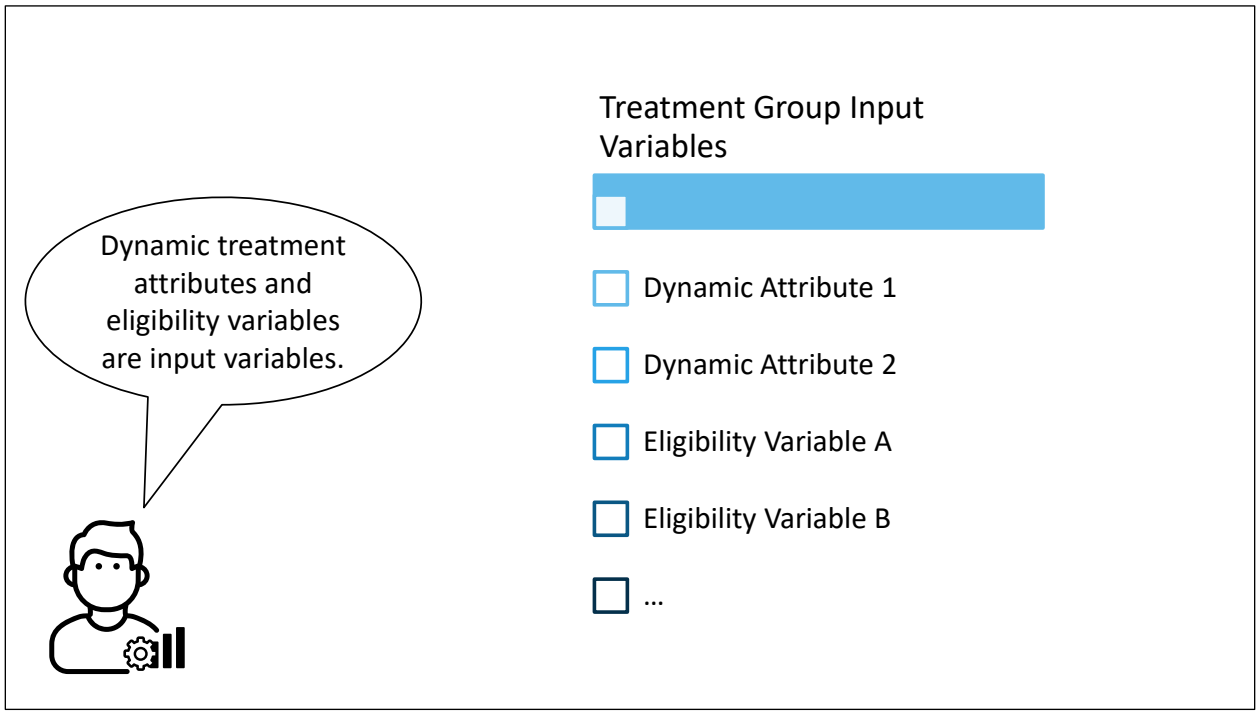


Aliases can be useful when attributes in different treatments within the group represent the same data but are defined with different names. For example, suppose these treatments have an attribute that represents the same monetary amount, but they have different names. You can assign an alias to be able to reference all of them with the same name. You can use aliases when performing treatment arbitration with differently named attributes, although you can also accomplish this task using variable mapping in the decision.

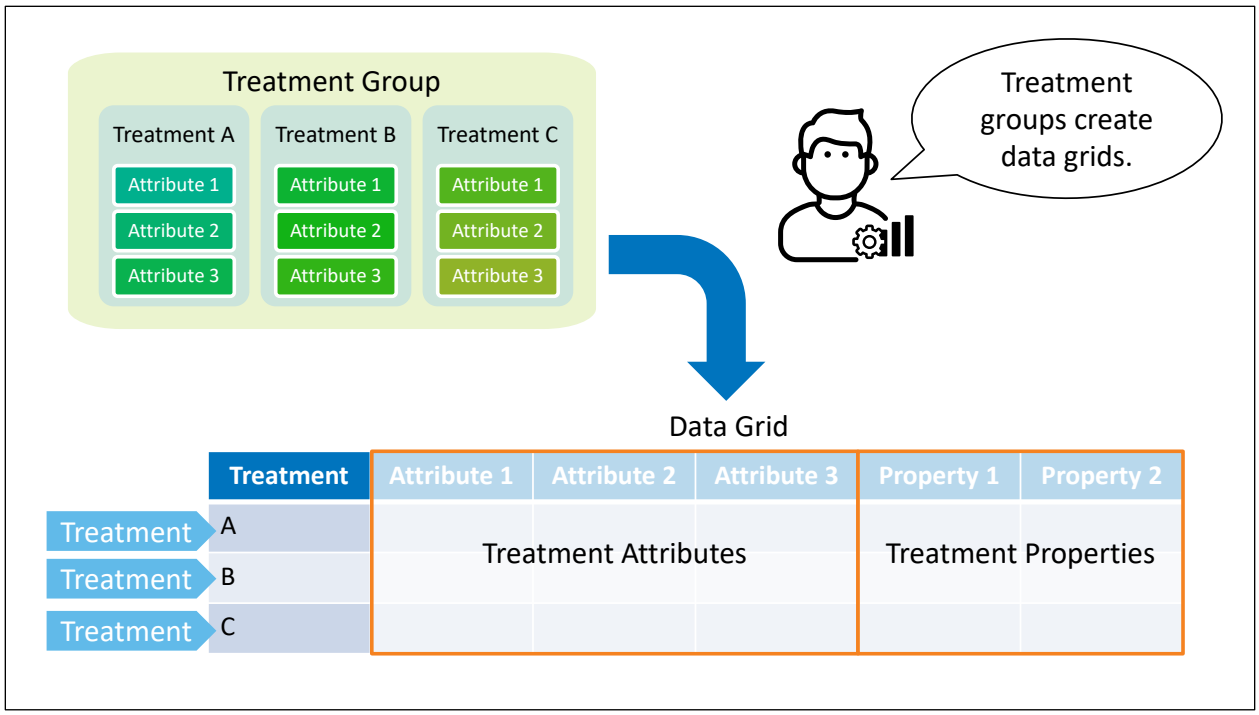


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Before you can use a treatment group in a decision, you must lock and then activate it. When you lock a treatment group, a new version is created. The activated version cannot be edited, but the new version can be.

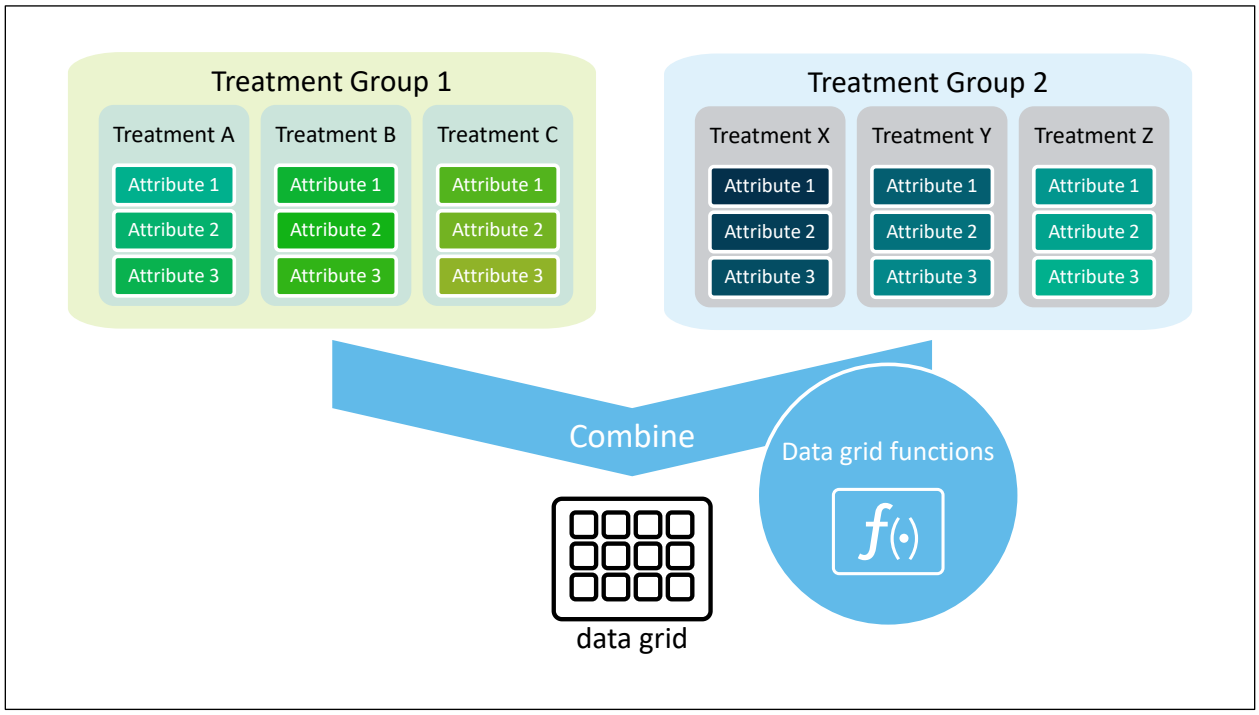


When you add a treatment group to a decision, any dynamic treatment attributes and eligibility variables in the group are included as input variables to the treatment group.

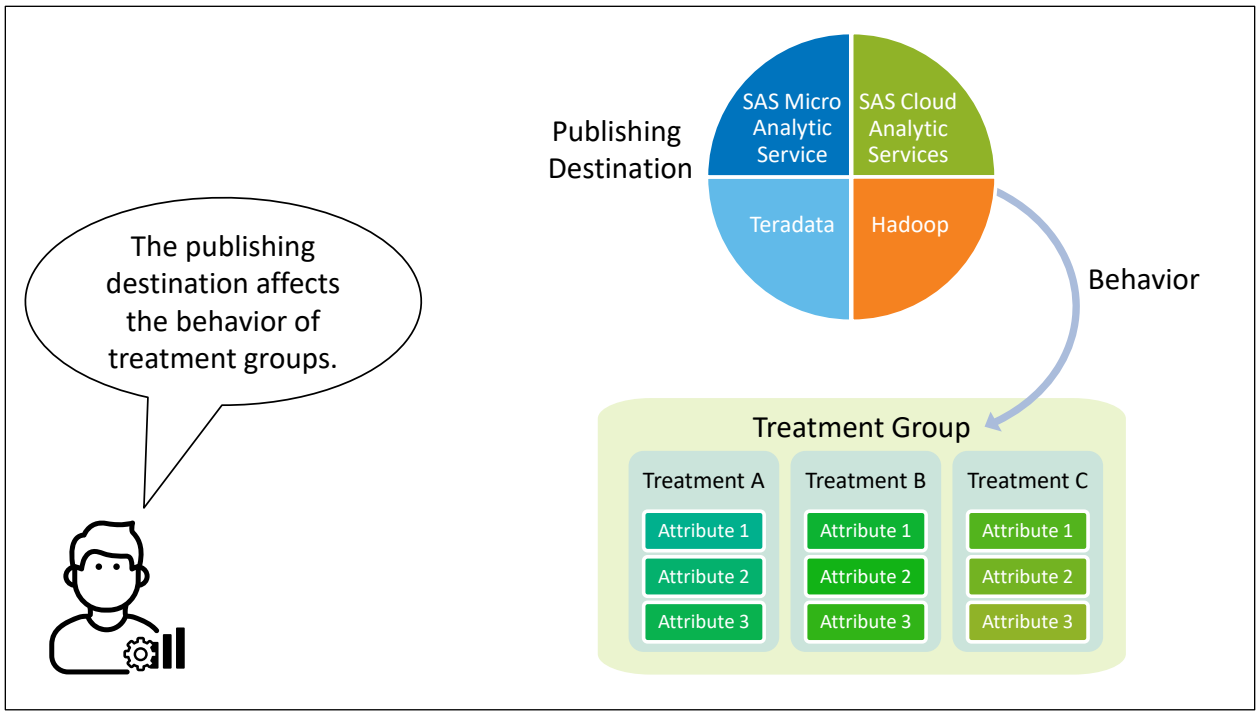


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

A treatment group in a decision creates an output data grid variable that contains one row for each treatment in the group for which eligibility rules are met. The columns of the data grid include the attributes that were defined for the treatment along with some standard treatment properties.

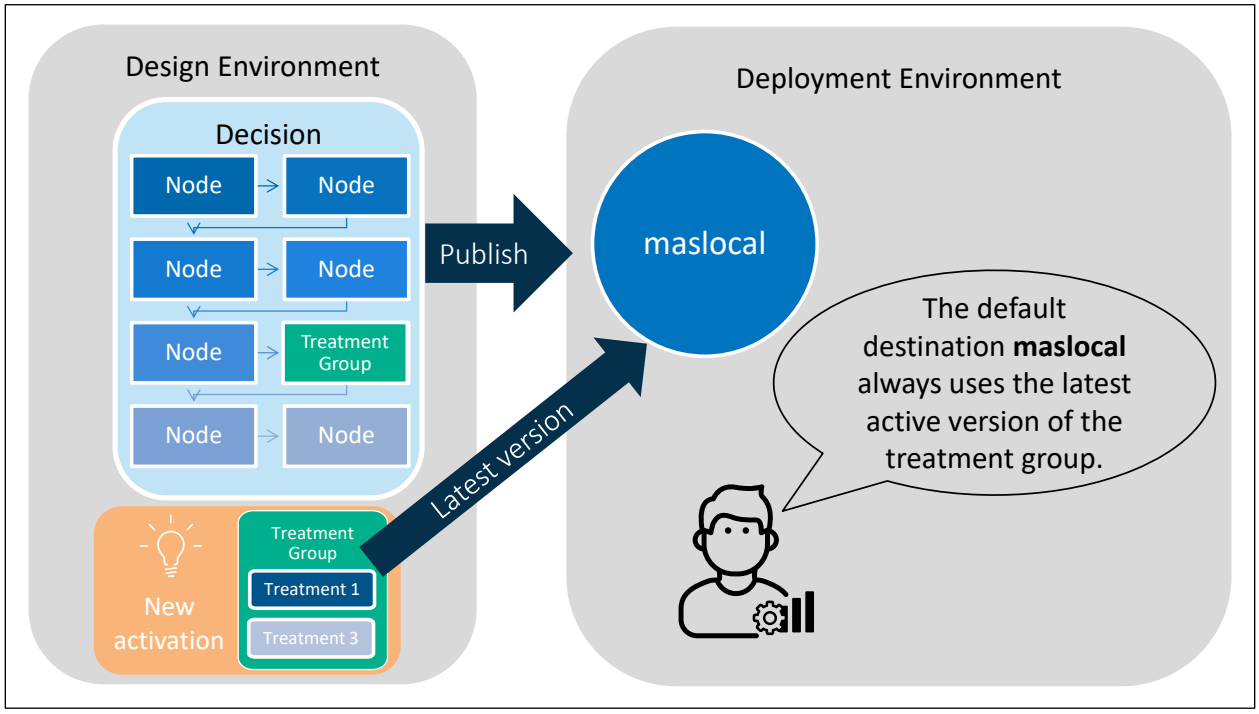


You can include multiple treatment groups in a decision. You can use data grid functions to combine the data grids into a single data grid if needed.

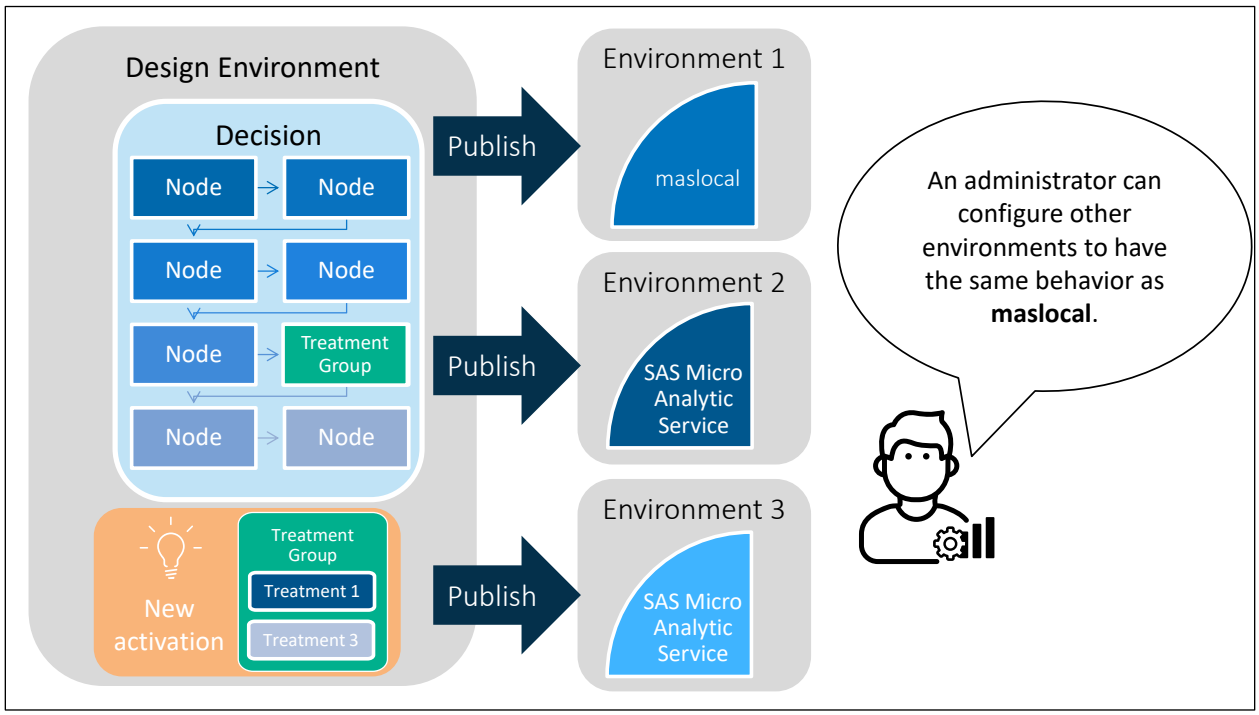


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The publishing destination for the decision drives some differences in the behavior of treatment groups after they are published.

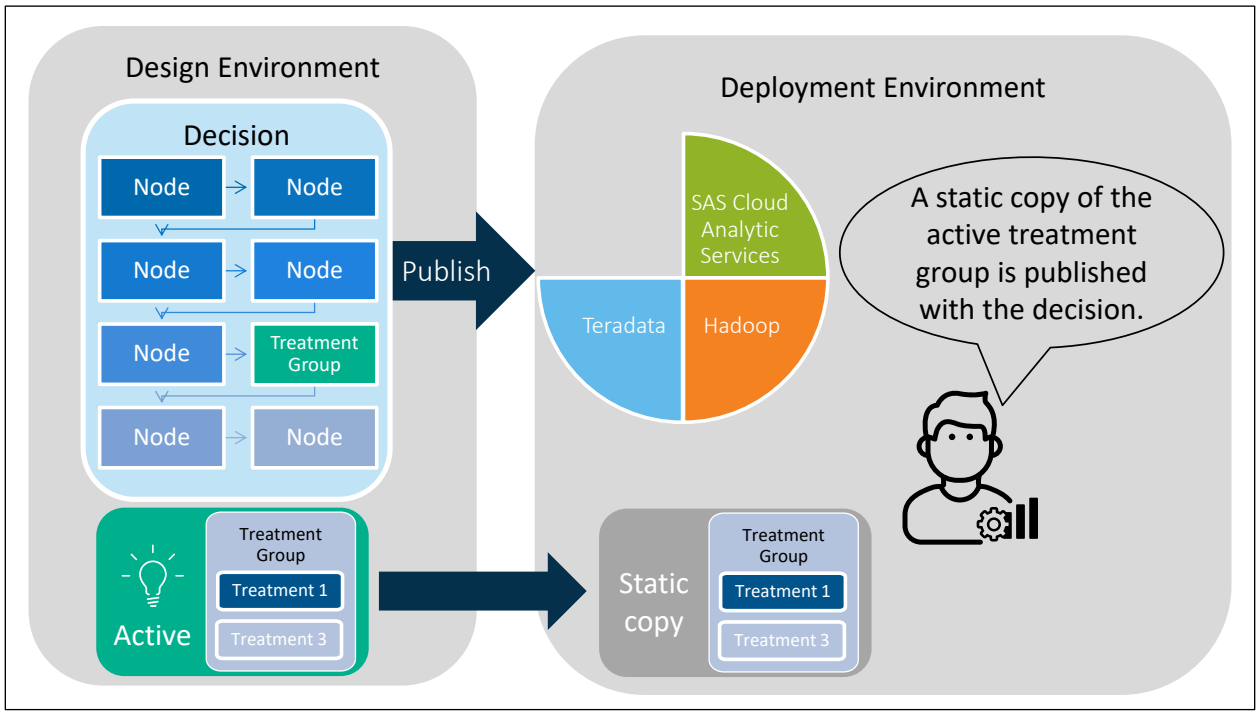


By default, a SAS Micro Analytic Service destination named **maslocal** is defined for you. After a decision that uses a treatment group is published to this destination, activations of new versions of that treatment group that occur later update the existing treatment module in the maslocal destination so that the decision picks up changes with newer activations.



Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

If you are publishing to multiple Micro Analytic Service destinations, an administrative user can configure them all to be included in treatment activation along with maslocal. This is done in SAS Environment Manager in the menu path **Configuration**, **Definitions**, **sas.treatmentdefinitions**, **activation.destinations**.



When you publish a decision that uses a treatment group to SAS Cloud Analytic Services, Hadoop, or Teradata, 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.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Creating a Treatment Group

This demonstration illustrates how to create and activate a treatment group.

Copyright © SAS Institute Inc. All rights reserved.

Practice

In this practice, you create a treatment group and activate it.

Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Adding a Treatment Group to a Decision

This demonstration illustrates how to add a treatment group to a decision and configure input variables.

Copyright © SAS Institute Inc. All rights reserved.

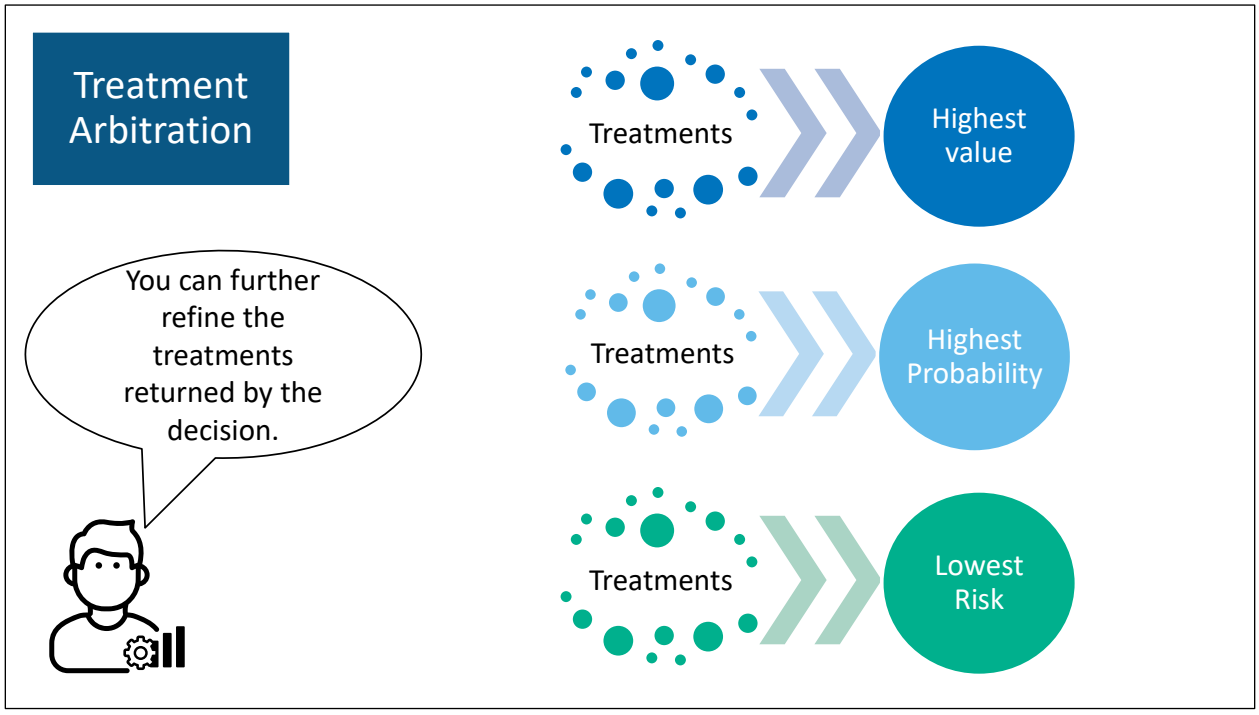
Practice

In this practice, you add a treatment group to a decision.

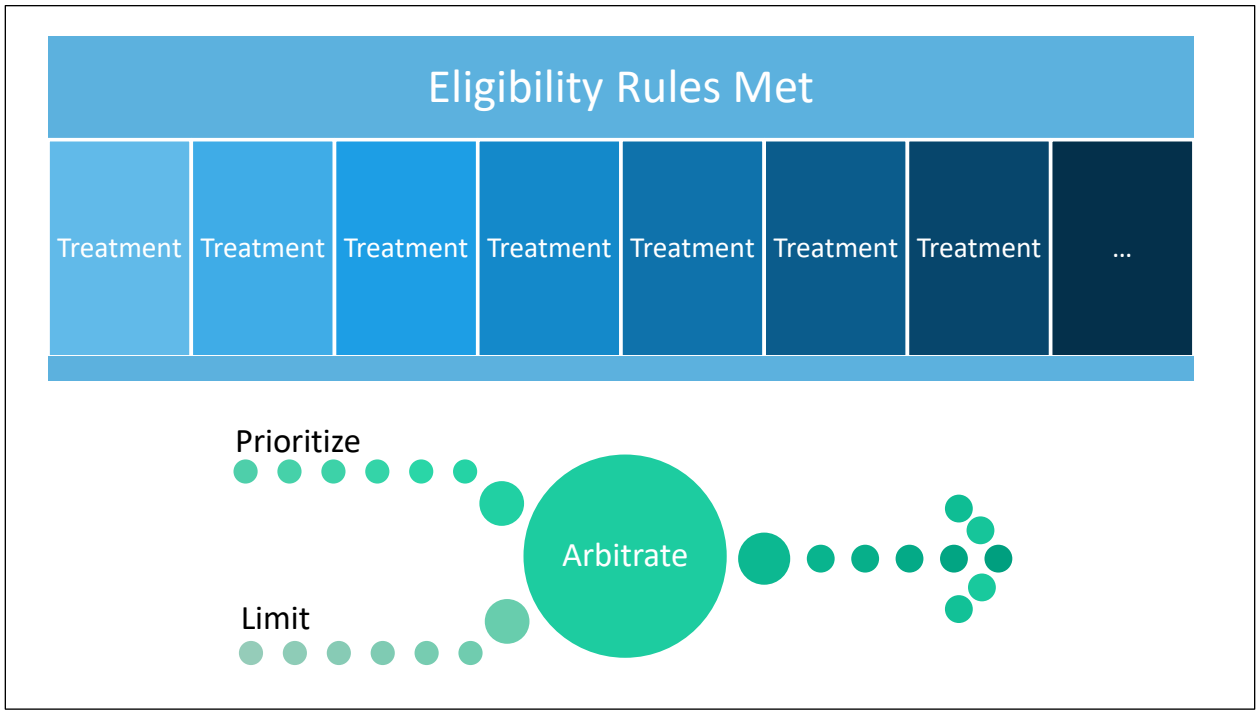
Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

4.2 Treatment Arbitration

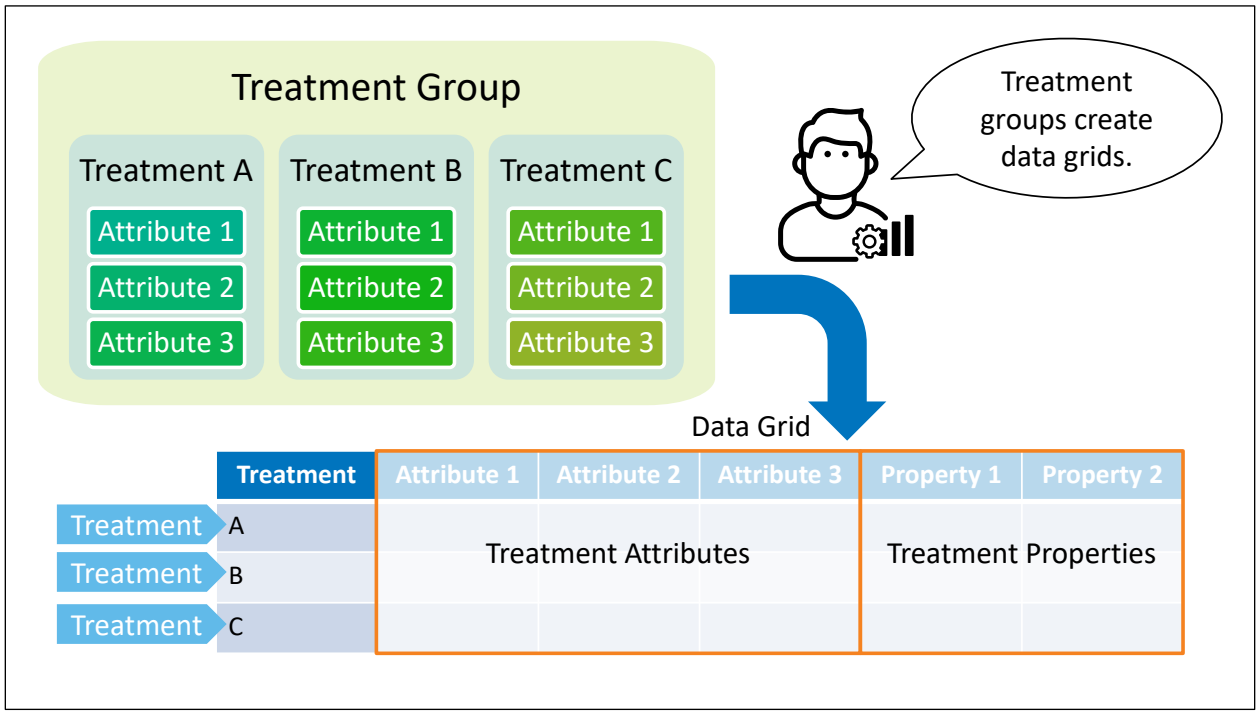


You can further refine the list of treatments returned by a decision. This process is known as treatment arbitration. For example, you could select a subset of treatments with the highest value or probability, or the lowest risk.



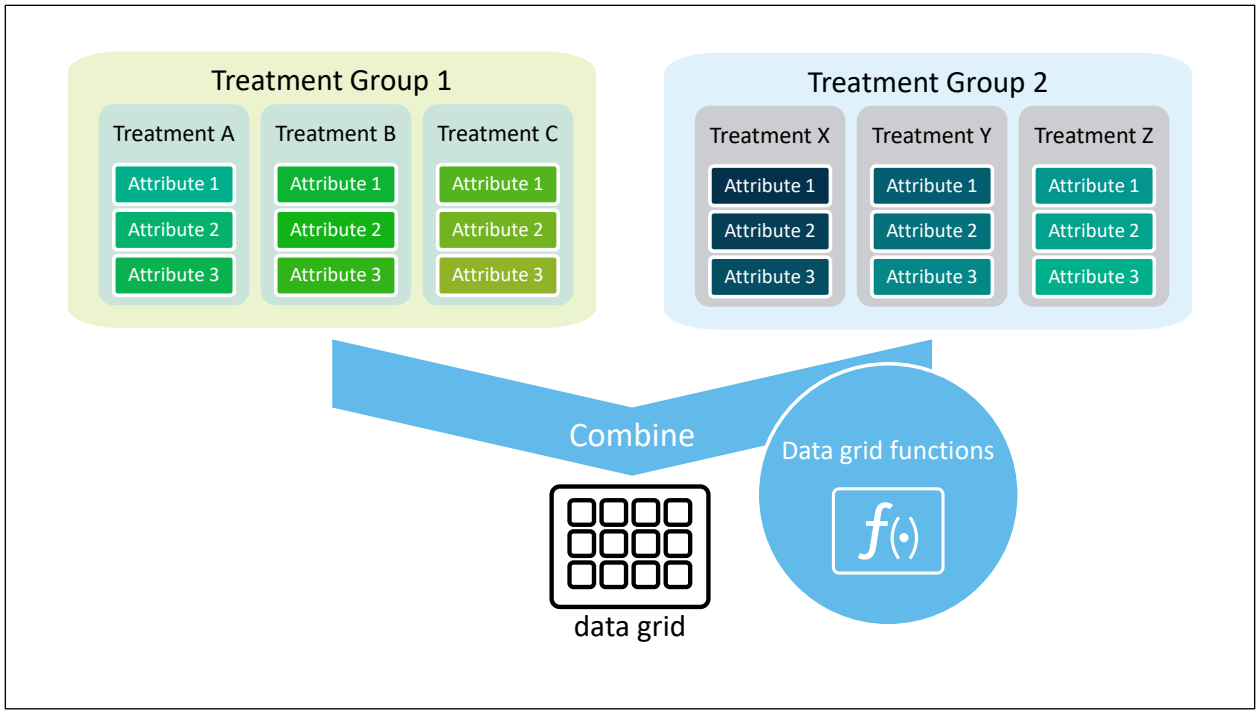
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

It's possible that eligibility rules for many treatments are met in a decision. You can define an arbitration process to prioritize the treatments to be returned by the decision. You can also limit the total number of treatments to be returned.

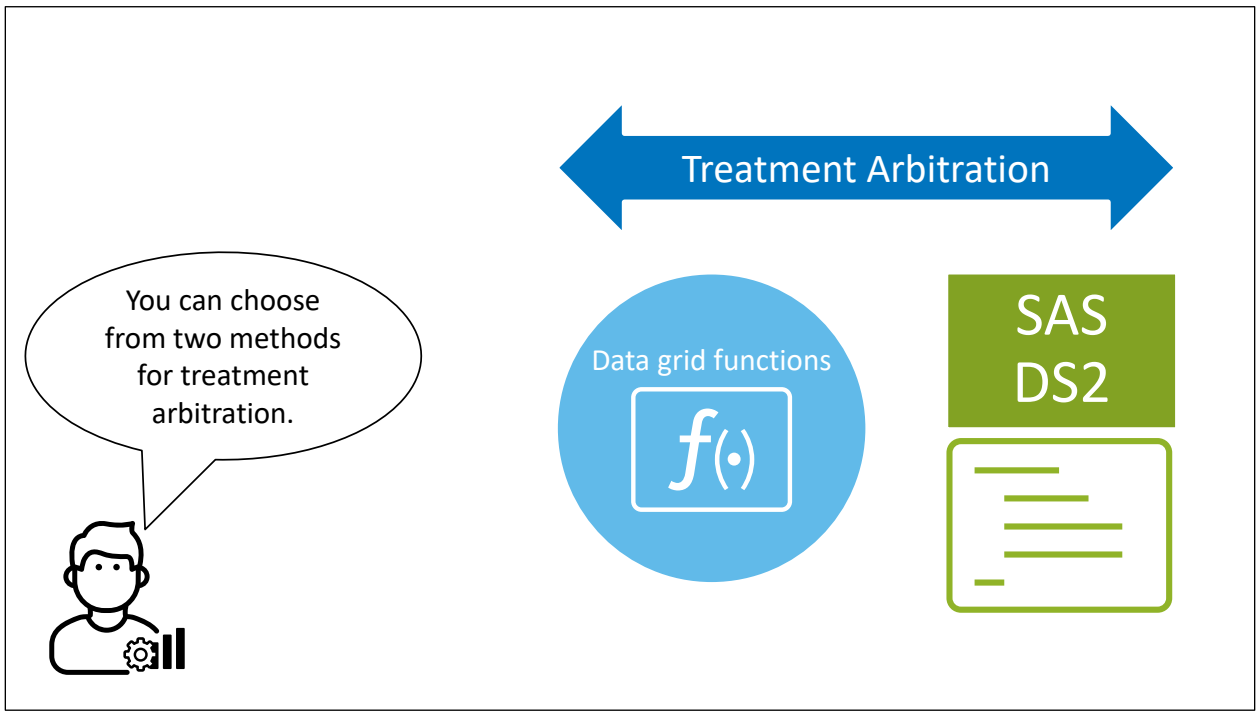


Recall that a treatment group in a decision creates an output data grid variable that contains a row for each treatment in the group for which eligibility rules are met. The columns of the data grid include the attributes that were defined for the treatments along with some standard treatment properties.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

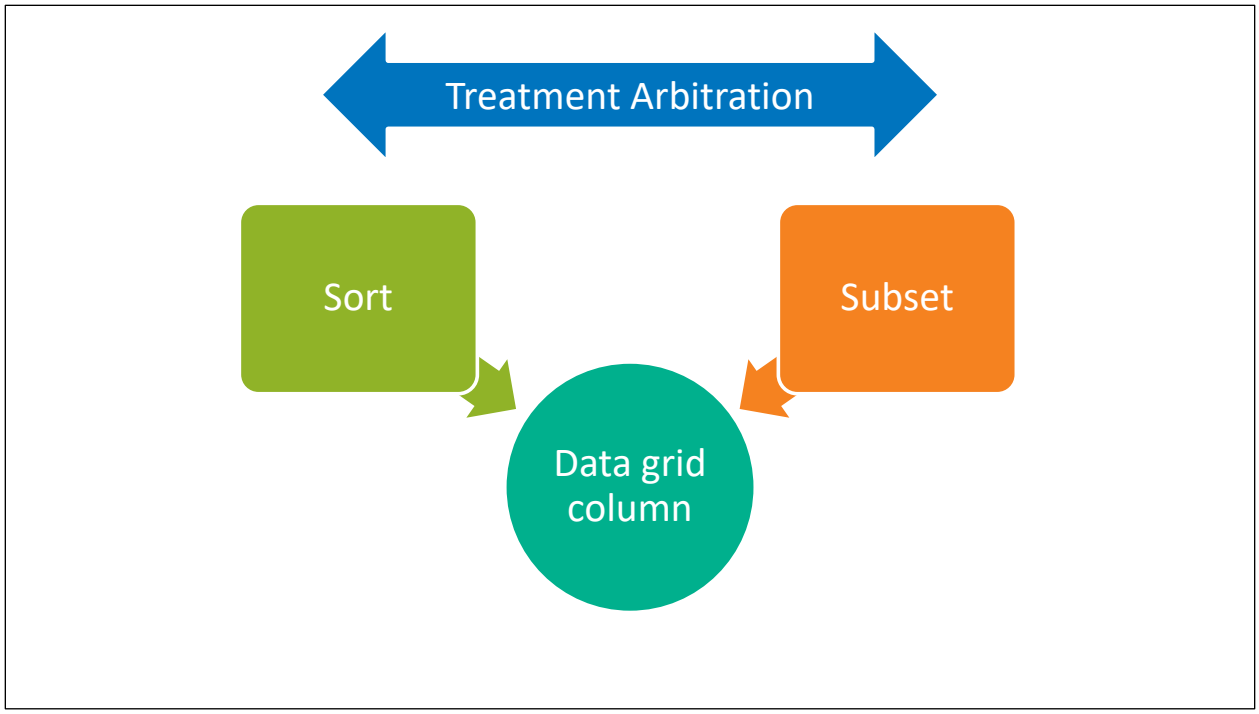


Also recall that you can include multiple treatment groups in a decision, and you can use data grid functions to combine the data grids into a single data grid if needed.

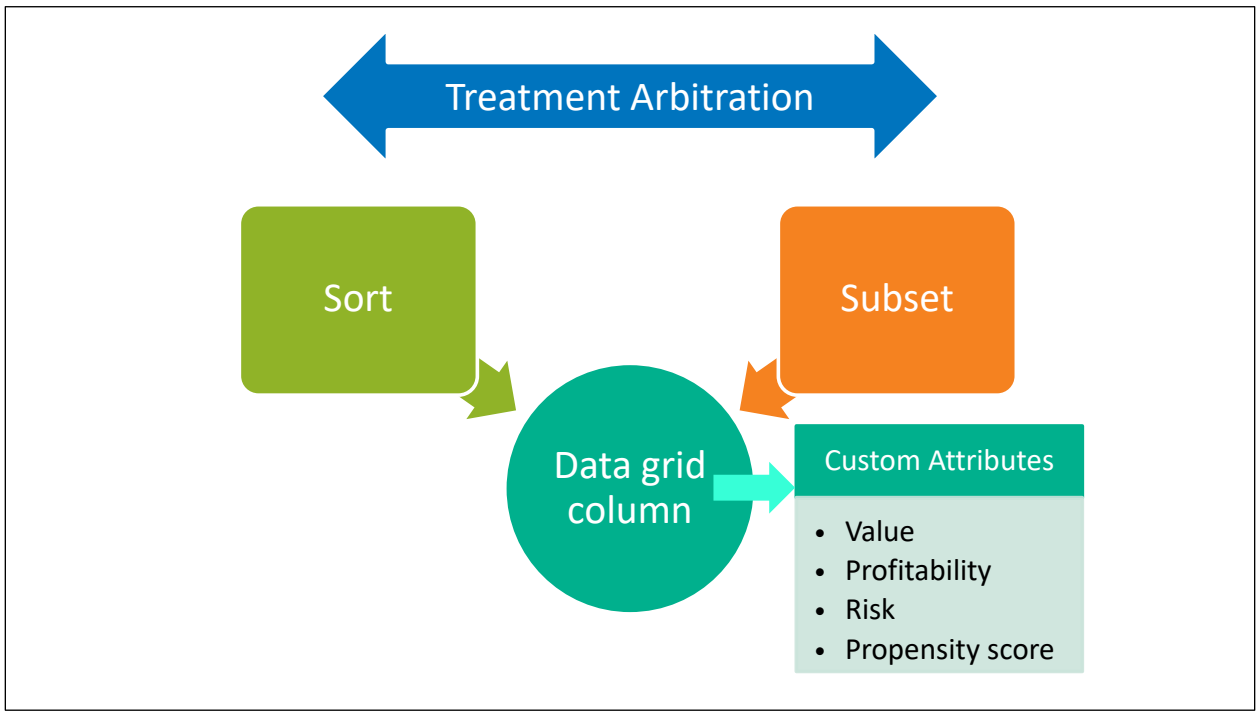


You can choose from two methods for treatment arbitration. You can use data grid functions in a rule set or you can write custom DS2 code.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

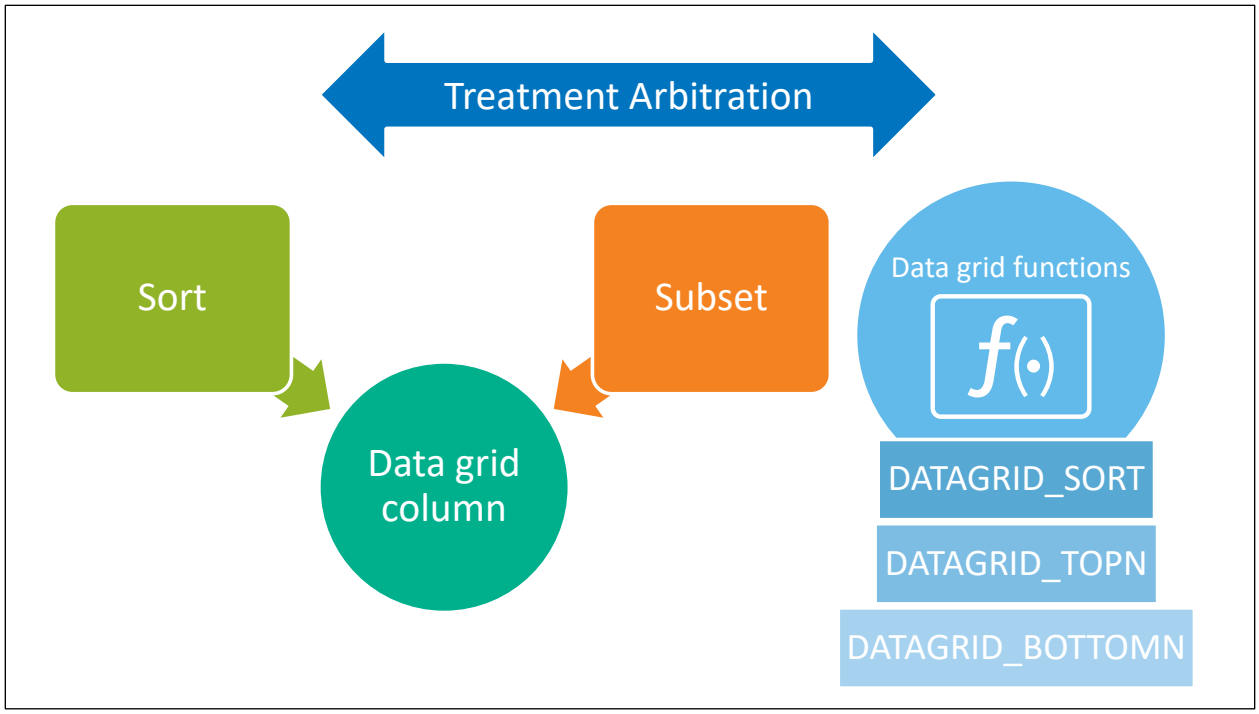


In the simplest arbitration scenario, you might want to sort and possibly also subset treatments based on the value of a data grid column.

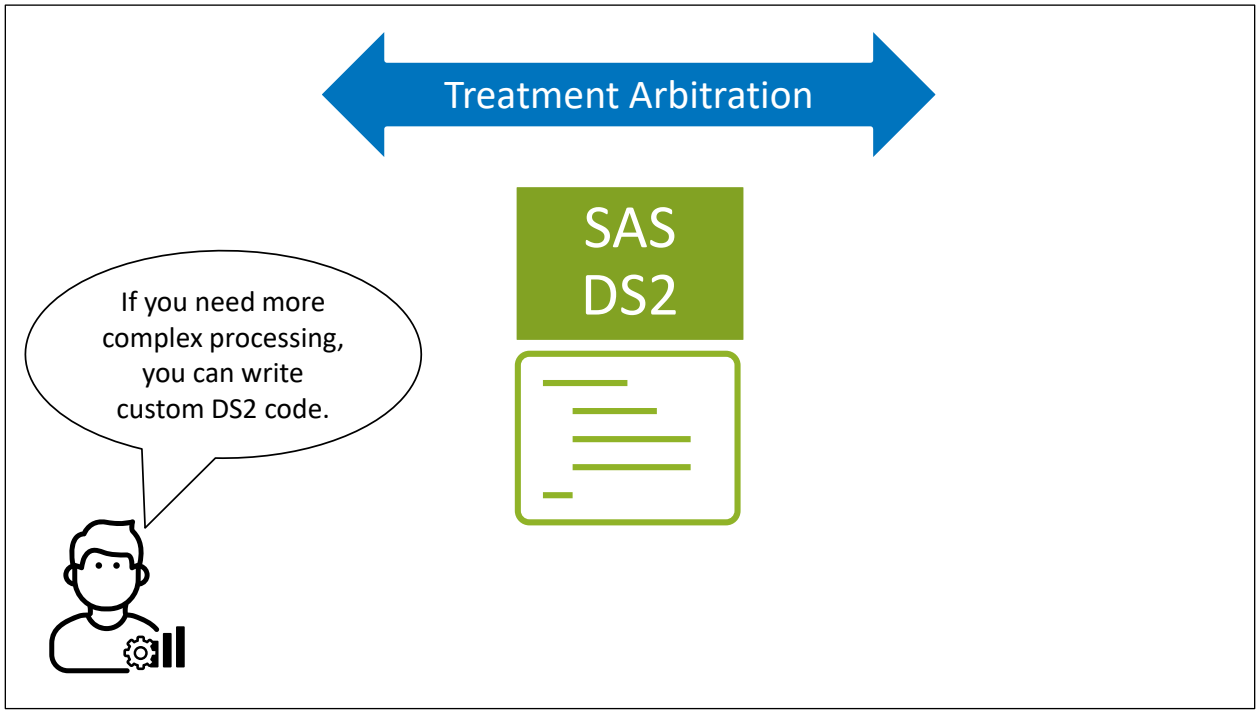


For example, you might have data grid columns for treatment custom attributes that quantify the value, profitability, risk, or a predicted propensity score returned by an analytical model. You might want to sort by one of these columns and return the N highest or lowest values.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



You can use data grid functions to subset and sort column values. Some examples of data grid functions that you can use for this purpose include DATAGRID_SORT, DATAGRID_TOPN, and DATAGRID_BOTTOMN. You could use them along with other data grid functions to perform more complicated processing if needed.



If your arbitration process is more complicated than can be accomplished using data grid functions, you can write custom DS2 code to arbitrate treatments.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Using an Assignment Rule Set for Treatment Arbitration

This demonstration illustrates creating and adding an assignment rule set to arbitrate treatments based on a model propensity score.

Copyright © SAS Institute Inc. All rights reserved.

Practice

In this practice, you create and add an assignment rule set to arbitrate treatments based on a fixed treatment attribute.

Copyright © SAS Institute Inc. All rights reserved.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

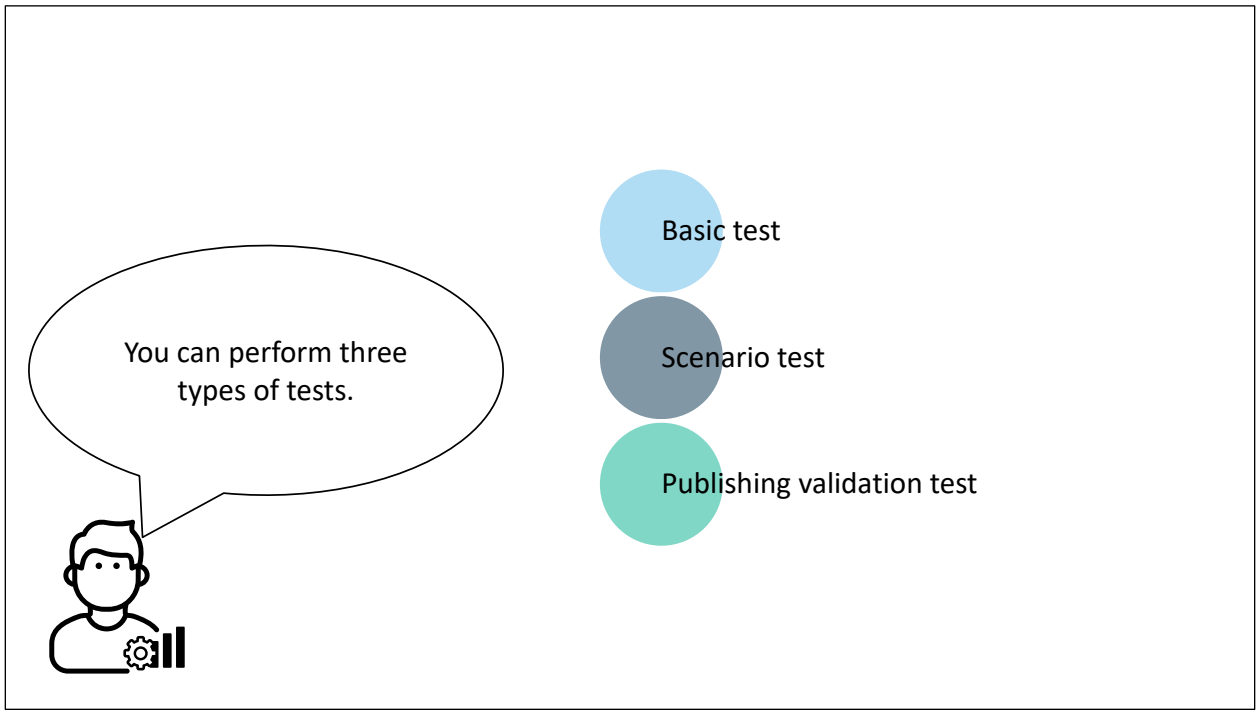
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Lesson 5 Testing, Publishing, Validating, and Troubleshooting

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

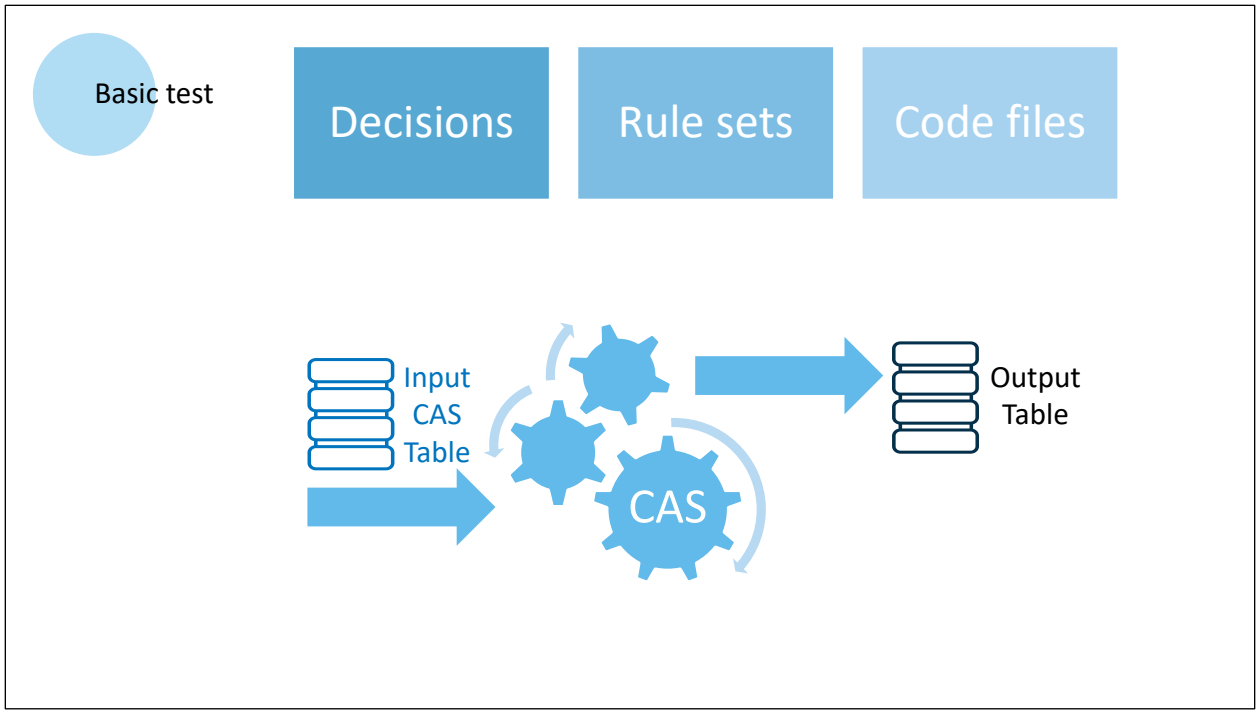
5.1 Testing

Introduction to Testing

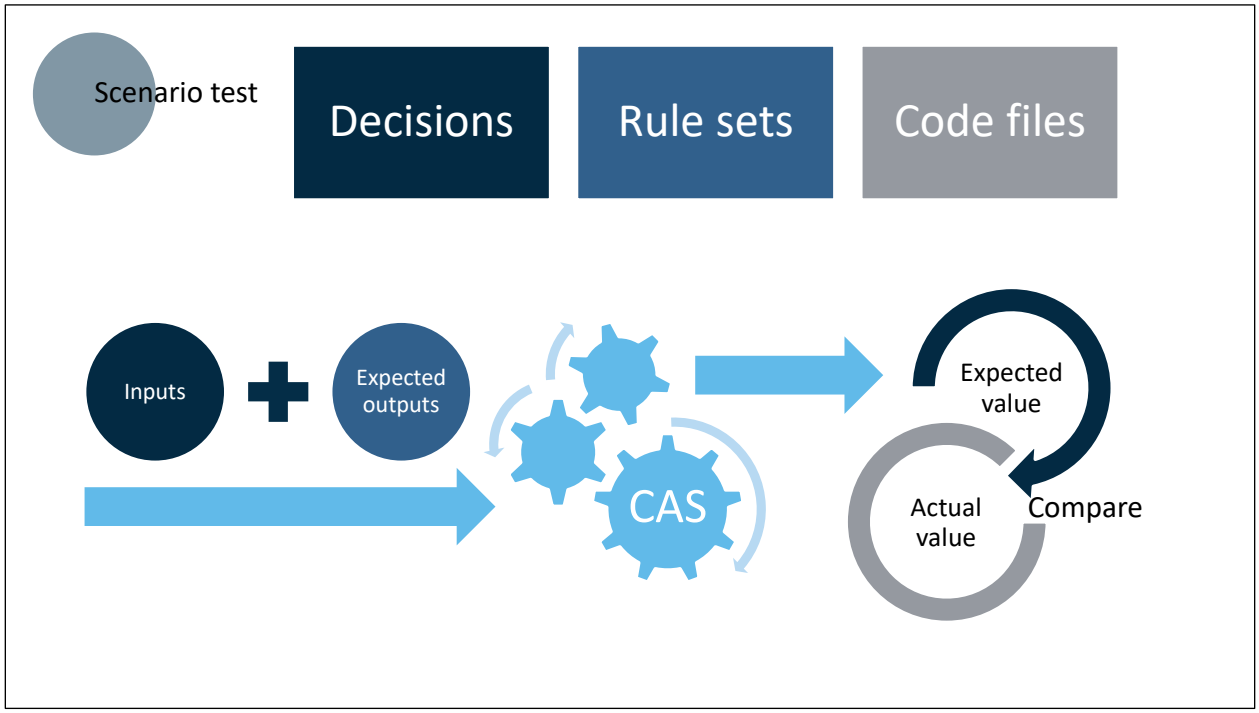


You can perform three types of tests. Testing is optional, but recommended. Testing enables you to discover any problems before a decision or rule set is published and incorporated into a production system.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

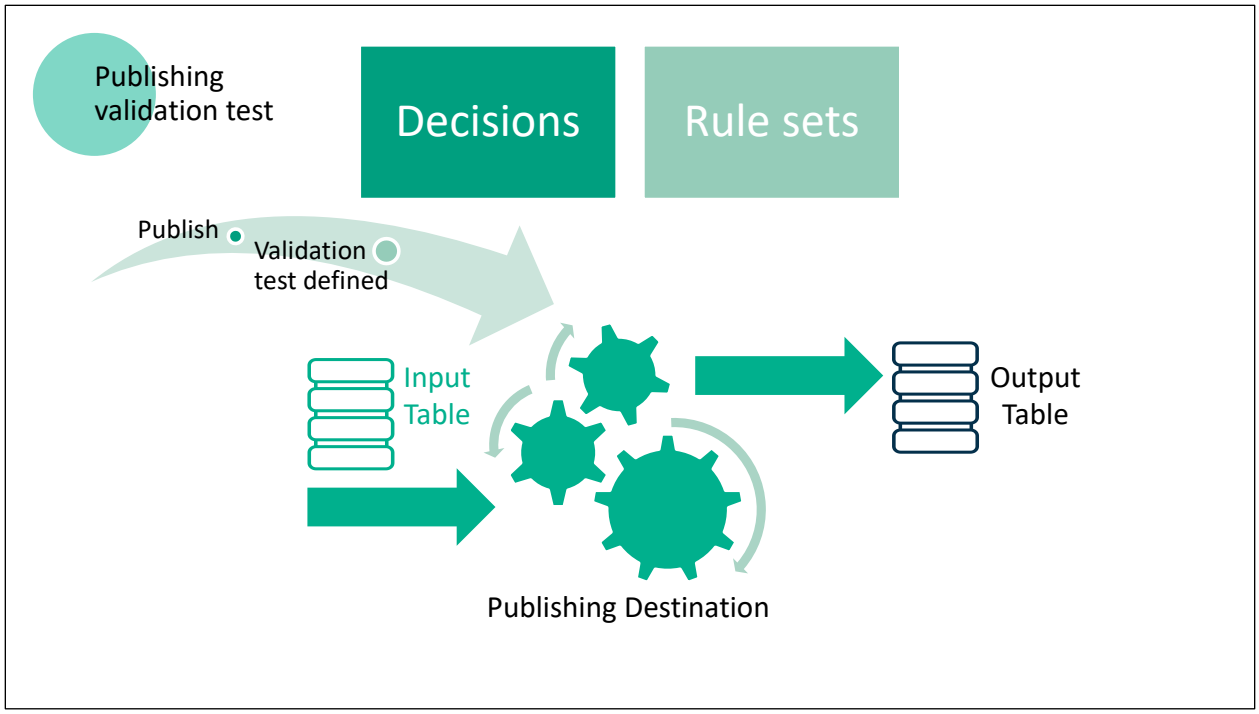


You can perform basic testing for decisions, rule sets, and code files. When you perform basic testing, the object is executed in CAS using an input table that you specify. The test generates and displays a table containing output values.

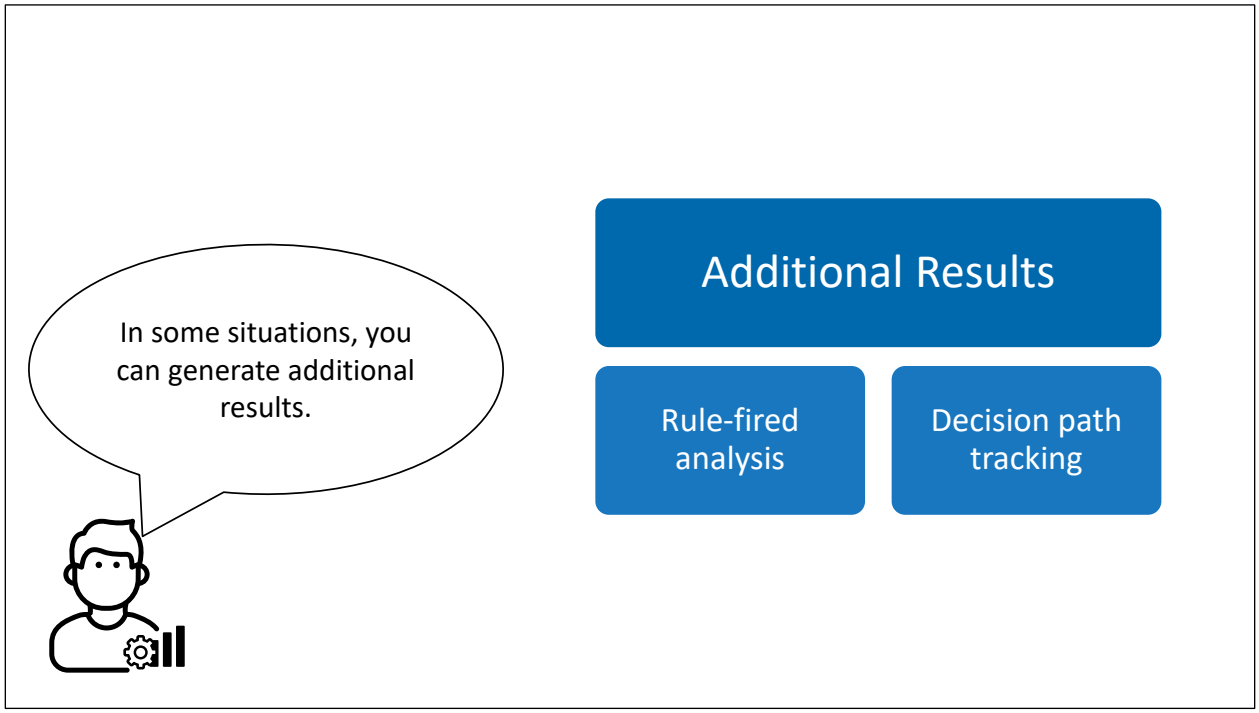


You can perform scenario testing for decisions, rule sets, and code files. When you perform a scenario test, you specify input variable values. You can also specify the corresponding output variable values that you expect the test to generate. A scenario test identifies differences between the expected and actual output values.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

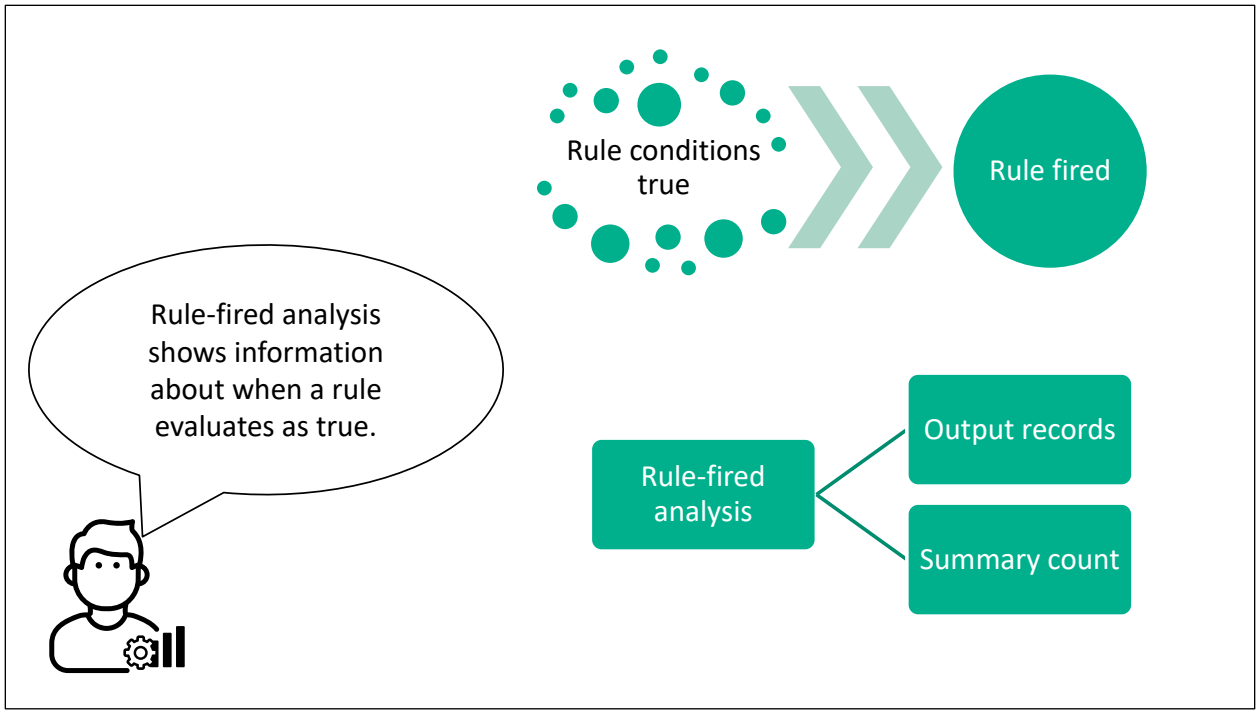


You can execute publishing validation tests for decisions and rule sets. When you publish a decision or rule set, a validation test is automatically defined in the selected publishing destination. The test executes the decision or rule set in the publishing destination using the input data that you specify.

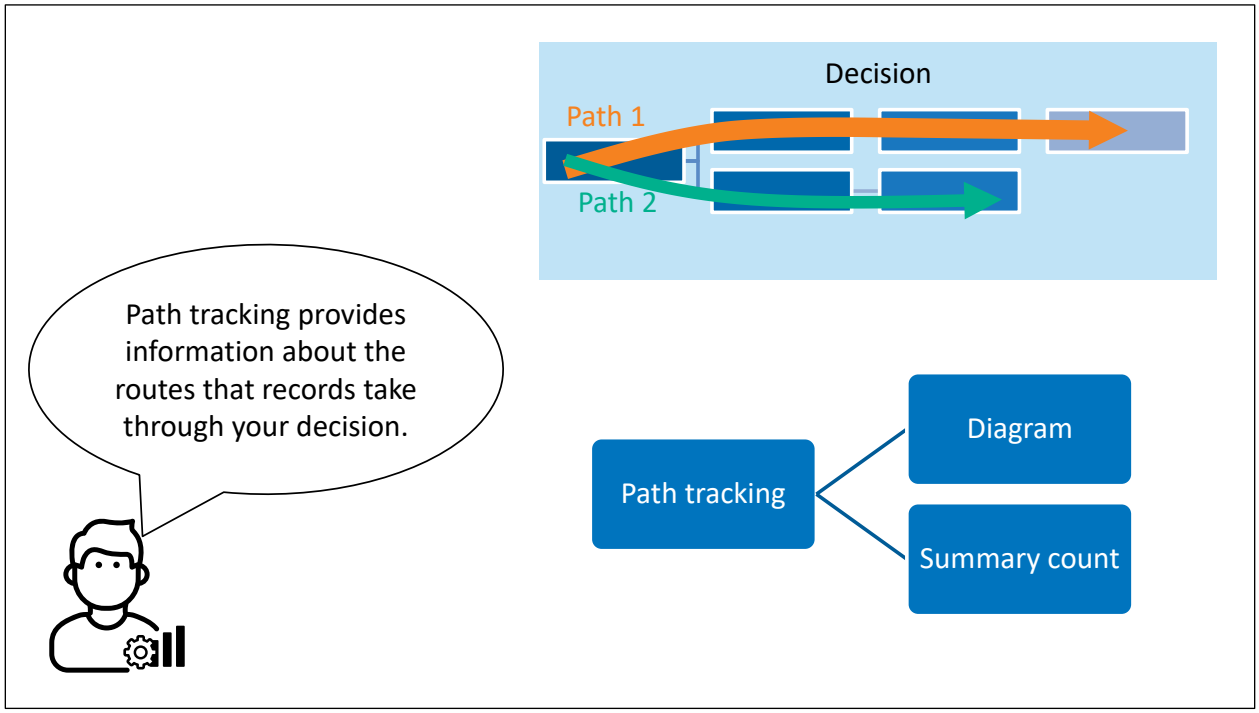


In some situations, you can perform rule-fired analysis and decision path tracking.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

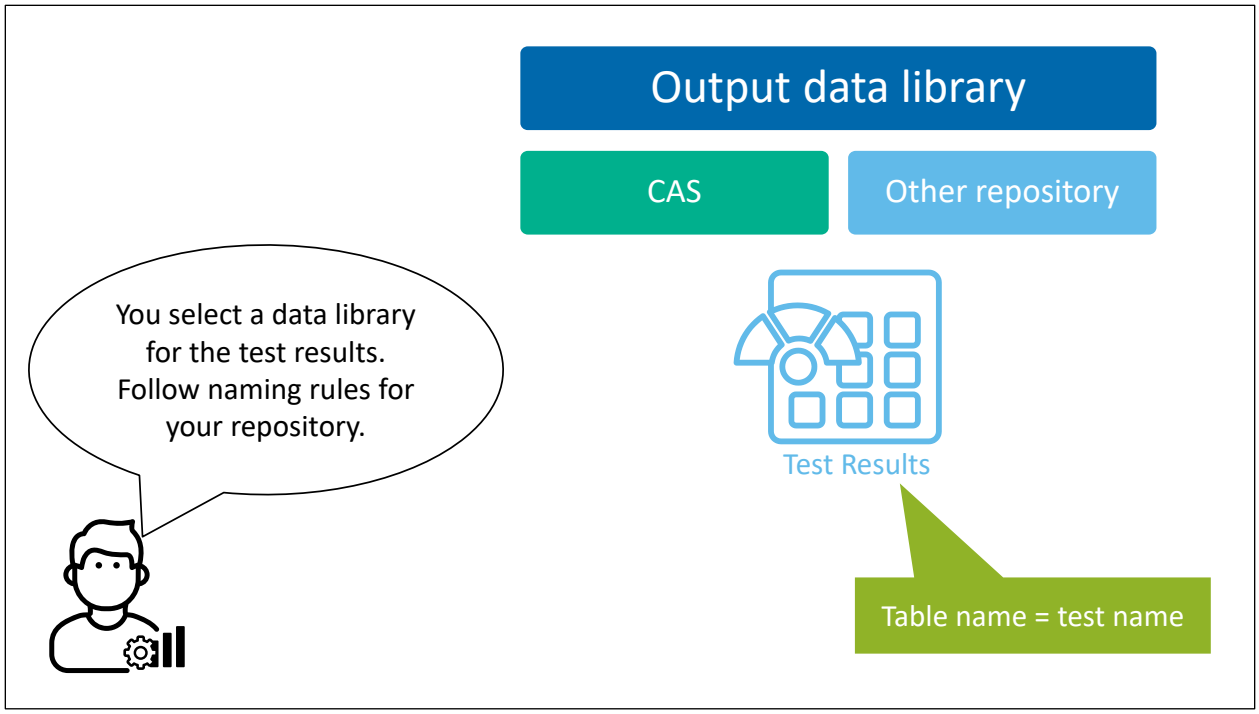


If a rule's conditions evaluate to true, then the rule is said to have *fired*. Rule-fired analysis results include output records with details for each rule that evaluates to true and a summary of the count of how many times each rule fired.

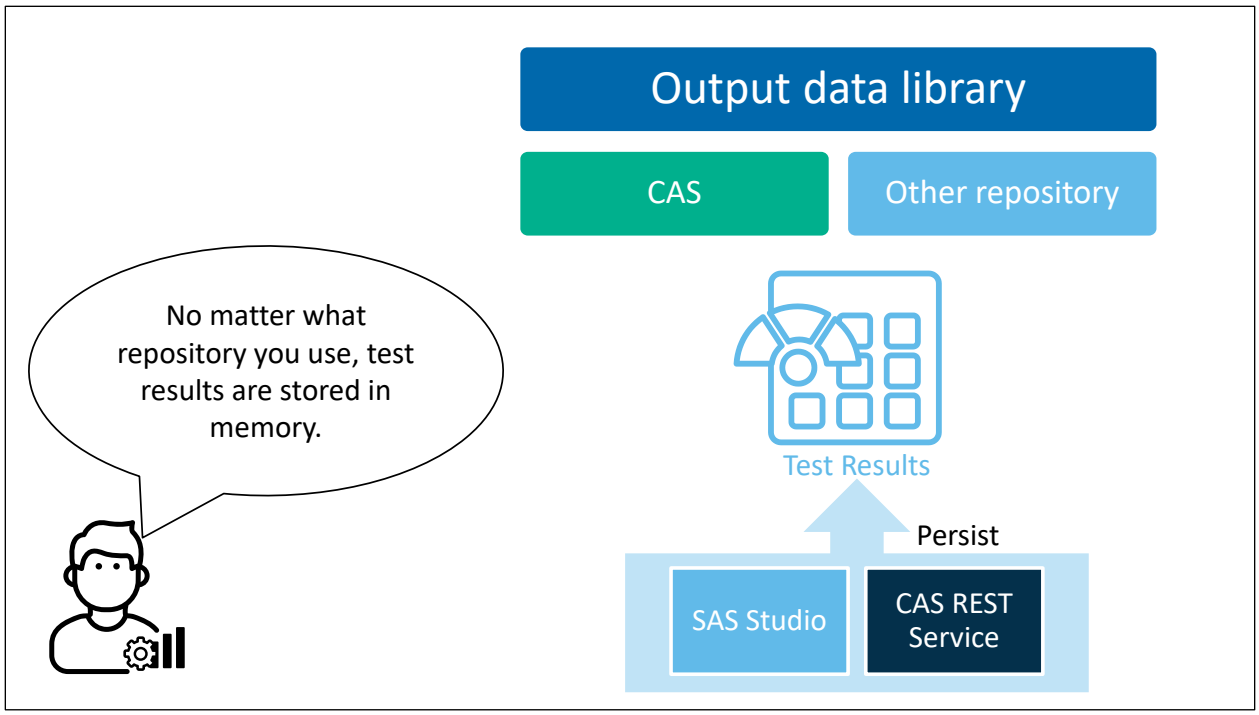


Path tracking provides information about the routes that input records take through your decision. Path tracking results include a diagram that shows the flow of the input records through the nodes in the decision. It also includes a summary count of records passing through each node.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

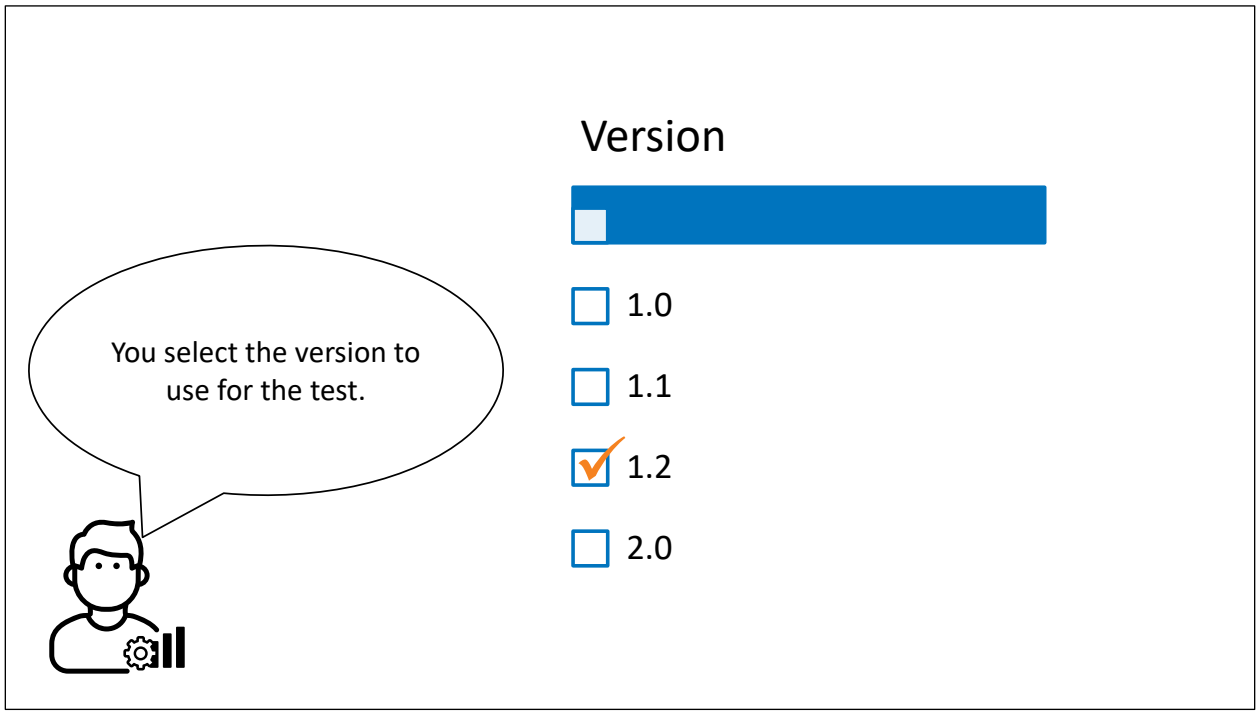


When performing basic and scenario tests, you select an output data library where you want to create a table containing the test results. This library can be a CAS library or it could point to a different data repository. The test results will be stored in a table that is named using the same name that you specify for the test or scenario. If you are using a repository other than CAS, you should make sure that the name of the file follows the naming rules for the repository that you are using.

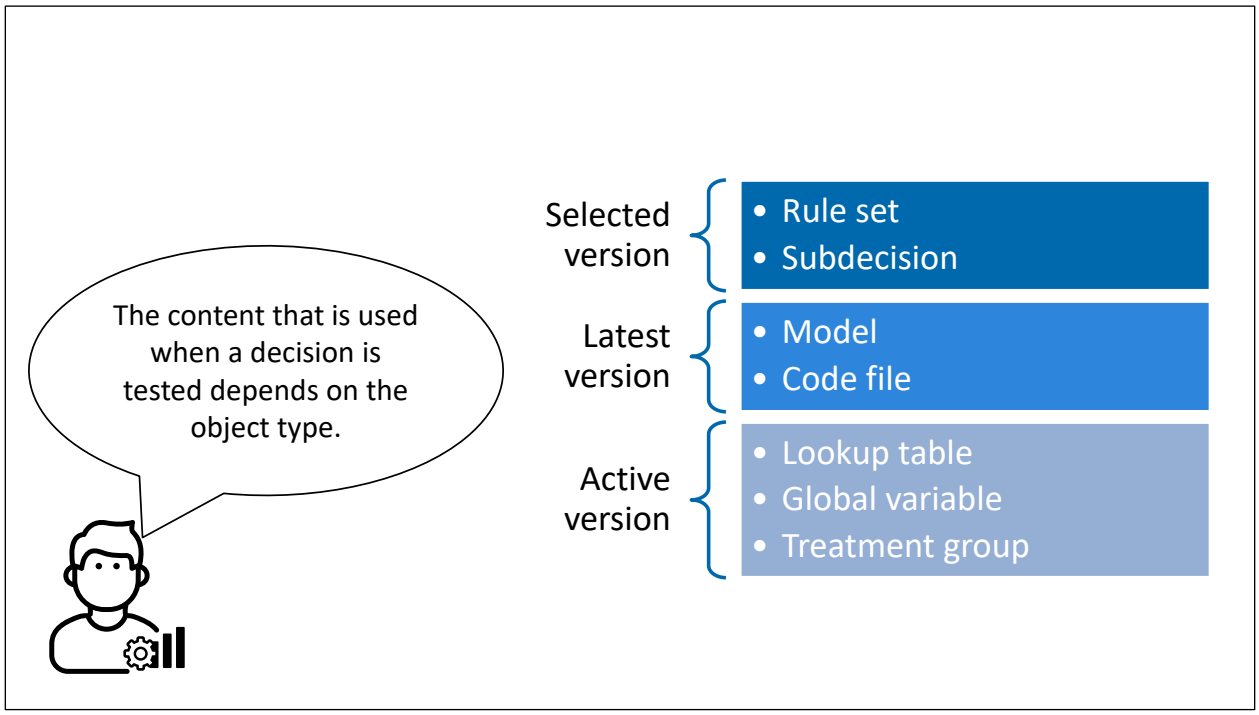


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

No matter whether you use CAS or another repository to store the results, they are stored in memory and do not persist if the server is restarted. If you want to persist the test results, you can access the test output using SAS Studio or the REST service in CAS.

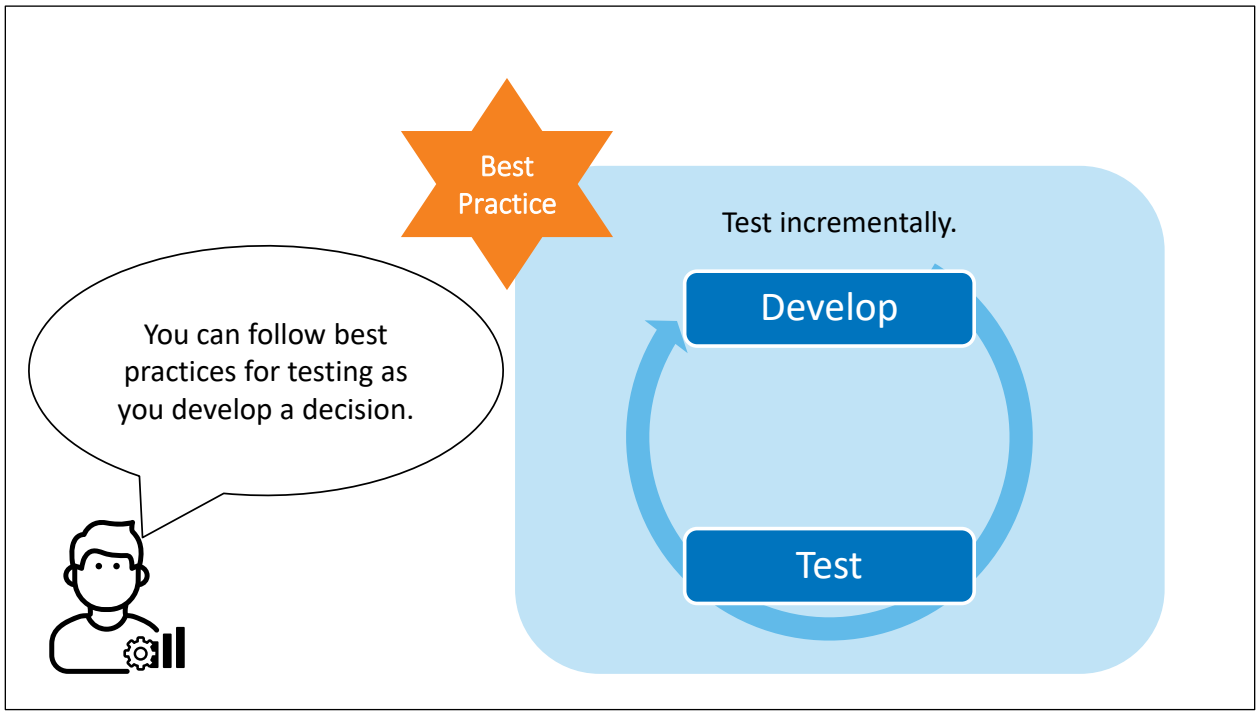


You can also choose the version of the object that you want to use for the test.



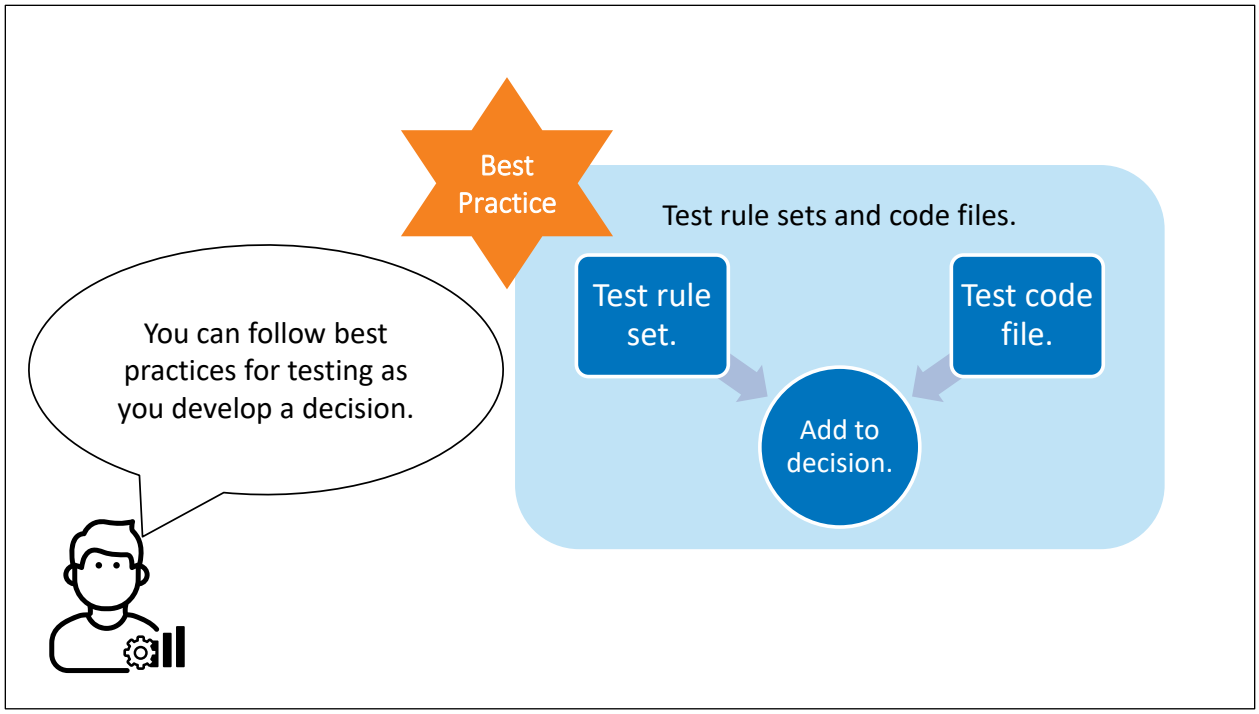
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The content that is used when a decision is tested depends on the object type. For rule sets and subdecisions, the test uses the version selected in the decision. For models and code files, the test uses the latest version. For lookup tables, global variables, and treatment groups, the test uses the active version.



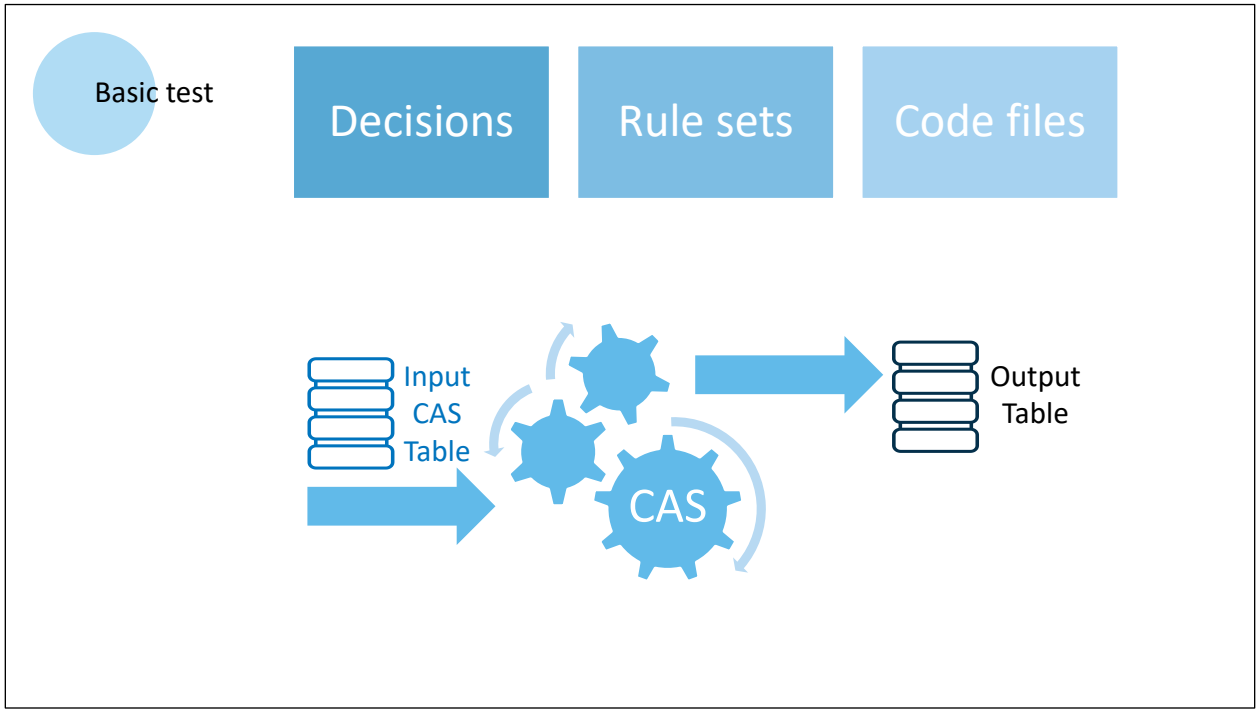
You can follow best practices for testing as you develop a decision. It's a best practice to test incrementally as you develop decisions and decision objects.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



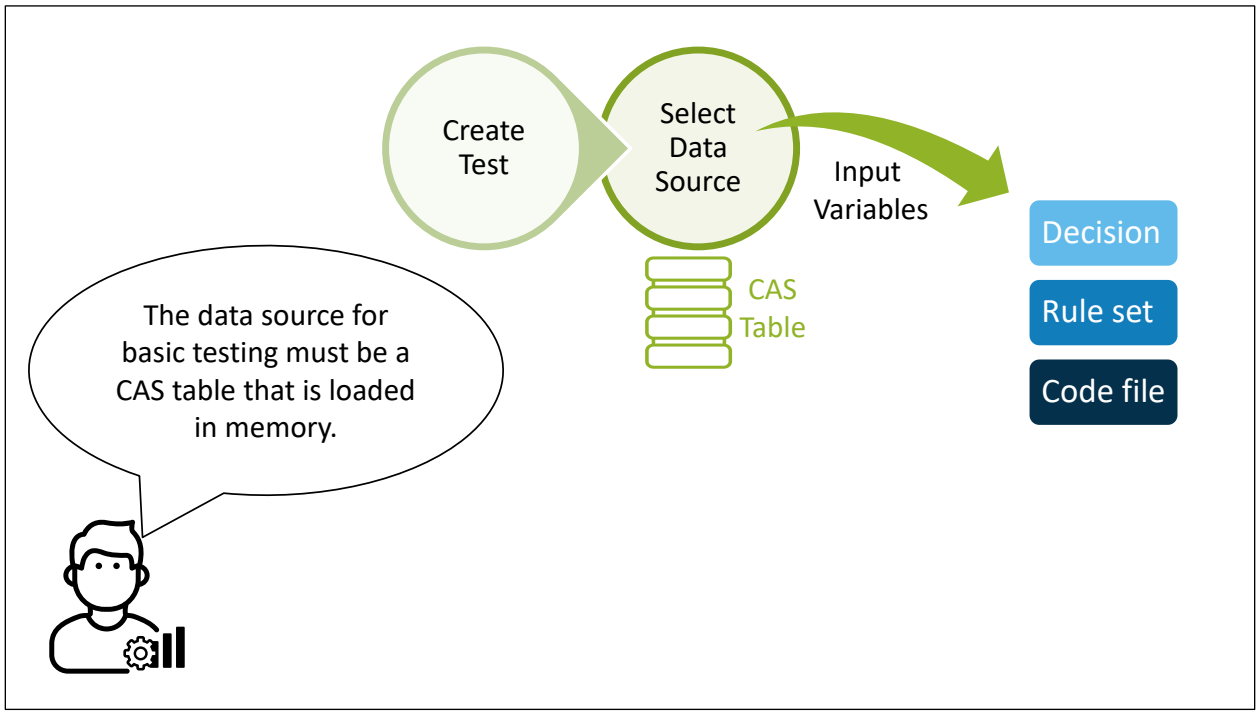
It's also a best practice to test rule sets and code files before adding them to a decision.

Basic Testing

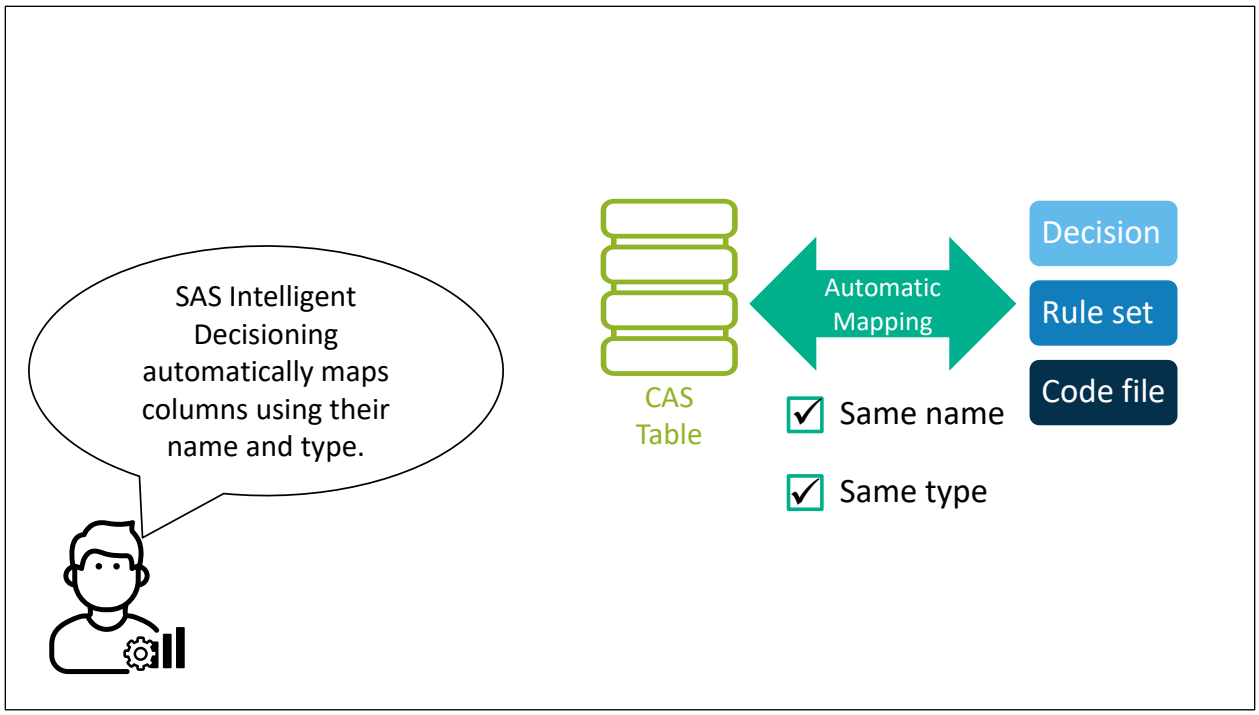


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

You can perform basic testing for decisions, rule sets, and code files. When you perform basic testing, the object is executed in CAS using an input table that you specify. The test generates and displays a table containing output values.

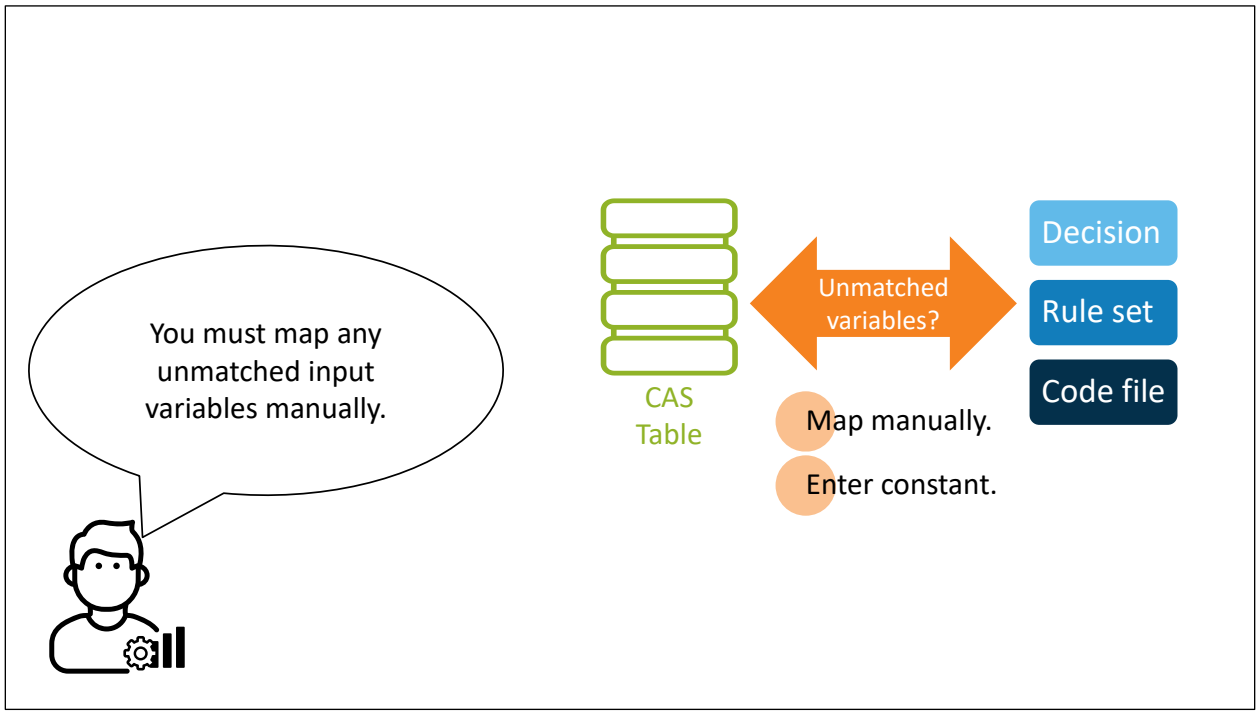


When you create a test, you select a data source to provide input variable values for testing. The data source must be a CAS table that is loaded in memory.

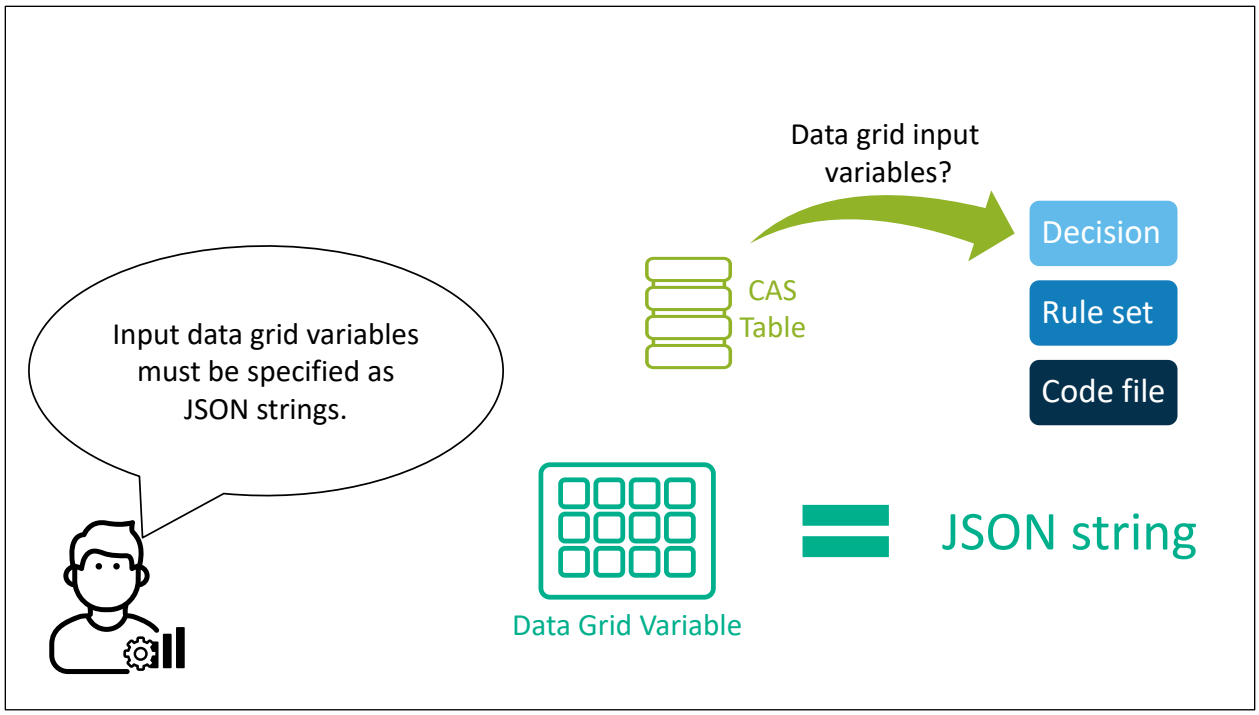


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

SAS Intelligent Decisioning automatically maps columns in the input table to the input variables in the object being tested when the names and data types of the variables match those of the table columns.

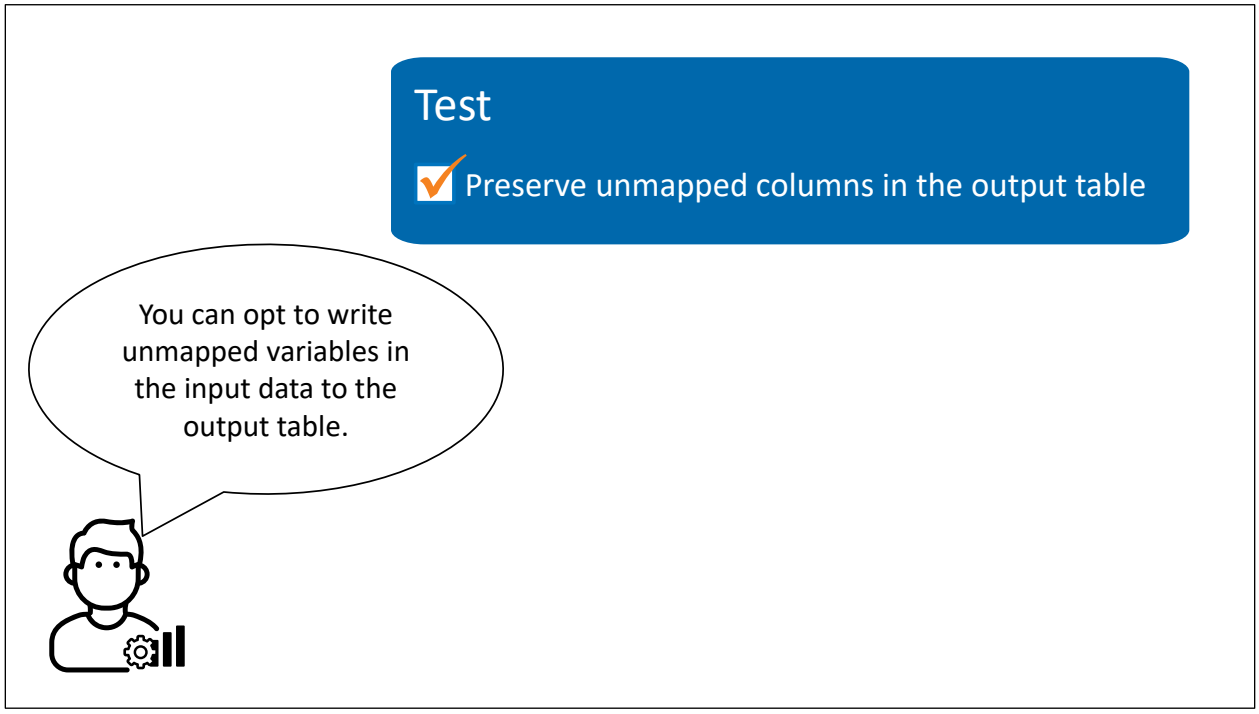


If any input variables do not have matching columns, you must map the variables manually. You can also enter constant values for variables of type Decimal, Integer, and Character.

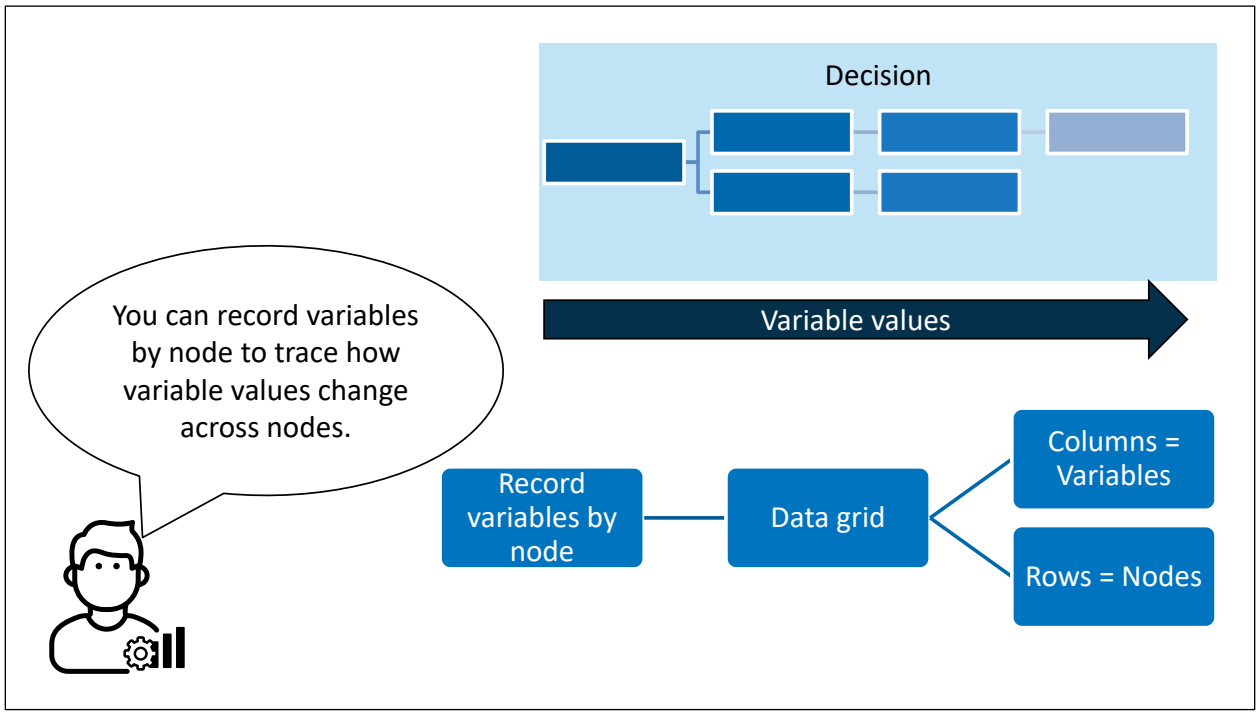


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

If the input variables to the object that you are testing include data grids, you must specify the input as JSON strings that define the structure and content of the data grid.



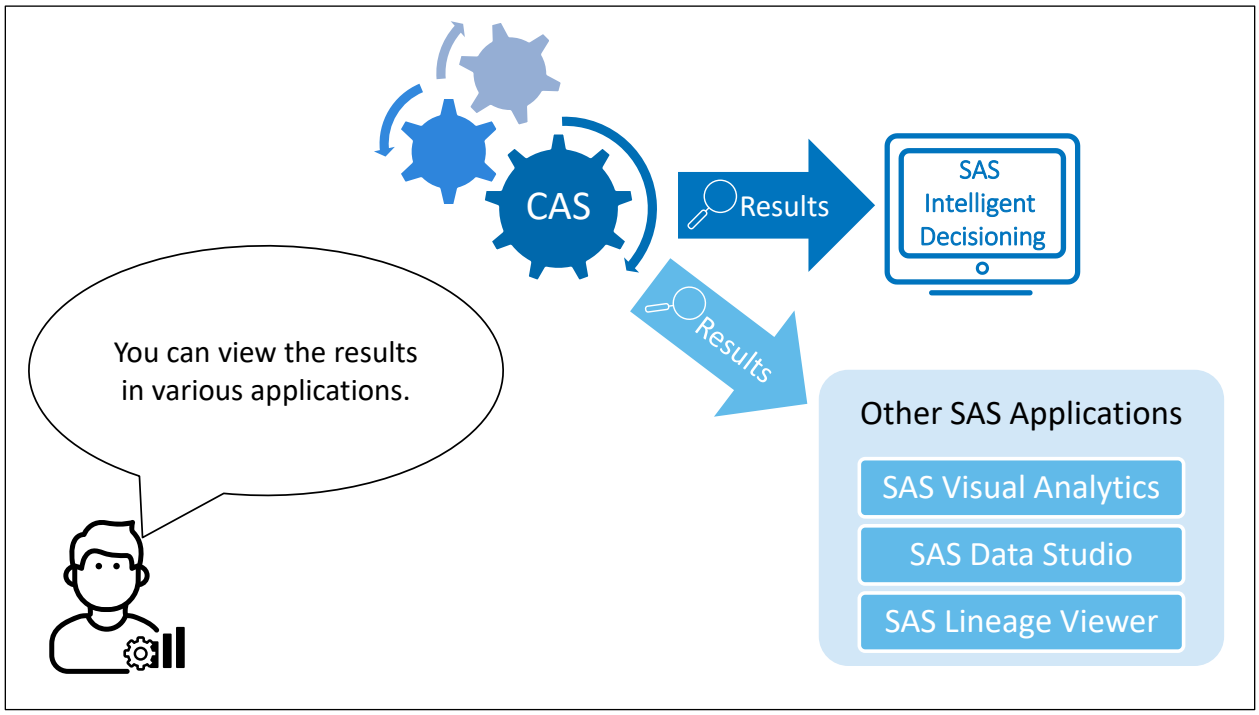
You can enable an option to preserve unmapped columns in the output table if you want columns in the input data that are not mapped to a decision output variable to be written to the output table.



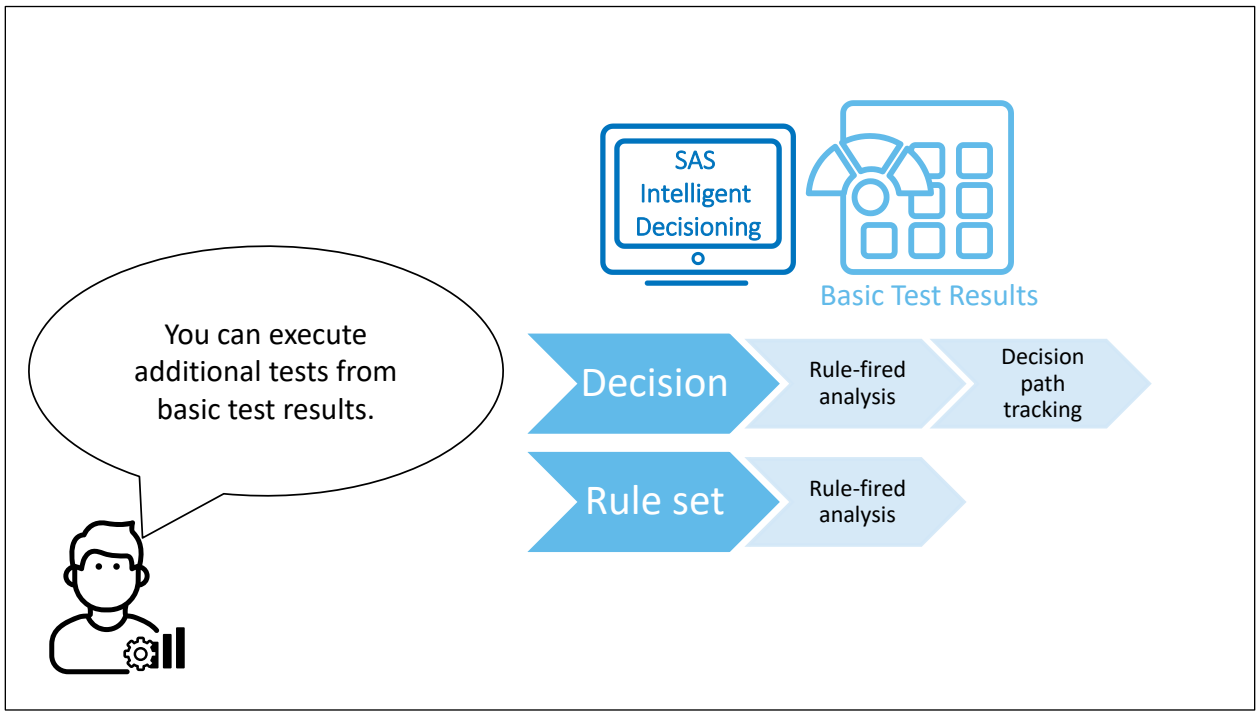
When you test a decision, you can select the **Record variable values by node** option to trace how variable values change for each node in the decision flow. When you select this option, SAS

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Intelligent Decisioning creates an output data grid variable named nodeTraceDataGrid. This data grid contains one column for every variable in the decision and a row for each rule set, model, subdecision, and code file node in the decision. Branch nodes and record contact nodes are not included.

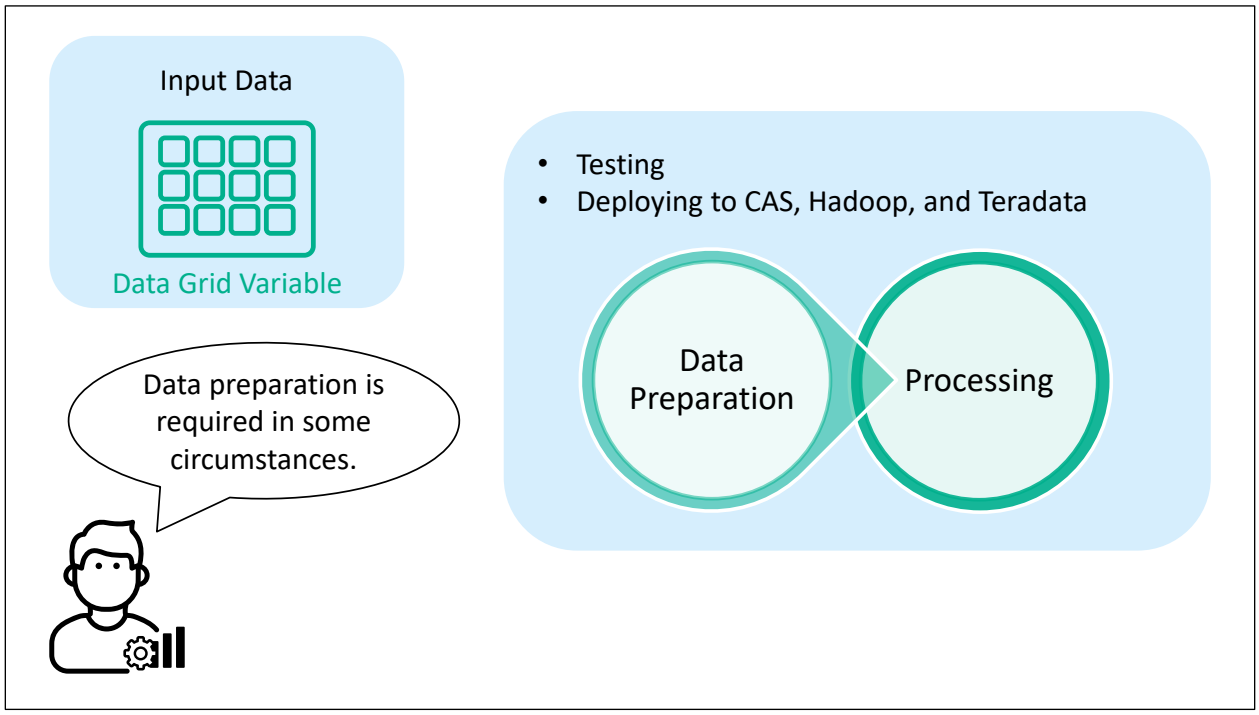


After you run a test, you can view the test results in the Intelligent Decisioning interface. You can also work with the output table in other SAS applications.

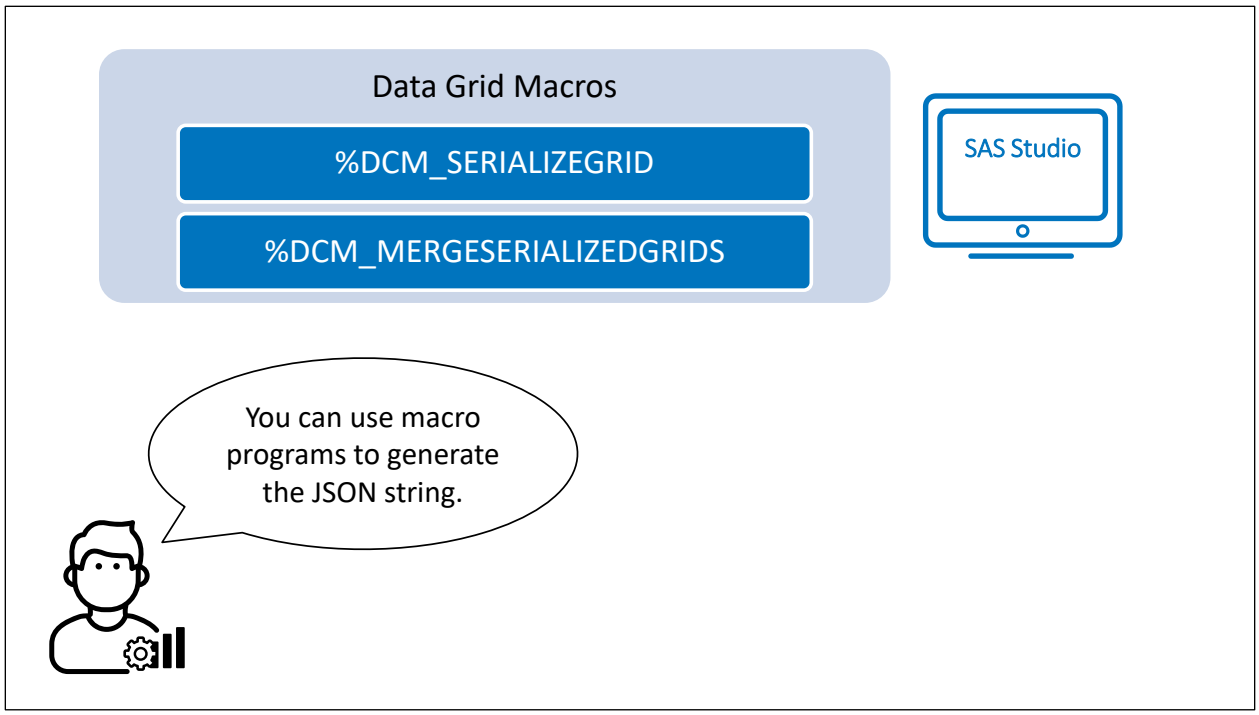


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

When viewing basic test results in SAS Intelligent Decisioning, you can execute additional types of tests for decisions and rule sets. For both you can choose to run rule-fired analysis. If you are testing a decision, you can also choose to run path tracking.

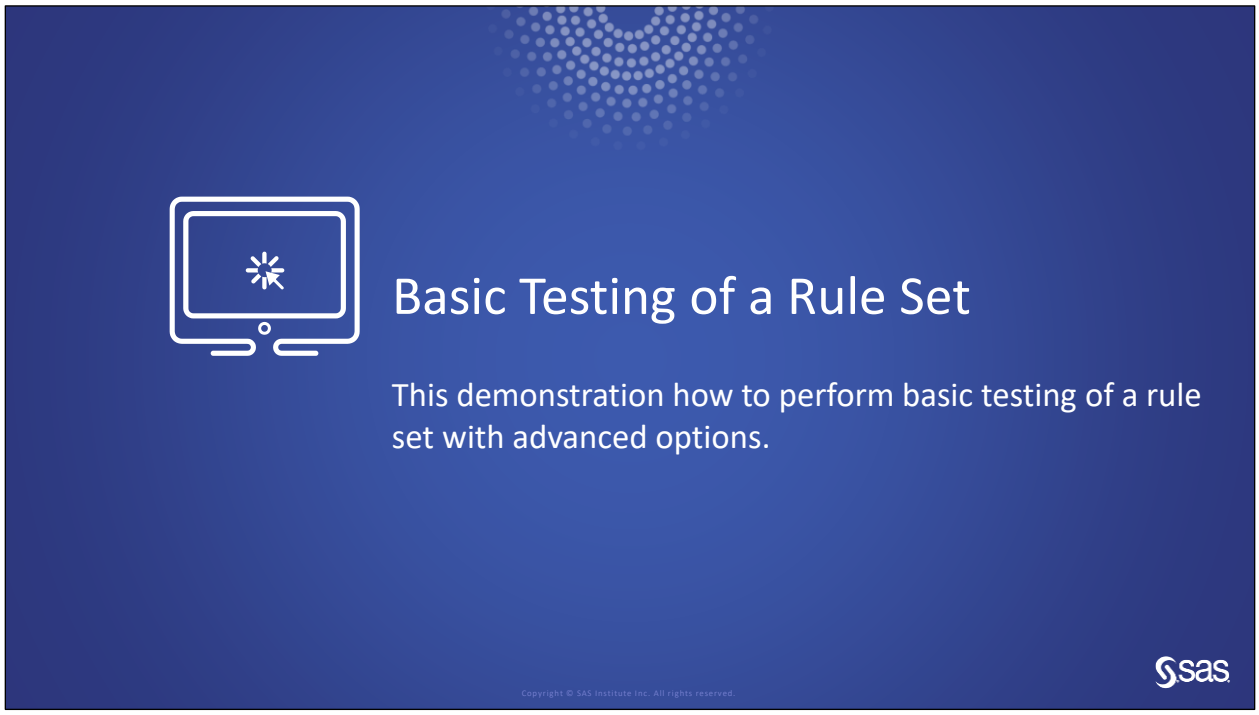


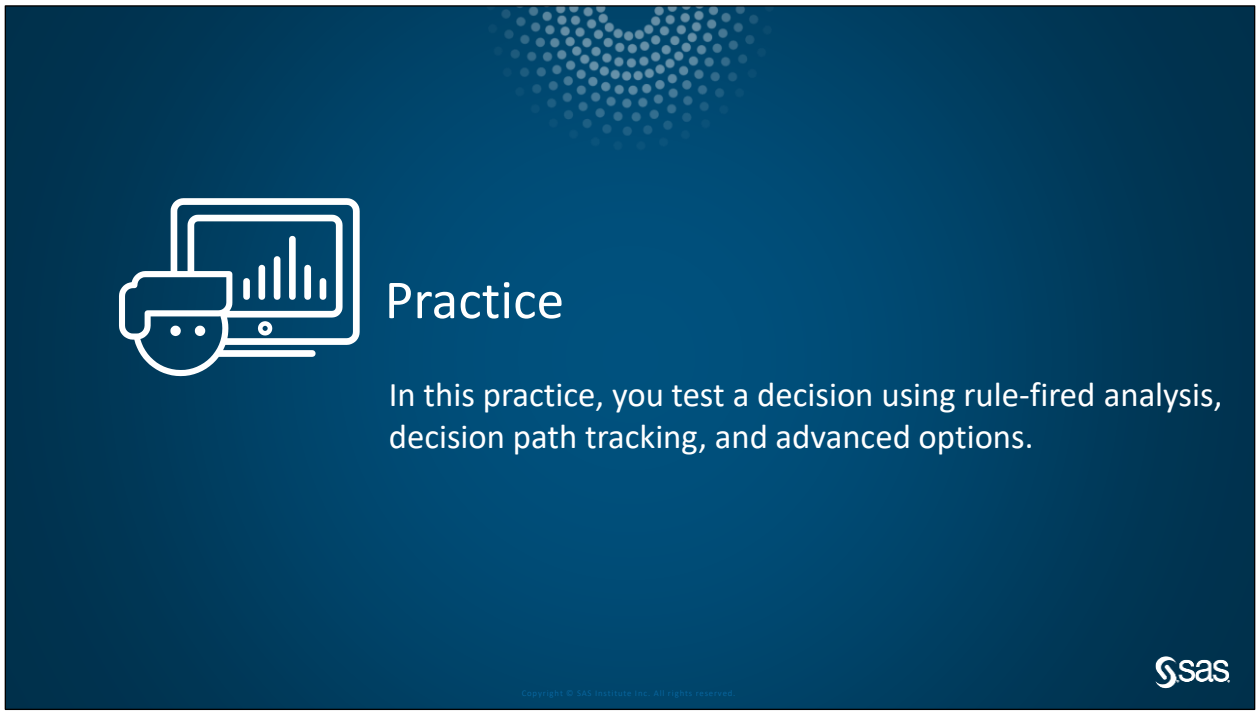
Recall that when the input data to your decision or rule set includes data grids, some steps to prepare for processing are required in some circumstances.



Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

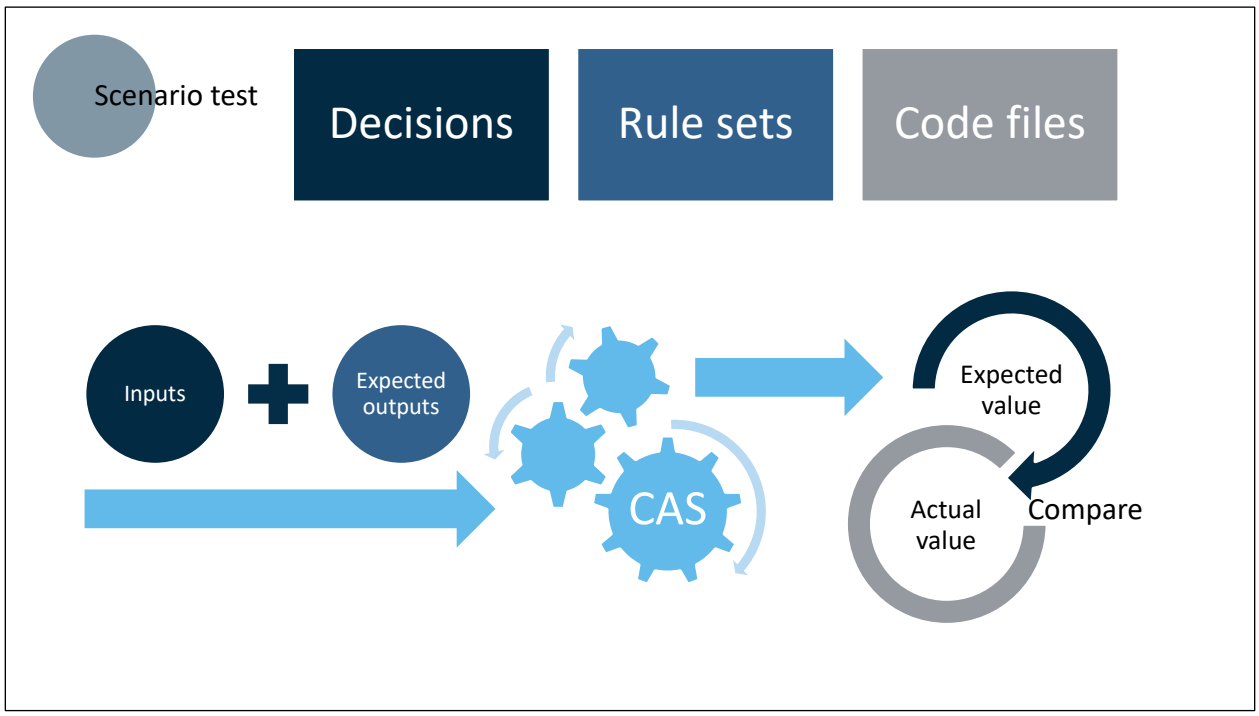
You can use two macro programs provided with SAS Intelligent Decisioning to generate JSON strings for data grid variables. You can use SAS Studio to run these programs. Refer to the lesson on data grids for more information.



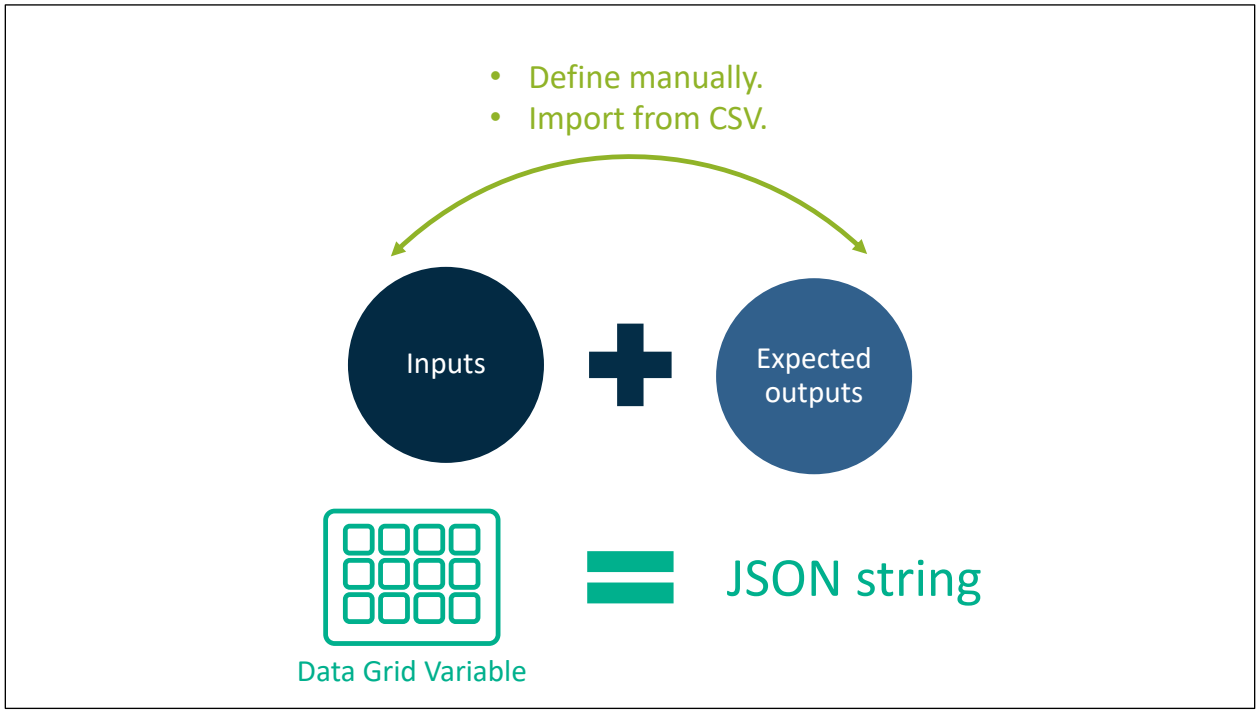


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Scenario Testing

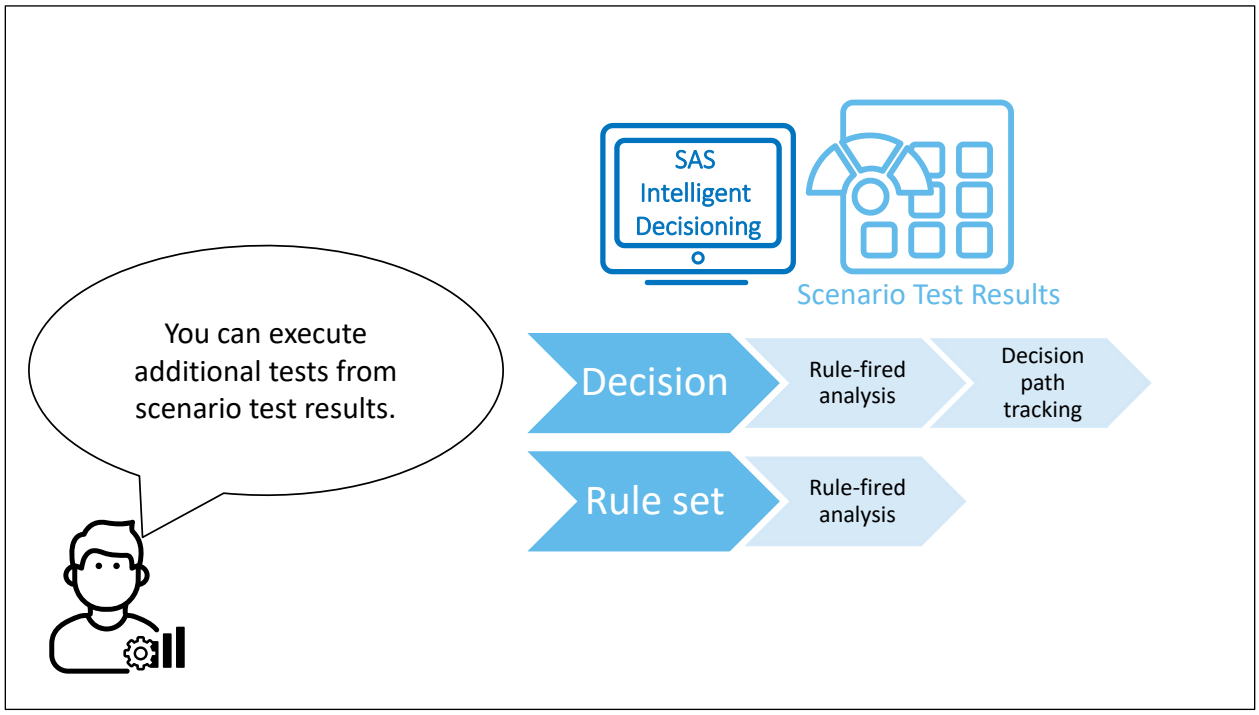


You can perform scenario testing for decisions, rule sets, and code files. When you perform a scenario test, you specify input variable values. You can also specify the corresponding output variable values that you expect the test to generate. A scenario test identifies differences between the expected and actual output values.

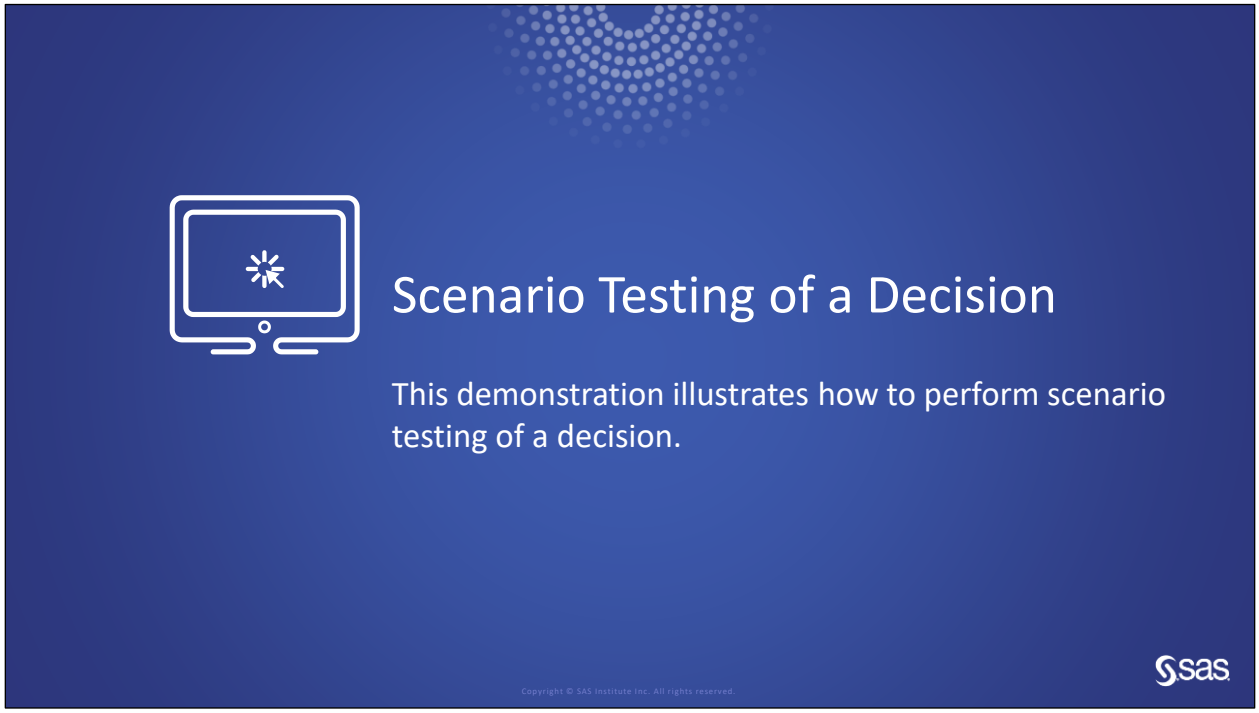


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

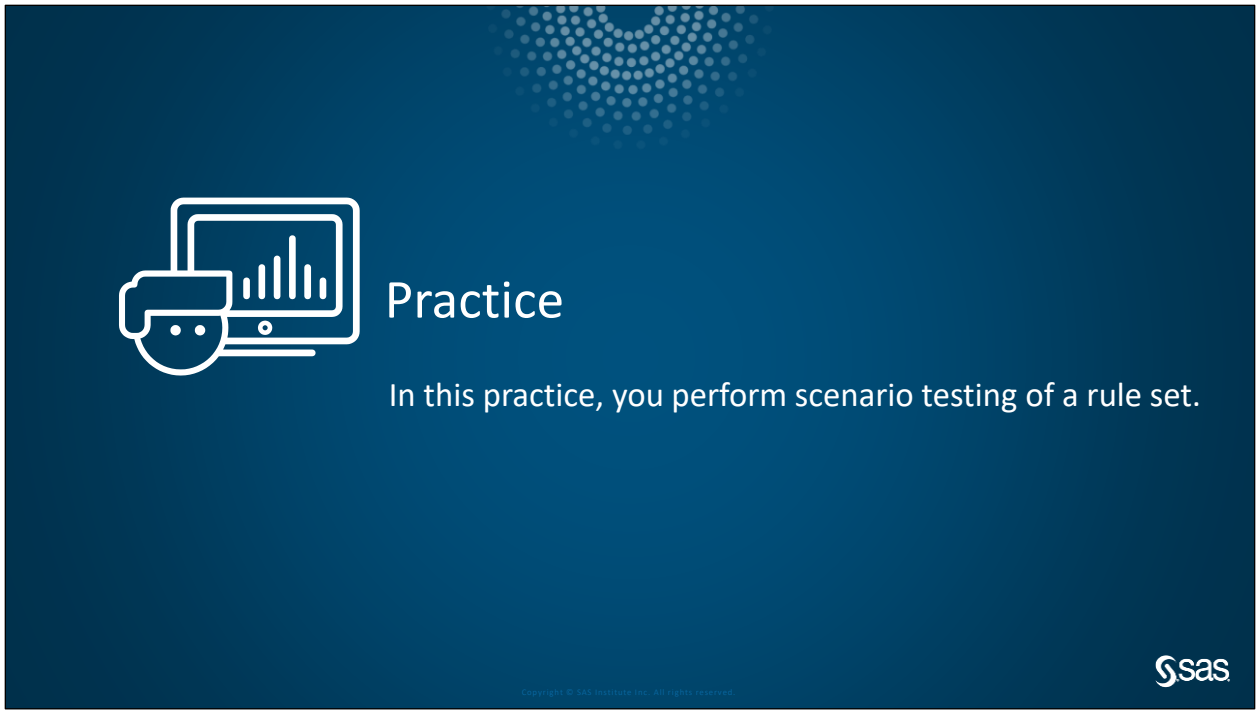
When you create a scenario test, you can define input and output values manually or import them from a CSV file. You must specify data grid variables as JSON strings that define the structure and content of the data grid.



When viewing scenario test results in SAS Intelligent Decisioning, you can execute additional types of tests for decisions and rule sets. For both you can choose to run rule-fired analysis. If you are testing a decision, you can also choose to run path tracking.



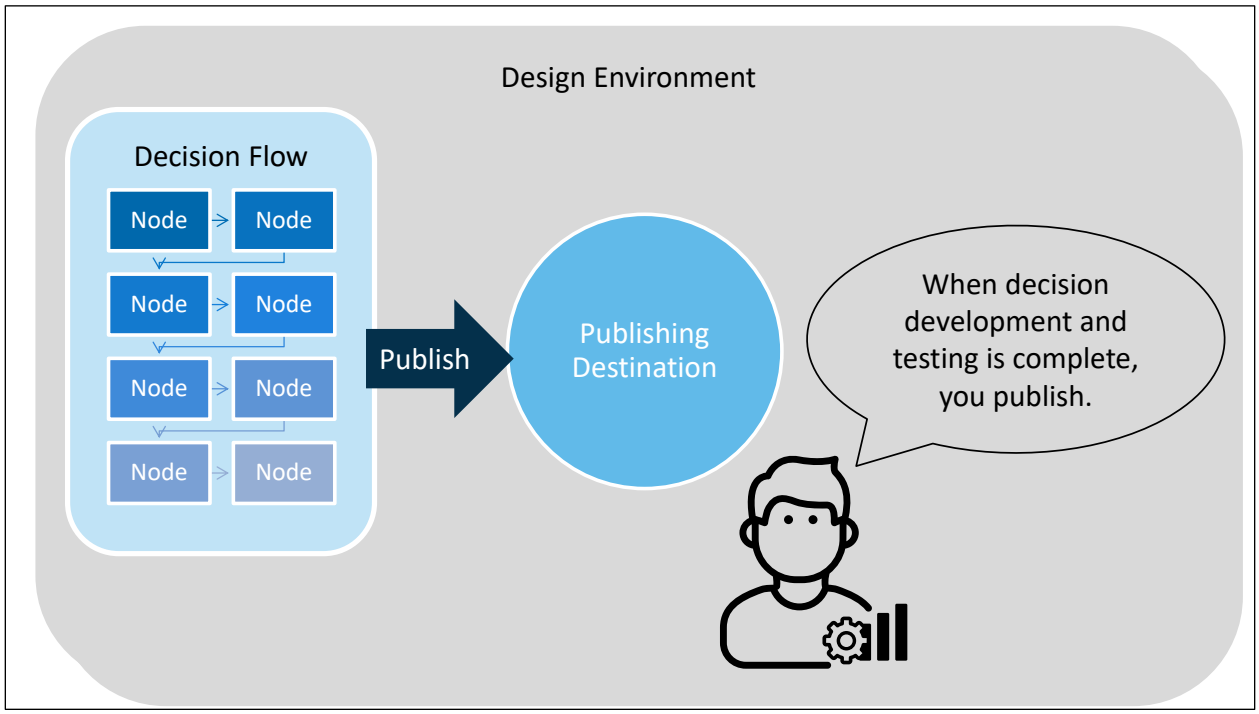
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

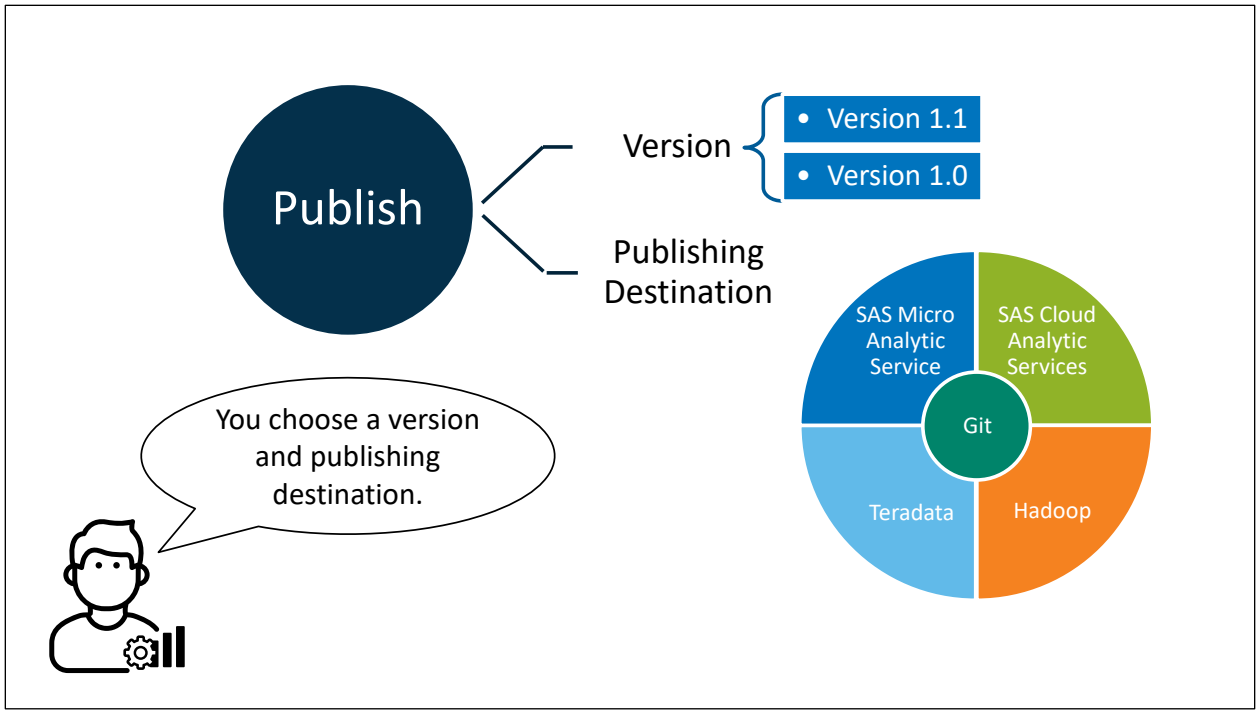
5.2 Publishing and Validating

Publishing

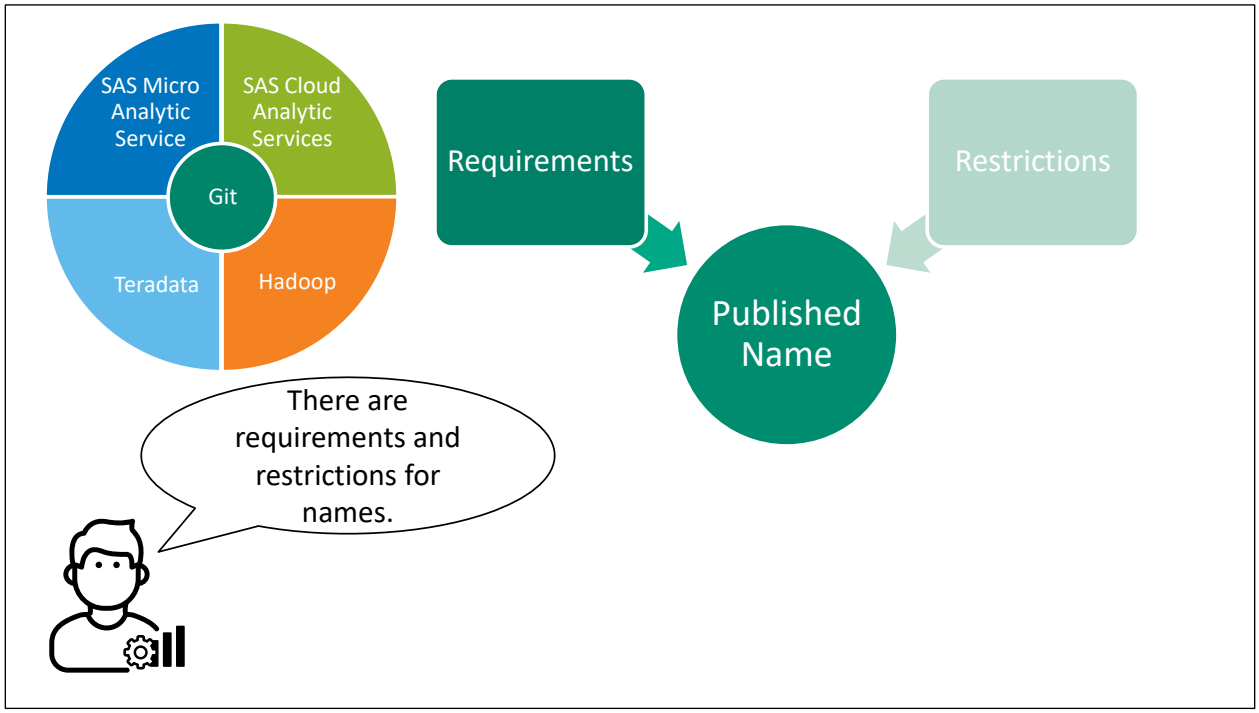


When decision development and testing is complete, you publish the decision. Publishing a decision creates an entity that can be managed and run in another environment.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

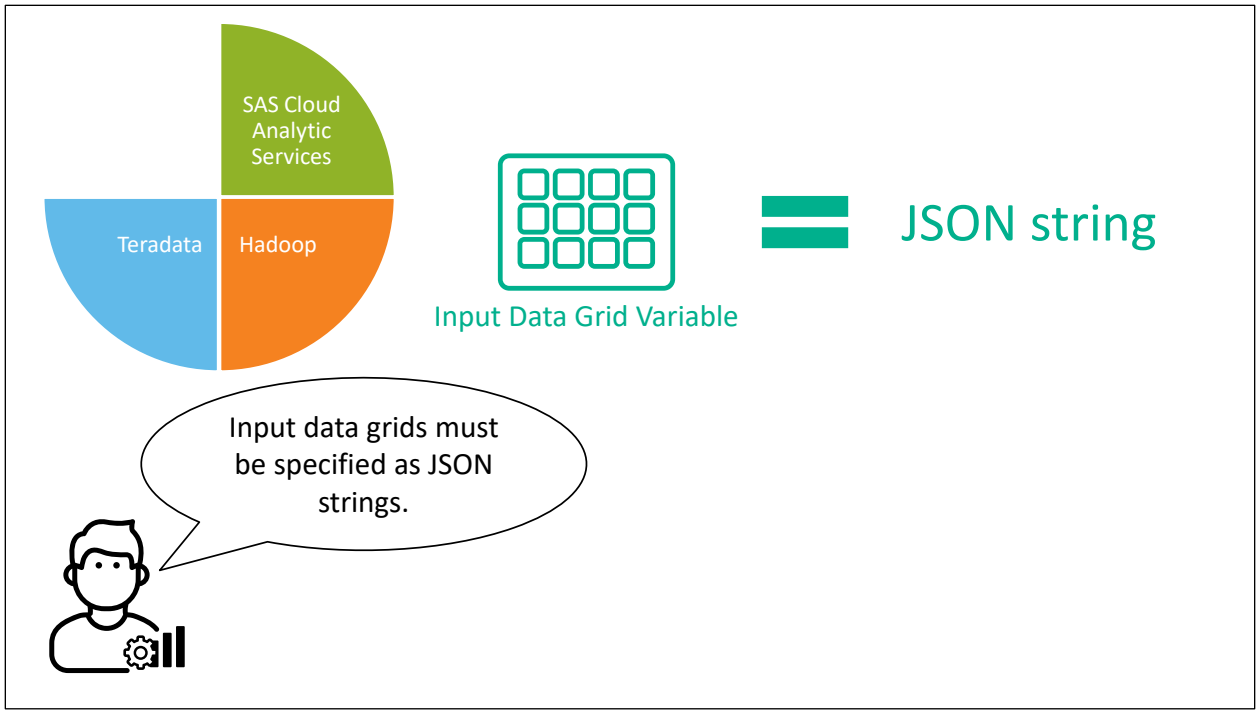


When you publish, you choose the version of the decision or rule set and the publishing destination. Recall that the publishing destinations that are available to you depend on your software license. They might also depend on configuration steps taken by an administrator.

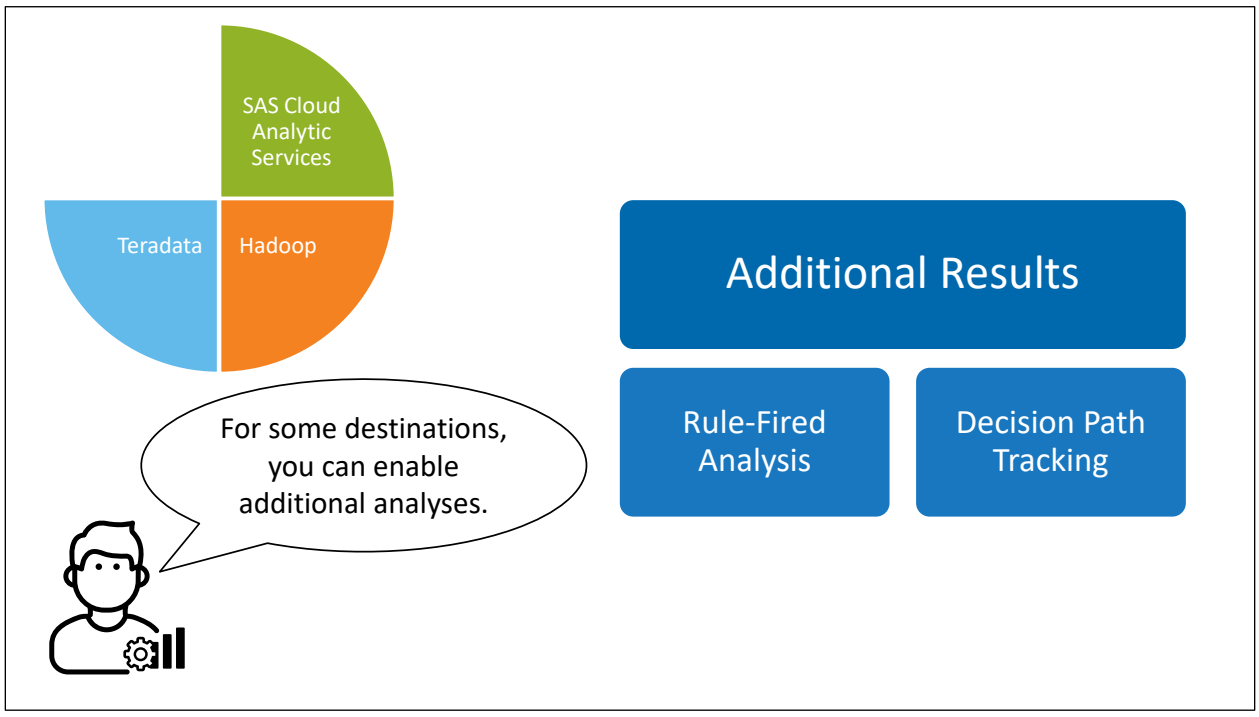


One of the properties of the published object is its name. There are requirements and restrictions for this name for each publishing destination. Refer to *SAS Intelligent Decisioning: User's Guide* for more information.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

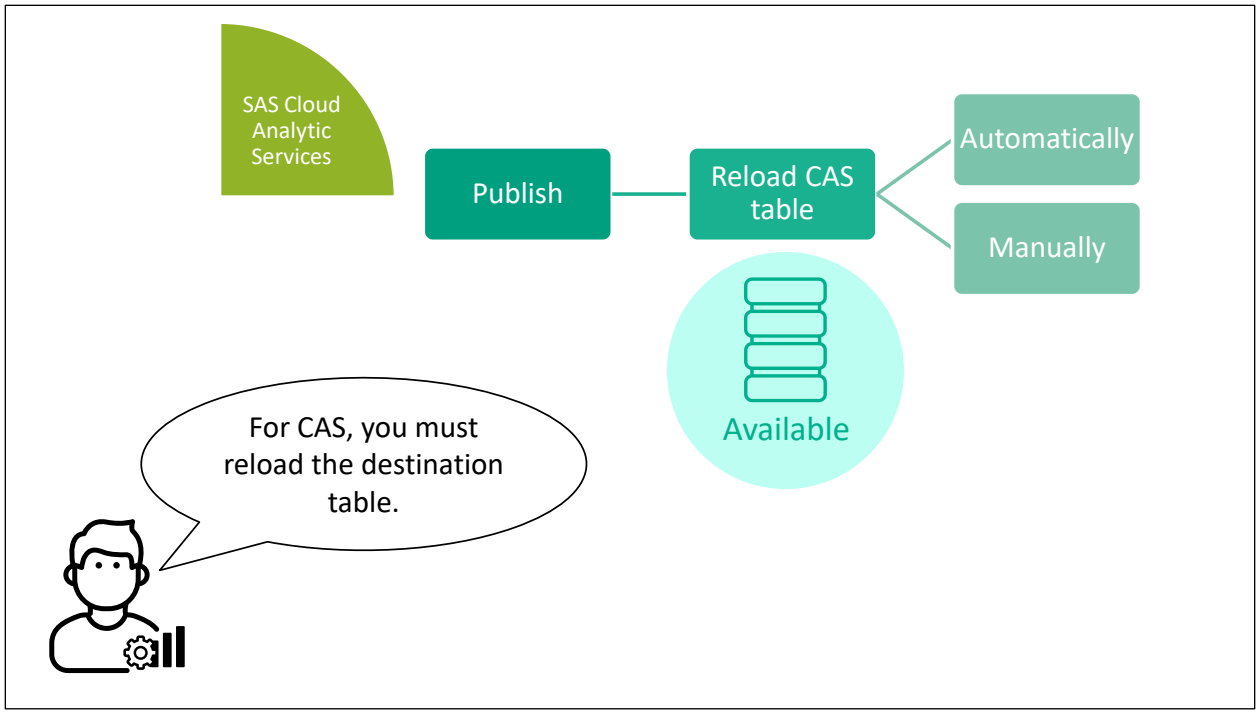


When you publish to CAS, Hadoop, or Teradata and the input data contains data grids, you must specify the input as JSON strings that define the structure and content of the data grid.



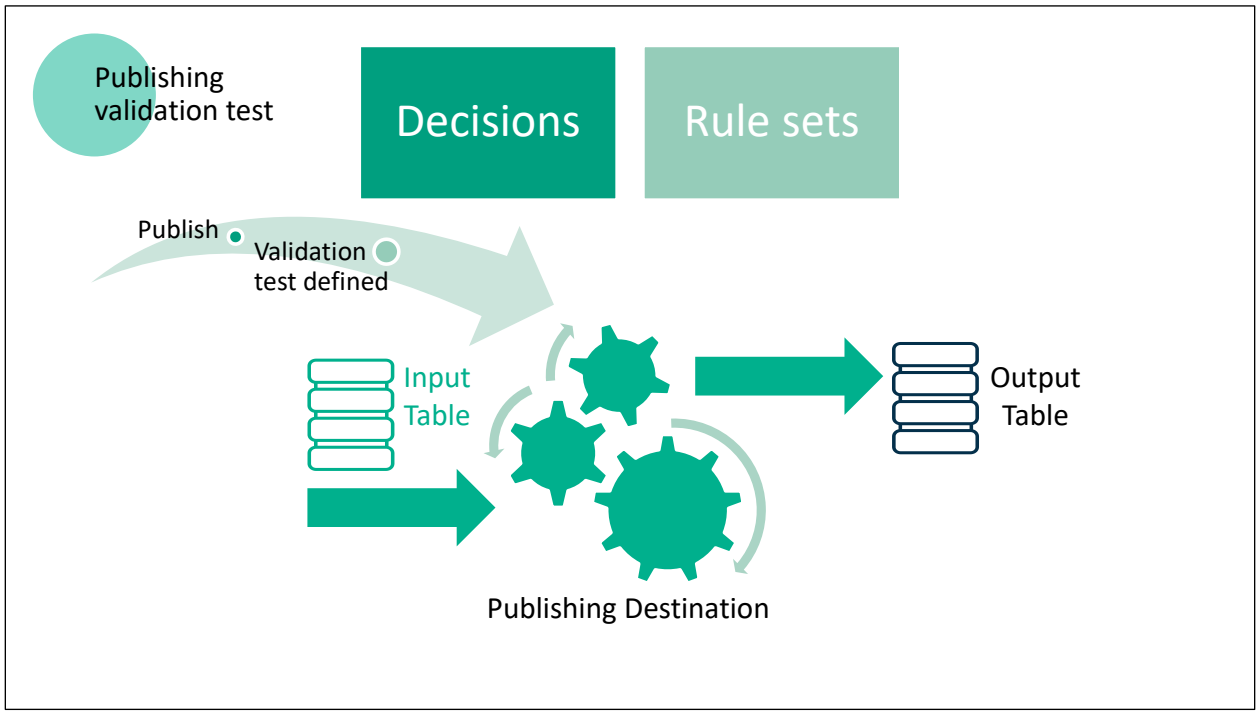
If you are publishing to a destination other than the SAS Micro Analytic Service, you can enable rule-fired tracking and (for decisions only) path tracking.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



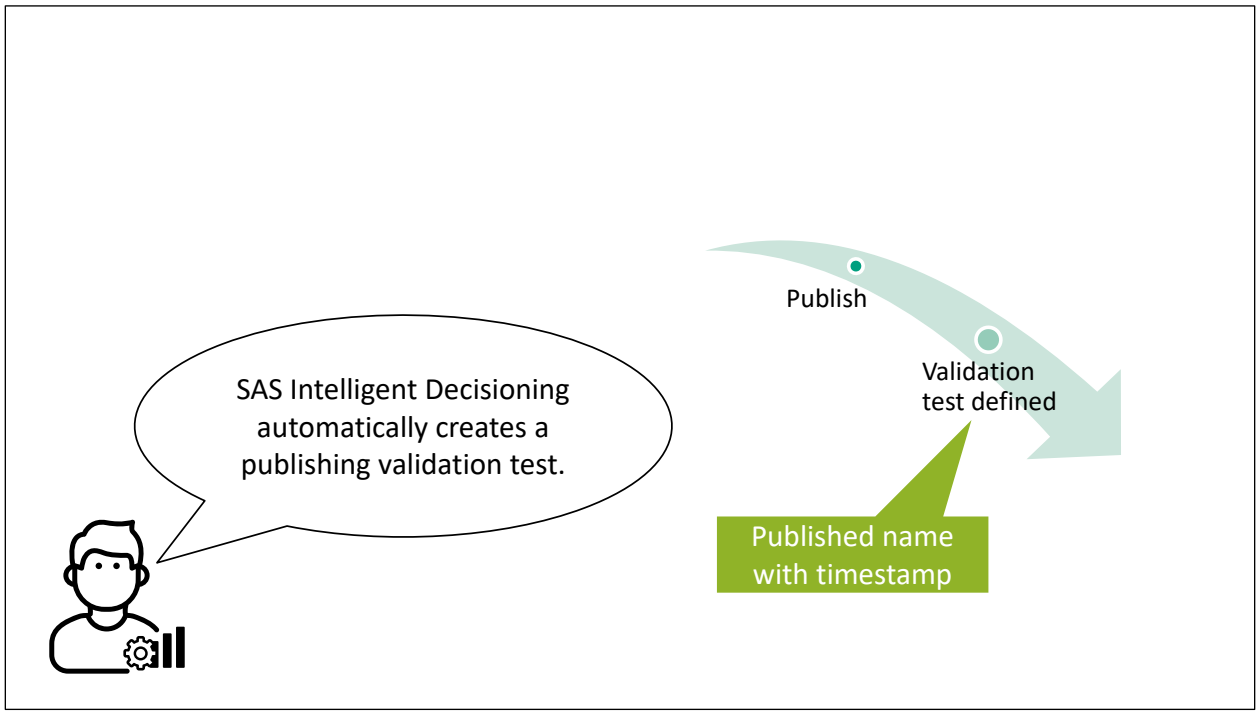
If you are publishing to CAS, you must reload the CAS destination table in order to make the newly published item available to other applications. You select whether to reload the table automatically or manually. Refer to *SAS Intelligent Decisioning: User's Guide* for more information.

Validating a Published Decision

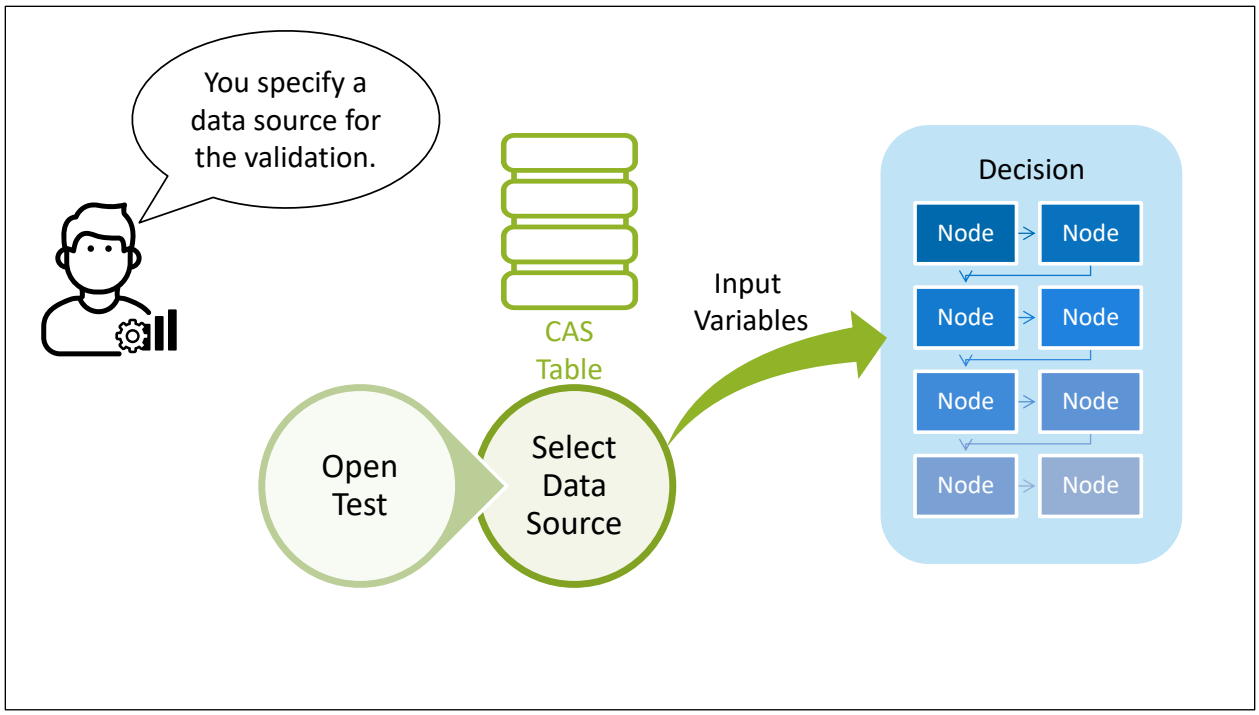


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

You can execute publishing validation tests for decisions and rule sets. When you publish a decision or rule set, a validation test is automatically defined in the selected publishing destination. The test executes the decision or rule set in the publishing destination using the input data that you specify.

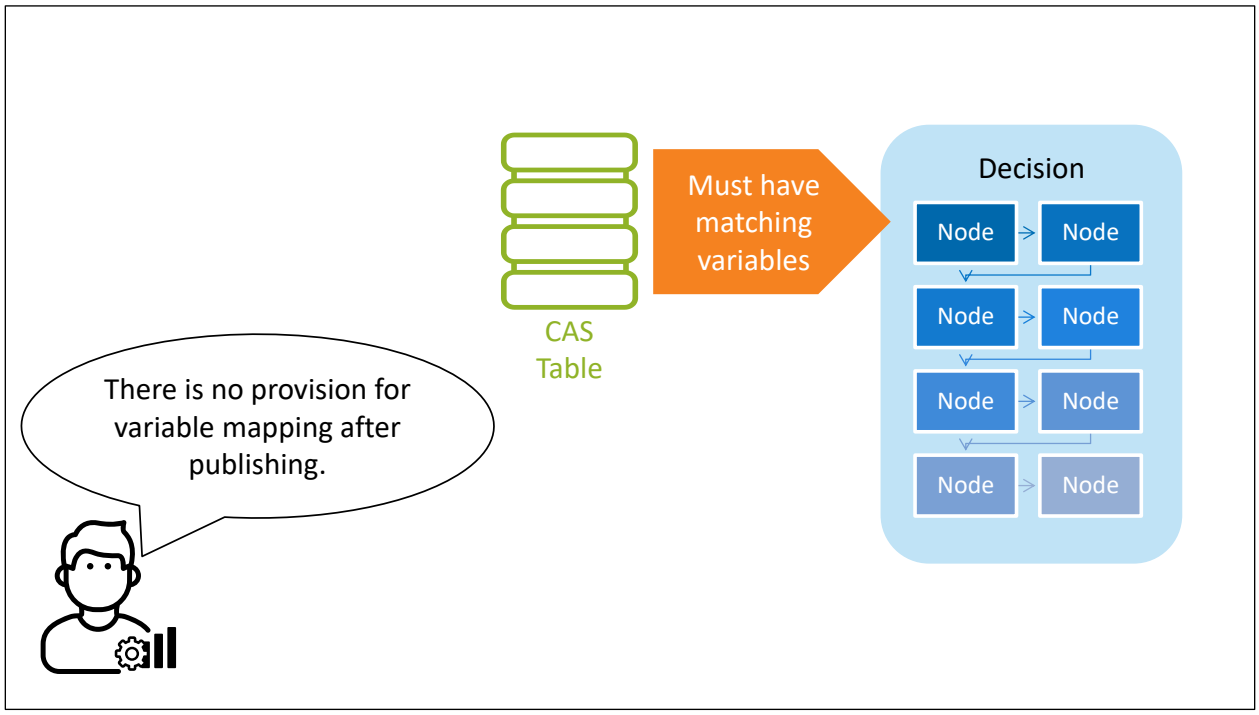


The name of the test is the name of the published decision or rule set with an appended timestamp indicating when the object was published.

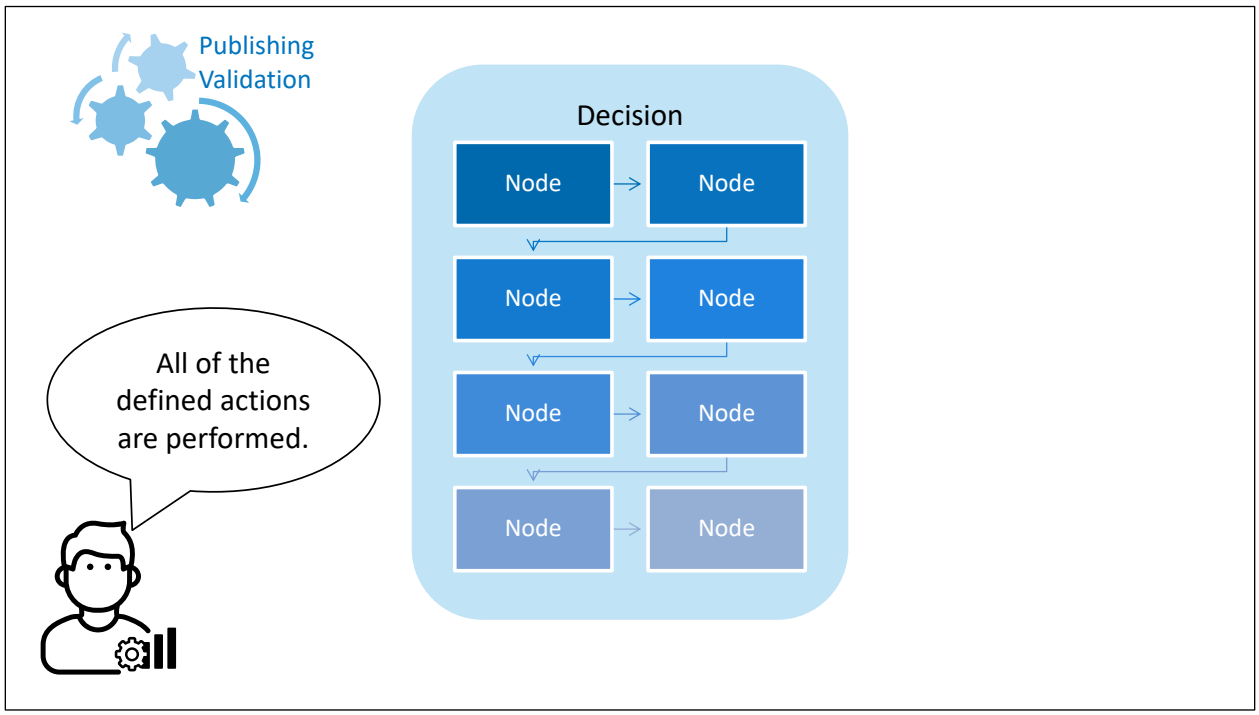


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

After you open the test, you must specify a data source to provide input variable values for the test. The data source must be a CAS table that is loaded in memory. SAS Intelligent Decisioning directs the data to the appropriate destination for validation.

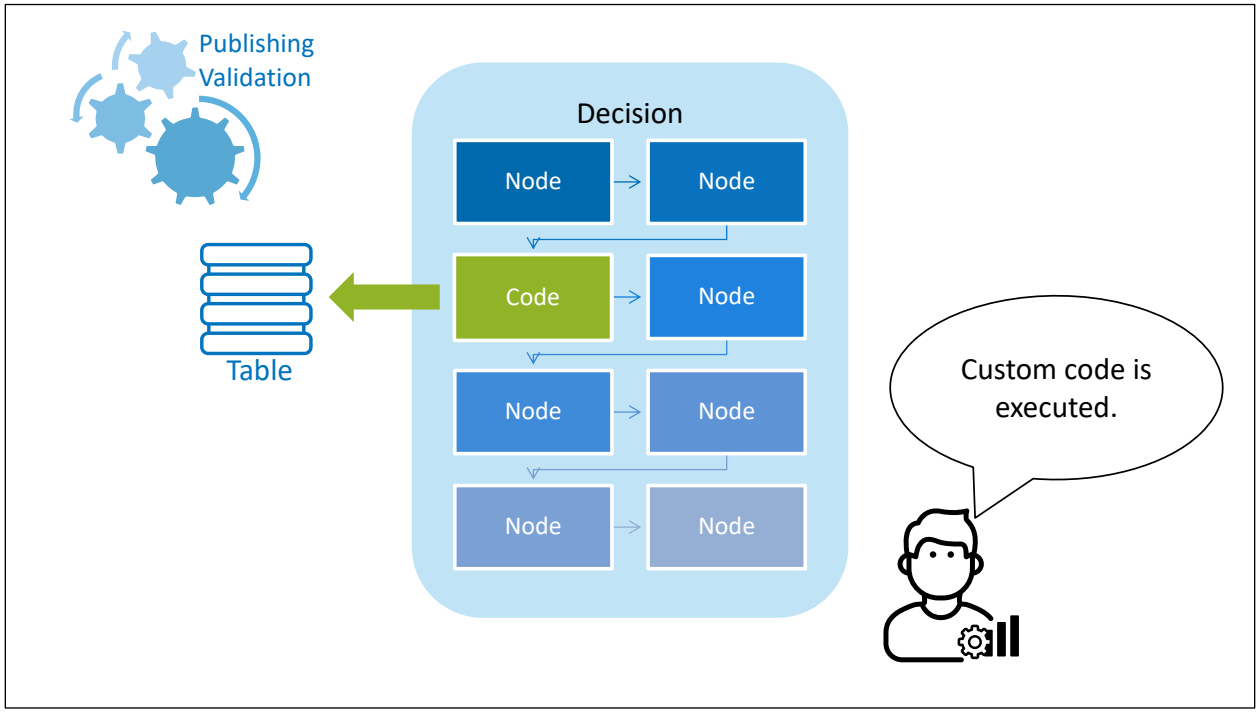


In order for the validation to be successful, the data source must contain variables matching the name and type of the input variables for the decision or rule set. There is no provision for variable mapping after publishing.

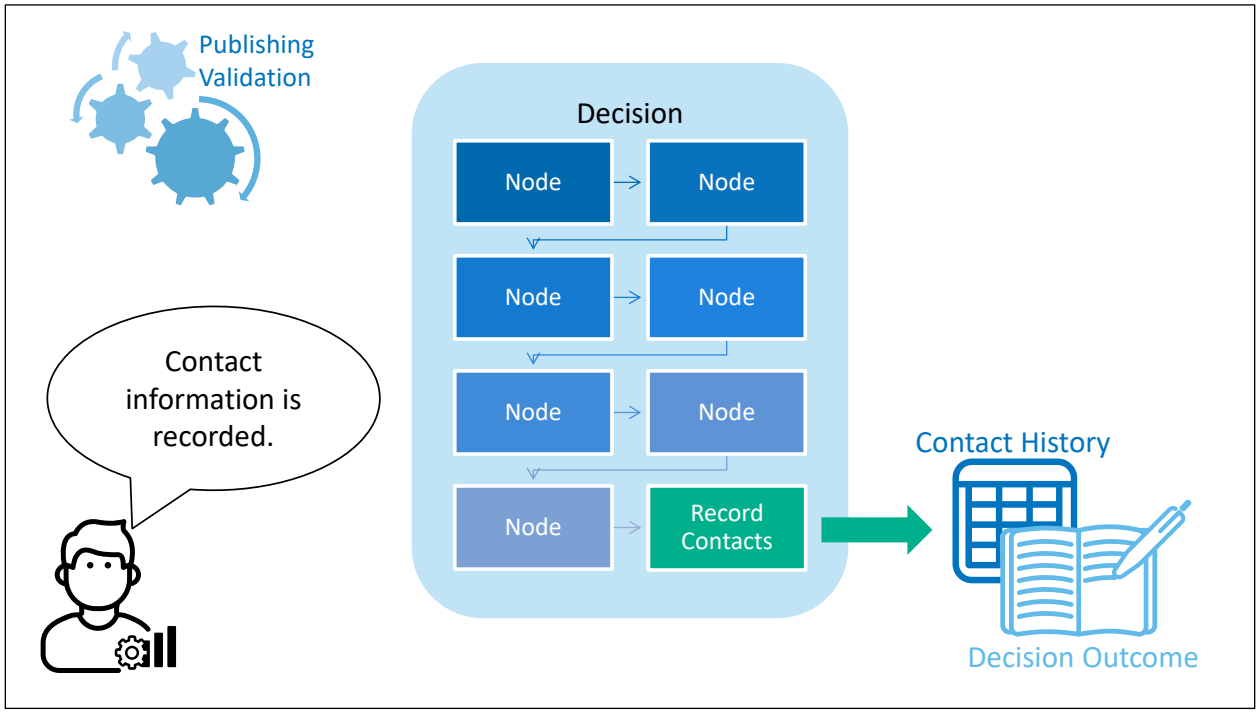


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

During publishing validation, all of the actions defined for a decision are performed.

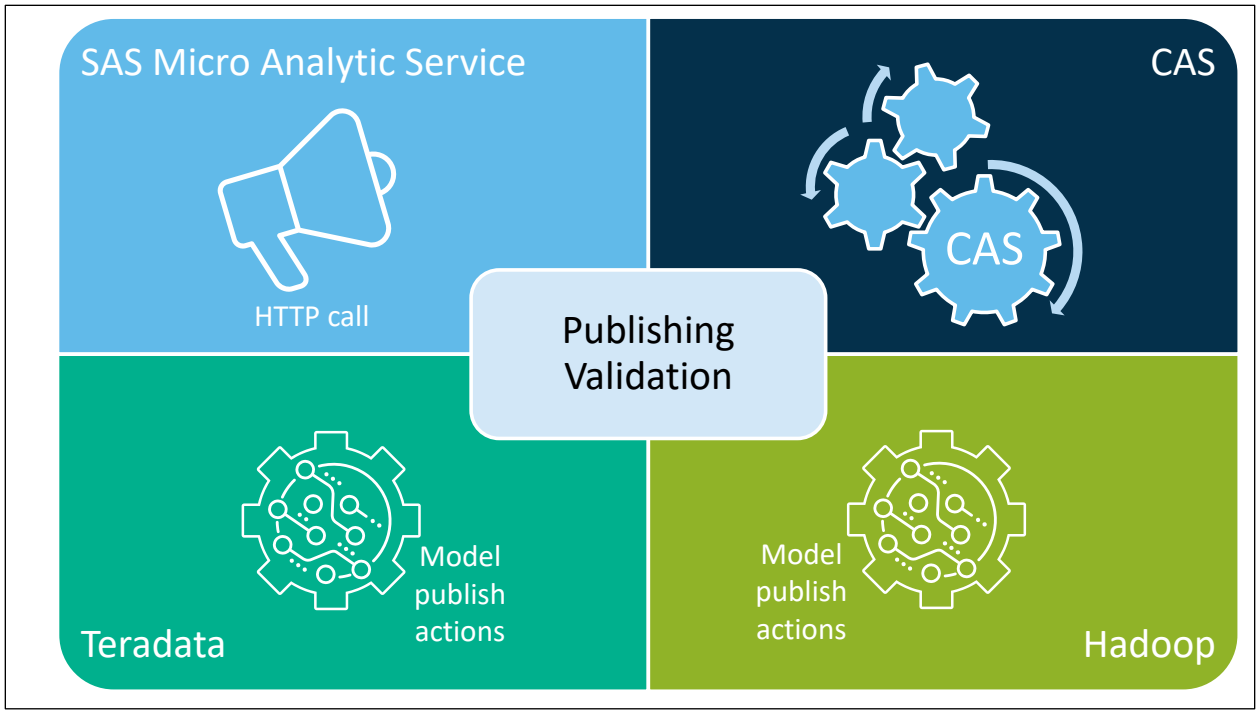


If the decision includes custom code that writes values to a database table, the corresponding values are written.

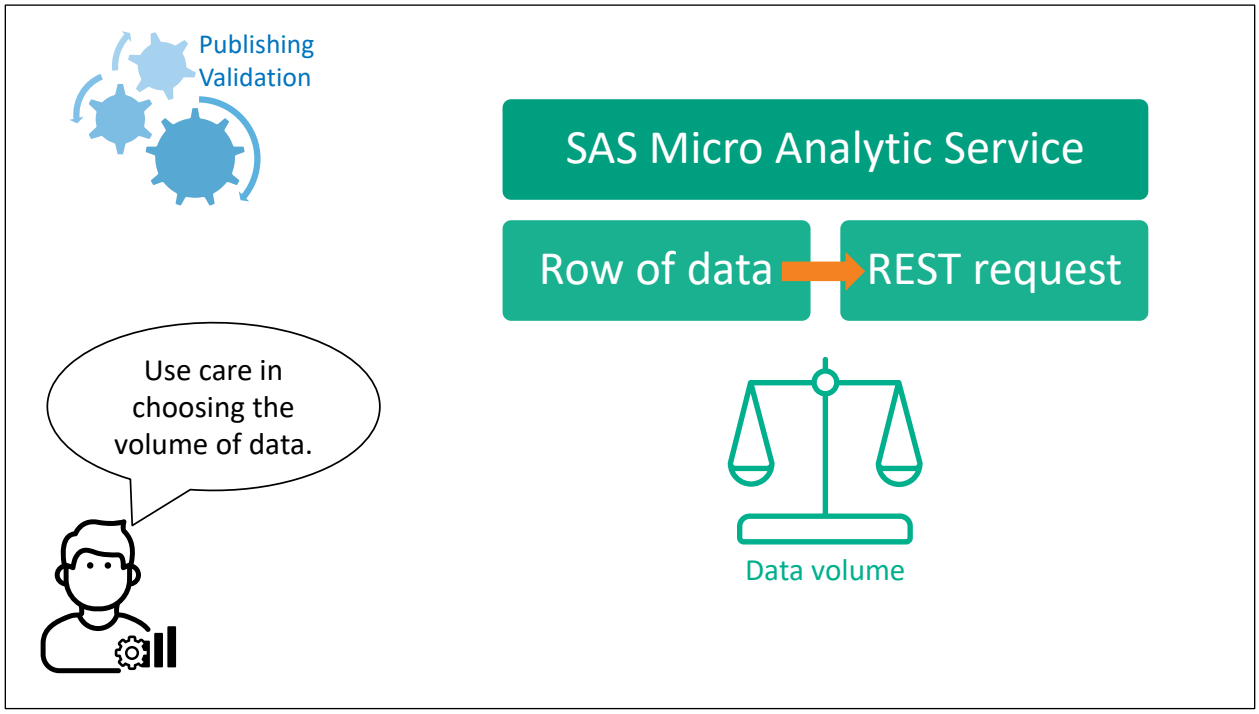


If the decision includes record contacts nodes, then contact information corresponding to the validation is recorded.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

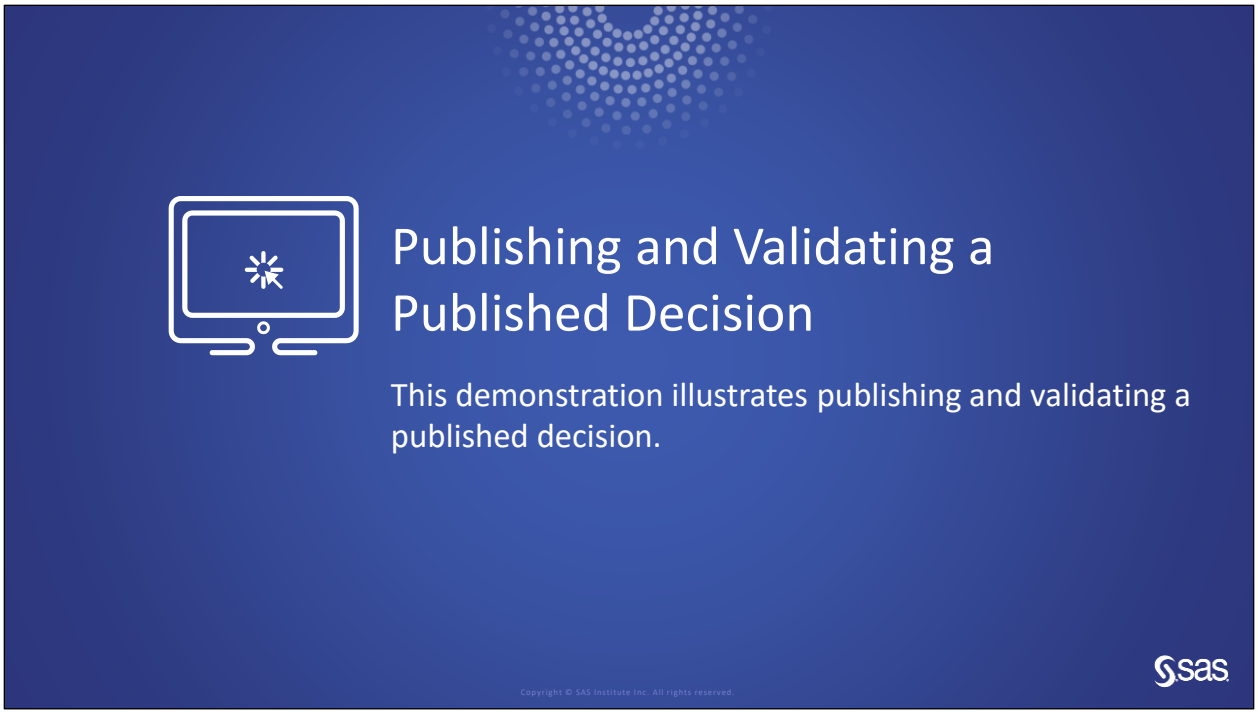


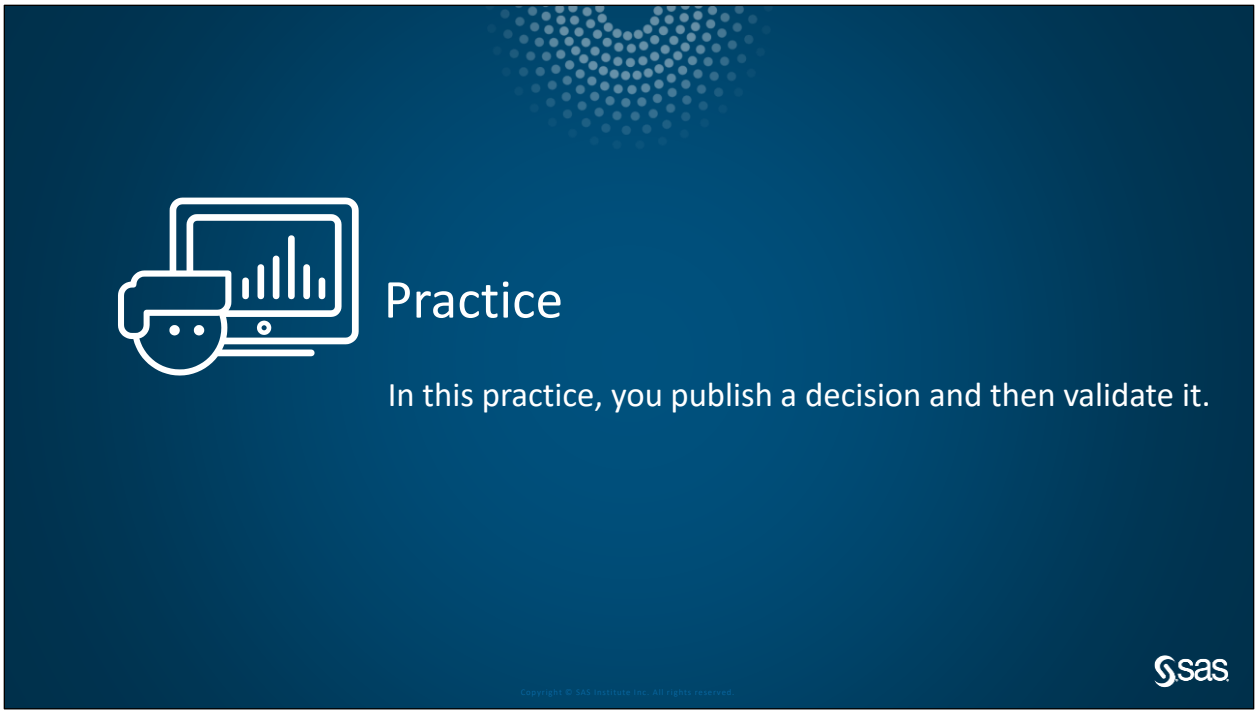
When you execute a publishing validation test for the SAS Micro Analytic Service, the test produces an HTTP call for each row in the input data. When you execute for CAS, the test executes in CAS. When you test for Hadoop or Teradata, the validation executes inside Hadoop or Teradata respectively by calling model publish actions in CAS.



You should use care in choosing the volume of data to use for publishing validation. Specifically for the SAS Micro Analytic Service, a REST request is produced for each row of data.

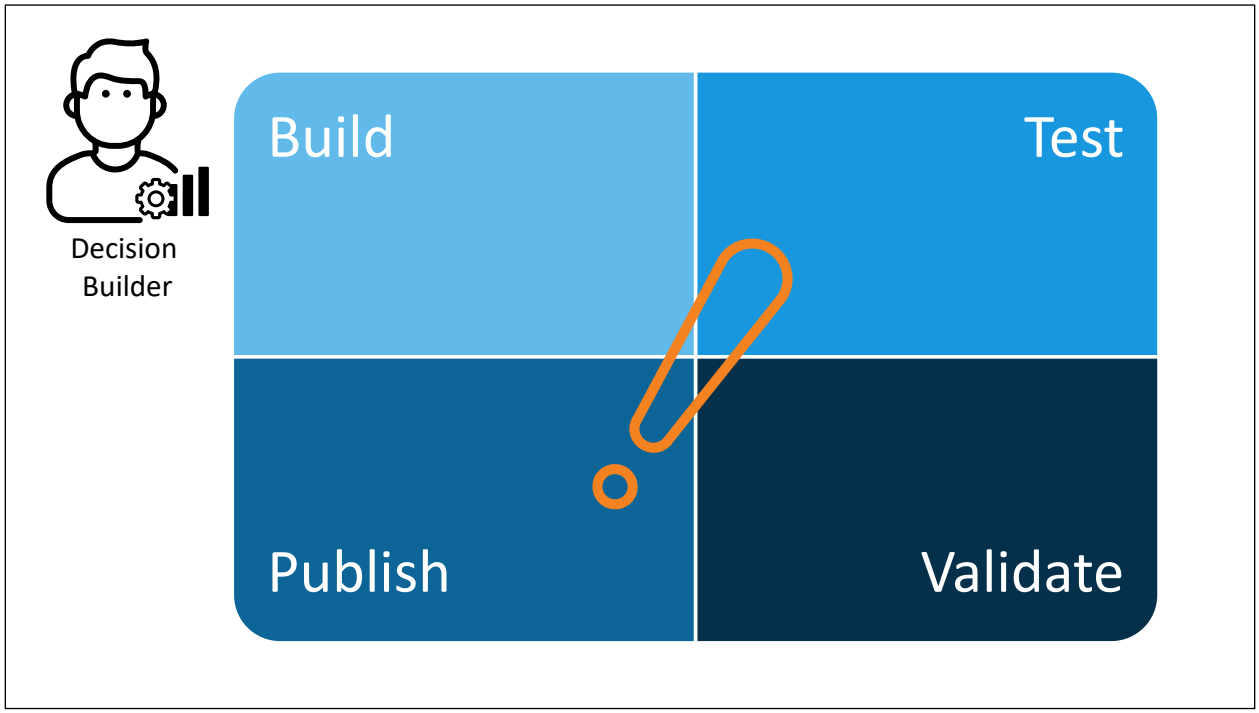
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



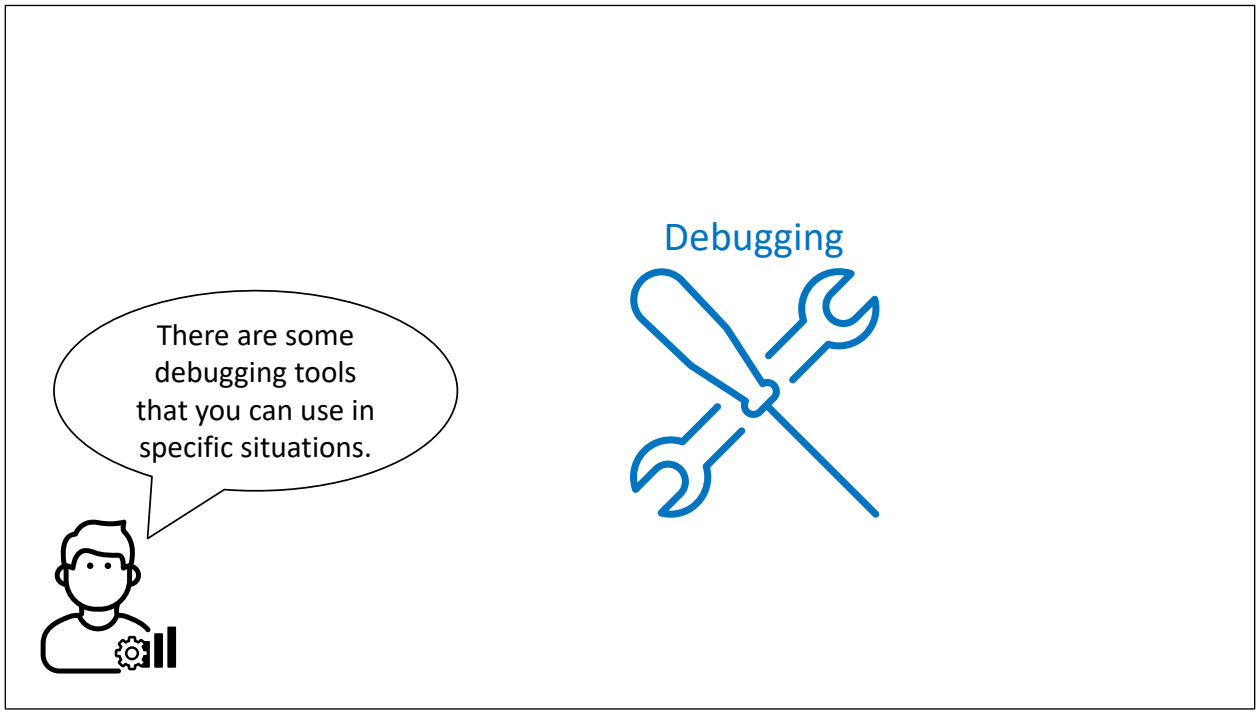


Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

5.3 Troubleshooting

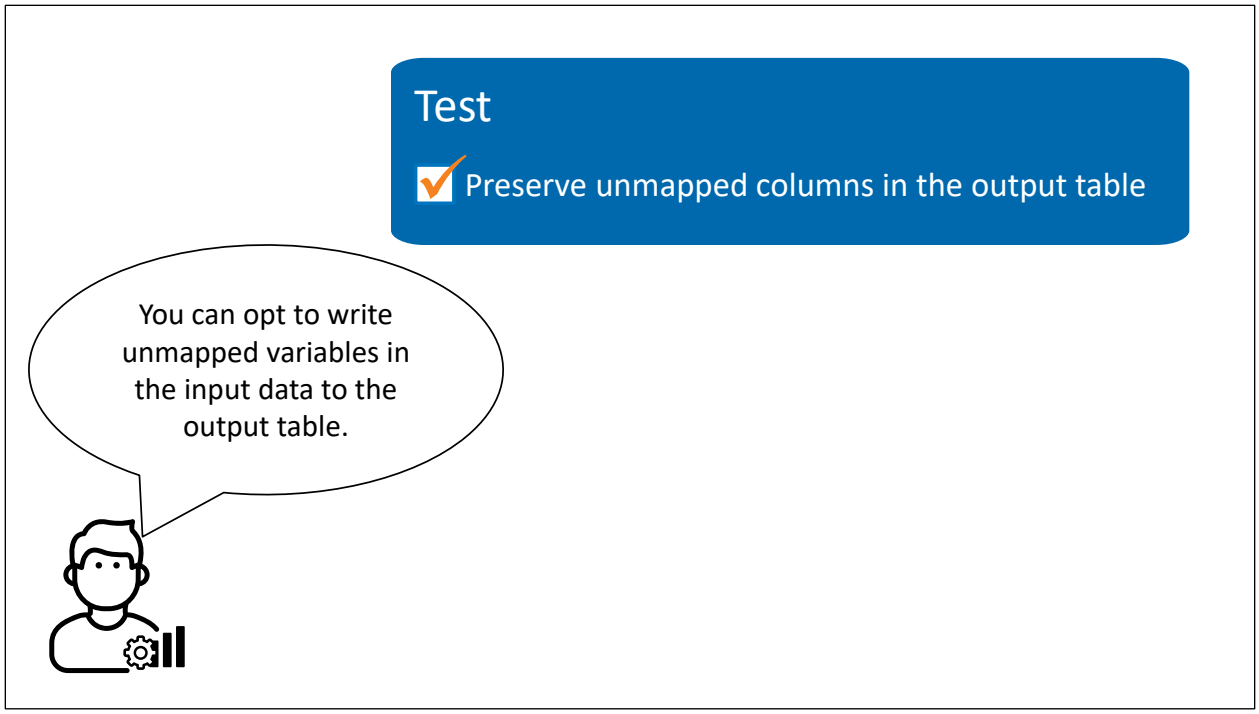


If you encounter issues when developing a decision, there are some resources and techniques that you can use for troubleshooting.

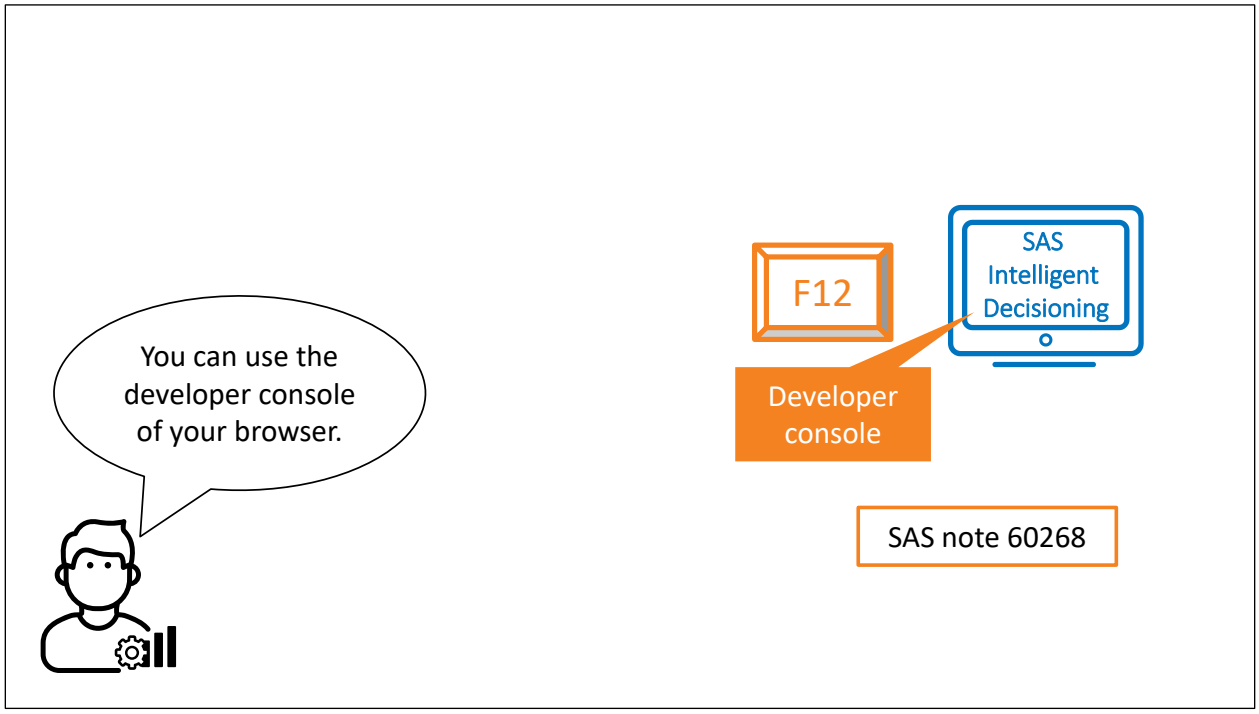


There are some debugging tools that you can use in specific situations.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

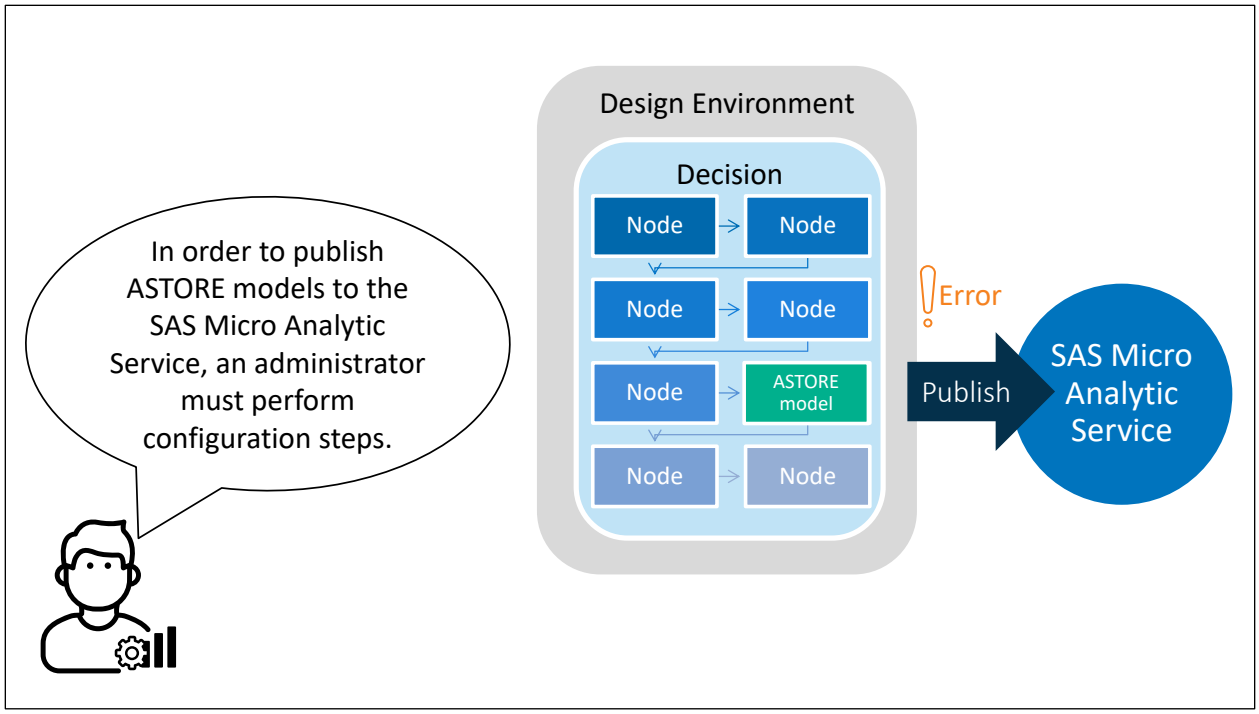


Recall that you can enable an option to preserve unmapped columns in the output table if you want columns in the input test data that are not mapped to an output variable to be written to output. You might want to enable this option when debugging.

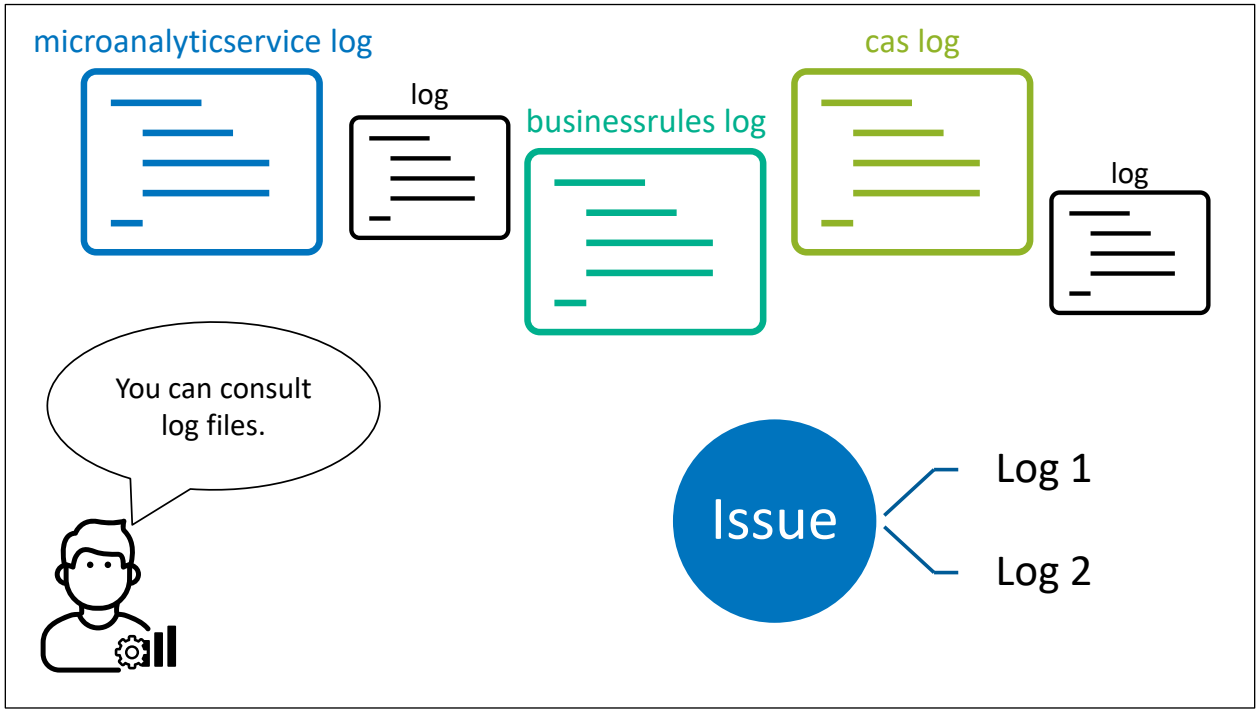


You can also use the developer console in the browser where you are logged on to SAS Intelligent Decisioning to help with debugging. In many browsers, you can use the F12 key to open the console. You can use the Network tab to see all of the network traffic. You can search for SAS note 60268 to learn more.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

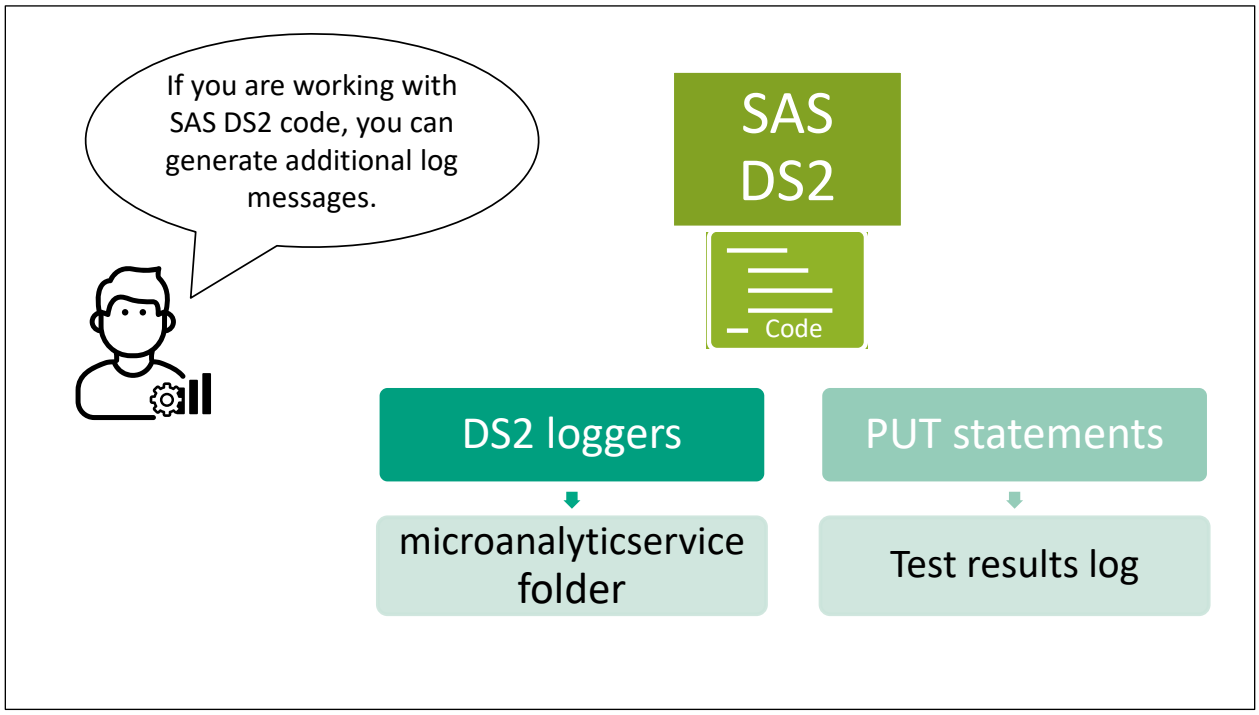


You might experience an error when publishing decisions that use analytic store (or ASTORE) models to the SAS Micro Analytic Service destination. In order for you to publish decisions using ASTORE models to this destination, an administrator must configure access to the location where the ASTORE files are located. Also, users who work with analytic store models must have Read and Write access to analytic store directories. For more information, refer to *SAS Intelligent Decisioning: Users Guide* and *SAS Model Manager: Administration*.



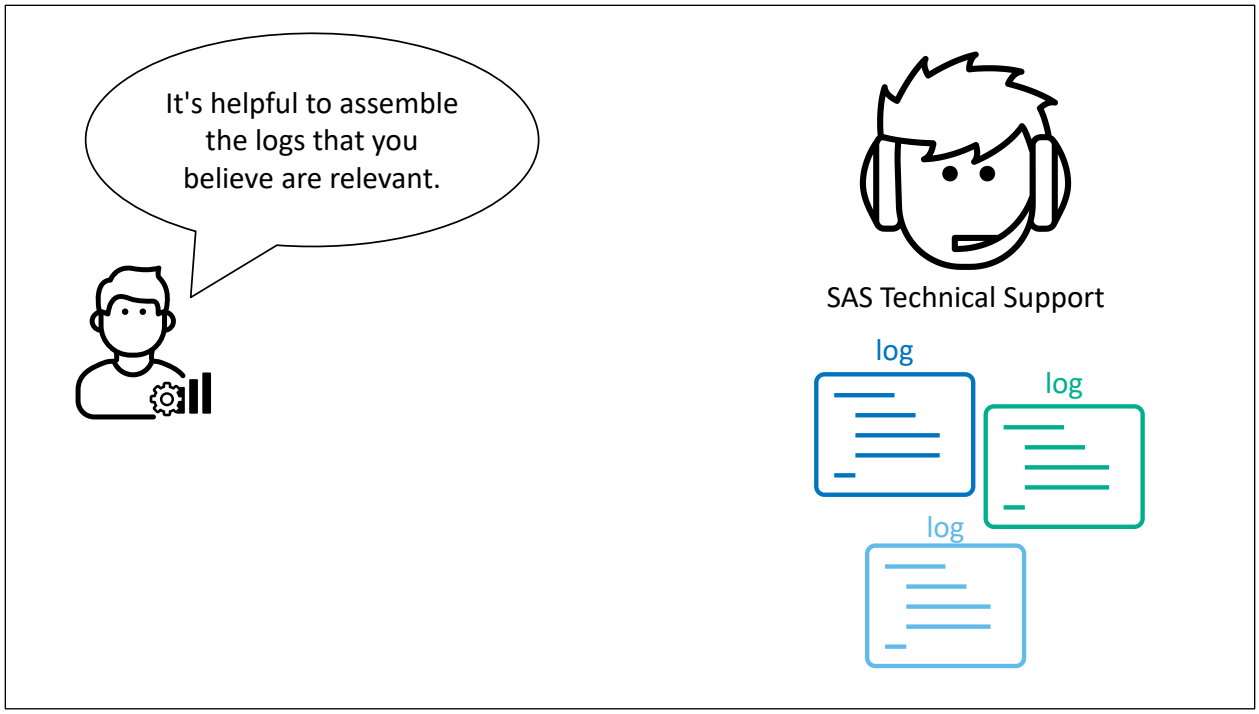
Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

You can consult log files generated by SAS Intelligent Decisioning and SAS Viya to determine the root cause of an issue. The specific log files that might be relevant depend on the issue that you are experiencing.



If you are working with SAS DS2 code, you can generate additional log messages using DS2 loggers or PUT statements. If you use DS2 loggers, the messages are written to logs in the microanalyticservice folder. If you use PUT statements, the messages are written to the log that appears with test results.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



If you ultimately find that you need to contact SAS Technical Support, it's helpful if you have assembled the logs that you believe might be relevant.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Copyright © 2021, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.