



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Genesys Rules Authoring Tool Help

## Using Split Testing to Compare Business Rule Outcomes

5/2/2025

# Using Split Testing to Compare Business Rule Outcomes

## Important

Certain features on which Split Testing depends were not implemented in the initial 9.0.000.13 release of GRAT. This functionality is implemented in release 9.0.000.17.

## Overview

### What problem does A/B/C Split Testing solve?

If your organization has a complex set of rules to define business decision-making, and you want to be able to compare alternative outcomes and scenarios, then rather than just changing a condition and hoping for the best you probably want to phase changes in slowly in order to measure the impact over time.

A/B/C Split Testing (hereafter, **Split Testing**) allows you to compare the business outcomes of alternative rule scenarios before rolling out significant changes to the way you make your business decisions. With Split Testing you can make, test and review changes incrementally to test their effects before committing to a particular change set.

For example, you might want to test several subtle changes in the way applications for credit cards are treated, in order to identify which has the best business outcome. You could split test for one income level against another, or one age range against another, or between different customer segments, or indeed any conditions in your rule package.

### Flexible Approaches to Split Testing

Split Testing offers several approaches. For example, you can choose to:

- Unconditionally force either path A or B or C.
- Tie Split Testing to certain conditions (whether the customer is from a particular city, of a certain age, not a Gold customer, and so on).
- Weight the paths for A, B and C by percentage. This allows you to take a more statistical approach and use larger test data sets.
- Tie Split Testing to a business calendar (for example only do split testing on Fridays after 5 PM or Saturday morning 9 AM - 12 PM).

## Split Testing Template

The GRSSplitTest\_template.xml template provides some basic Facts (Conditions and Actions) for the Split Testing feature. To implement the feature, GRAT users must import this template (from the **Samples** folder) and attach it to all rules packages for which the Split Test functionality is required.

### Important

The template—GRSSplitTest\_template.xml—is shipped with type samples. To use this template with other rule package types, you can import the template into GRAT, change the name (for example, GRSSplitTestForMyType) and the type (to match your rule package type) and publish. Then you can use it with another package type.

## Split Test Column in Rule Packages

A Split Test column displays at the rule package level.

General		Rules	Audit Trail	Package History	
New Decision Table		New Linear Rule	Import Rule		
ID	Name	Description	Phase	Split Test	
Rule-100	by default assign to Sales D		classification	*	

When the Split Test template has been imported into GRAT, this drop-down field will show the values imported from the template (A, B and C are the default values in the shipped template). If the template has not been imported, only the wildcard (\*) symbol will be available.

Duplicate (copy and paste) the rule change that you wish to test. On the duplicated rule, make your logic change (for example, adjust the income requirement needed to offer a credit card).

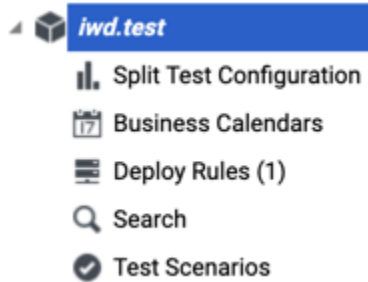
Mark the original rule as “A” and the new rule as “B” by selecting from the “Split Test” column.

Once “A”, or “B” is selected, the rows color-code for easier viewing. The wildcard symbol (\*) means that the Split Test condition is ignored, so this rule will always fire.

The rules marked as “A” or “B” (or “C”) will fire according to the logic in the Split Test Configuration. For example, you can weight the distribution such that “A” fires 75% of the time and “B” fires 25% of the time, etc. See the next section on Split Test Configuration for more options.

## Split Test Configuration Node

A Split Test Configuration node appears in the left navigation tree. You can use this node to create (and display existing) Split Test rules that will apply at the rule package level. In rules created under this node, the **Split Test** column is hidden to avoid confusion.



## What Can I Do with Split Test Rules?

### Unconditionally force either path A or B or C

In rules created under this node, the `splitTestValue` can be selected as a Parameter for the rule. You can use this simple rule to force the Split Test path to any of the values in the template (the default shipped values in the template are A, B and C, but you can change these in the template if required).

Rules

Audit Trail


New Decision Table

New Linear Rule

Import Rule

More

▼

ID	Name	Description	Phase	Calendar
 Rule-100	Set Split Path Default		*	
Rule-102	Distribute Split Test	25% A, 25% B, 50% C	*	

Add Condition

▼

Add Action

▼

Group

▼

Section	Expression	Parameters			
When					
Then					
	Set Split Path to	A			

## Tie Split Testing to a Condition

To add more flexibility and precision to your use of Split Test rules, you can add to your rule any other condition available from the template(s) attached to the rule package. The example below has lower and upper age limits:

Add Condition

▼

Add Action

▼

Group

▼

Section	Expression	Parameters			
When					
	Age is between	21	and	40	
Then					
	Set Split Path to	A			

### Weight the paths for A, B and C by percentage

You could add a condition to perform Weighted Distribution Split Testing. In this example the Split Test rule distributes the incoming rule evaluation requests across the A, B and C paths in the percentage proportions 25/25/50. This should look something like this:

Add Condition

▼

Add Action

▼

Group

▼

Section	Expression	Parameters					
When							
Then							
	Split weight distribution:	25	%A	25	%B	50	%C

### Tie Split Testing to a Business Calendar

You can also tie Split Testing to a Business Calendar. For example, you might prefer to keep Split Testing outside normal business hours, or maybe run Split Testing in response to specific customer offers or other commercial events. To do this you can either use an existing Business Calendar or create a new one for your specific purpose. You then add this calendar to the rule to which you want to apply Split Testing.

So, when the parameters set for the Business Calendar apply, the weighted Split Testing begins and ends automatically.

The screenshot displays the 'Rules' tab in the Genesys Rules Authoring Tool. At the top, there are buttons for 'New Decision Table', 'New Linear Rule', 'Import Rule', and 'More'. Below these is a table listing rules. The table has columns: ID, Name, Description, Phase, Calendar, and Pending Snapshot. The rule 'Rule-102' is highlighted, with the name 'Distribute Split Test', description '25% A, 25% B, 50% C', and phase '\*'. The calendar is set to 'WeekendSplitTest'. Below the table, there are buttons for 'Add Condition', 'Add Action', and 'Group'. The 'When' section is expanded, showing a condition 'Today is not a working day'. The 'Then' section is also expanded, showing a split weight distribution: 25% for %A, 25% for %B, and 50% for %C. A curved arrow points from the 'WeekendSplitTest' calendar field in the rule table to the 'Today is not a working day' condition in the 'When' section.

ID	Name	Description	Phase	Calendar	Pending Snapshot
Rule-100	Set Split Path Default		*		✓
Rule-102	Distribute Split Test	25% A, 25% B, 50% C	*	WeekendSplitTest	✓

Section	Expression	Parameters
When	Today is not a working day	
Then	Split weight distribution:	25 %A 25 %B 50 %C

## Using Test Scenarios to Check Your Split Test Logic

GRAT's Test Scenario feature allows you to run pre-deployment checks on the logic of any rule package that you want to deploy. Because A/B/C Split Testing enables an additional level of logic to be built into a rule package, it's advisable to run more complex Split Test logic through a Test Scenario before deploying it to a production rules engine.

### Example

1. You have a simple Call Treatment rule which states that only Gold customers will be offered a callback.

General		Rules		Audit Trail		Package History	
New Decision Table		New Linear Rule		Import Rule		More <span>▼</span>	
ID	Name	Description	Phase	Split Test	Calendar	Pending Snapshot	
Rule-100	Gold		*	*			

Add Condition <span>▼</span>		Add Action <span>▼</span>		Group <span>▼</span>	
Section	Expression	Parameters			
When					
	Customer segment is	Gold			
Then					
	Offer call back	<input checked="" type="checkbox"/>			

2. You want to test what would happen if you started offering callback to Gold and Silver customers. You decide you want to test this change by integrating it in for only 50% of the calls. So, you make the standard rule split test **A**:



General		Rules		Audit Trail		Package History	
New Decision Table		New Linear Rule		Import Rule		More ▾	
ID	Name	Description	Phase	Split Test	Calendar	Pending Snapshot	
Rule-100	Gold		*	A			

Add Condition ▾		Add Action ▾		Group ▾			
Section	Expression	Parameters					
When							
	Customer segment is	Gold					
Then							
	Offer call back	<input checked="" type="checkbox"/>					

3. The next step is to copy and paste the rule. Make the second rule split test **B**, then change the rule logic to allow **Gold** or **Silver**:

General		Rules	Audit Trail	Package History				
New Decision Table		New Linear Rule	Import Rule	More	▼			
ID	Name	Description	Phase	Split Test	Calendar	Pending Snapshot		
Rule-100	Gold		*	A		✓		
Rule-113	Gold or Silver		*	B		✓		

Add Condition ▼		Add Action ▼	Group ▼					
Section	Expression	Parameters						
When								
	Customer segment is	Gold						
	or							
	Customer segment is	Silver						
Then								
	Offer call back	✓						

4. Next, create a simple Split Test rule, which will determine when to use rule **A** and when to use rule **B**. In this use case, you want this distributed 50% **A** and 50% **B**. So, select the action Split weight distribution 50% A, 50% B and 0% C. This action is provided in the Split Test example template.

Rules

Audit Trail

New Decision Table

New Linear Rule

Import Rule

More

▼

ID	Name	Description	Phase	Calendar	Pending Snapshot	Start Date
Rule-107	Split Test Rule		*		✓	

Add Condition

▼

Add Action

▼

Group

▼

Section	Expression	Parameters						
When								
Then								
	Split weight distribution:	50	%A	50	%B	0	%C	

5. Now, create a test scenario to test the changes. In this case, if the customer is **Gold**, you expect to always offer a callback. If the customer is **Silver**, you expect to offer a callback 50% of the time:

Test Scenarios

New Test Scenario

Run Test Scenario

Run All

Import Test Scenario

More

ID	Name	Description	Phase	Business Hierarchy	Simulated Date	Simulated Time
TS-103	Test		*	*		

Add Given

Add Expectation

Add Row

ID	Name	Results	pCustomerSegment	pOfferCallBack	Actions
TSR-104			Gold	<input checked="" type="checkbox"/>	<div>+ ↺ −</div>
TSR-105			Silver	<input checked="" type="checkbox"/>	<div>+ ↺ −</div>

6. If you now run the test scenario multiple times, you will see that 50% of the time you get a green light on the Silver (indicating that you offered a callback) and 50% of the time you get a red light (indicating that you did not offer a callback):

### First run

Test Scenarios

New Test Scenario

Run Test Scenario

Run All

Import Test Scenario

More

ID	Name	Description	Phase	Business Hierarchy	Simulated Date	Simulated Time	Time Zone	Result	Actions
TS-103	Test		*	*			(UTC+0) Greenwich Mean Time : Gt	<div>✓</div>	<div>⋮</div>

Add Given

Add Expectation

Add Row

Undo

Redo

ID	Name	Results	pCustomerSegment	pOfferCallBack	Actions
TSR-104		<div>✓</div>	Gold	<input checked="" type="checkbox"/>	<div>+ ↺ −</div>
TSR-105		<div>✓</div>	Silver	<input checked="" type="checkbox"/>	<div>+ ↺ −</div>

### Second run

Test Scenarios									
<div> New Test Scenario Run Test Scenario Run All Import Test Scenario More </div>									
ID	Name	Description	Phase	Business Hierarchy	Simulated Date	Simulated Time	Time Zone	Result	Actions
TS-103	Test		*	*			(UTC+0) Greenwich Mean Time : GA	<span></span>	<span></span>
<div> Add Given Add Expectation Add Row Undo Redo </div>									
ID	Name	Results	pCustomerSegment	pOfferCallback	Actions				
TSR-104		<span></span>	Gold	<span></span>	<span></span> <span></span> <span></span>				
TSR-105		<span></span>	Silver	<span></span>	<span></span> <span></span> <span></span>				

7. After testing that the A/B/C split logic is correct, you can deploy your rule package. At this point, the A/B/C split logic will run as you have it configured during normal rule evaluation. To adjust the A/B/C configuration (for example, make the **A** path 25% instead of 50% and the **B** path 75% instead of 25% and so on), you must make the changes in GRAT and then redeploy the rule package.

## Allowing an Application to Override Split Test Configuration Rules

In addition to setting the A/B/C path in the Split Test Configuration section, it is possible for the invoking application (GVP, IVR, ORS and so on) to set the value. This can either be used as a default, or can be used to override the logic in the Split Test Configuration.

The application can pass in the split test value in the **\_GRS\_Environment** fact (field name: **splitTestPath**).

If you want the invoking application to be able to override the normal logic in Split Test Configuration, then you can simply add the Split Test Path is not set condition to your rule.

Add Condition		Add Action		Group	
Section	Expression	Parameters			
When					
	Split test path is not set				
Then					
	Set Split Path to	A			

This condition will check whether the value has already been passed in by the invoking application. If it has been passed in, then the Split Test logic will be bypassed. If not, then the normal Split Test rule logic will apply.