



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

12/19/2025

Using Split Testing to Compare Business Rule Outcomes

Overview

[+] DETAILS

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

If your organization has a complex set of rule packages and rules to define business decision-making, if 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).

New Split Testing Template

A new template ships with GRS 8.5.3—GRSSplitTest_template.xml—and this 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

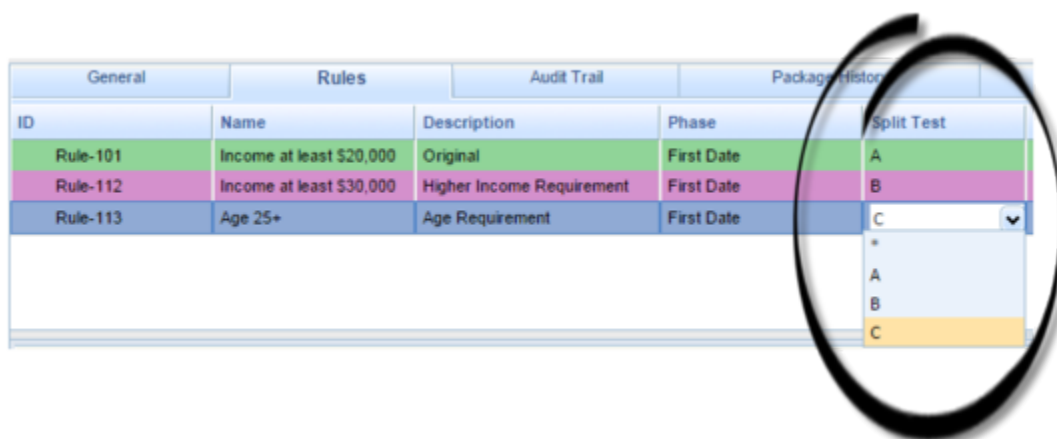
The new template—GRSSplitTest_template.xml—is shipped with type samples. This means it can only be added to rule packages of type samples, because GRAT only shows templates in the list with compatible types. To use this template with other rule package types, you can import the template from GRAT into GRDT, change the name (for example, GRSSplitTestForMyType) and the type (to match your rule package type) and publish to GRAT. Then you can use with another package type.

Changes in GRAT

[+] DETAILS

New Split Test Column in Rule Packages

A new Split Test column displays at the rule package level.



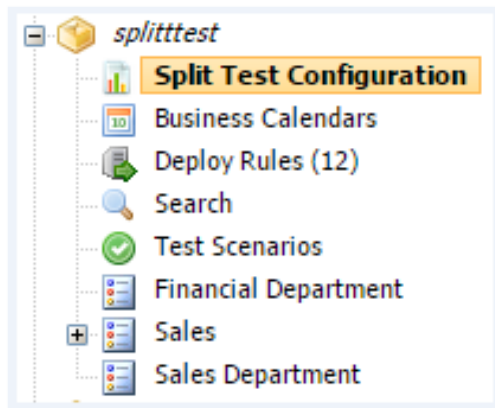
General	Rules	Audit Trail	Package History	Split Test
ID	Name	Description	Phase	
Rule-101	Income at least \$20,000	Original	First Date	A
Rule-112	Income at least \$30,000	Higher Income Requirement	First Date	B
Rule-113	Age 25+	Age Requirement	First Date	C

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.

When a path is selected in this column, the rows color-code for easier viewing. The wildcard symbol (*) means that the Split Test condition is ignored, so this rule will always fire.

New Split Test Configuration Node

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

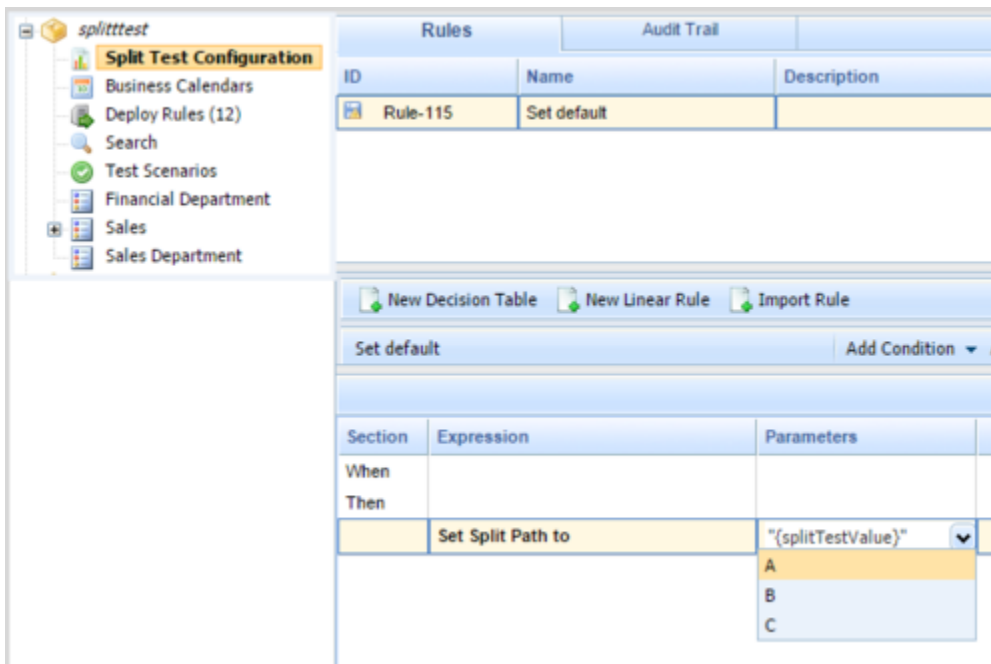


What Can I Do with Split Test Rules?

[+] Force the Split Test Path to A, B or C

Unconditionally force either path A or B or C

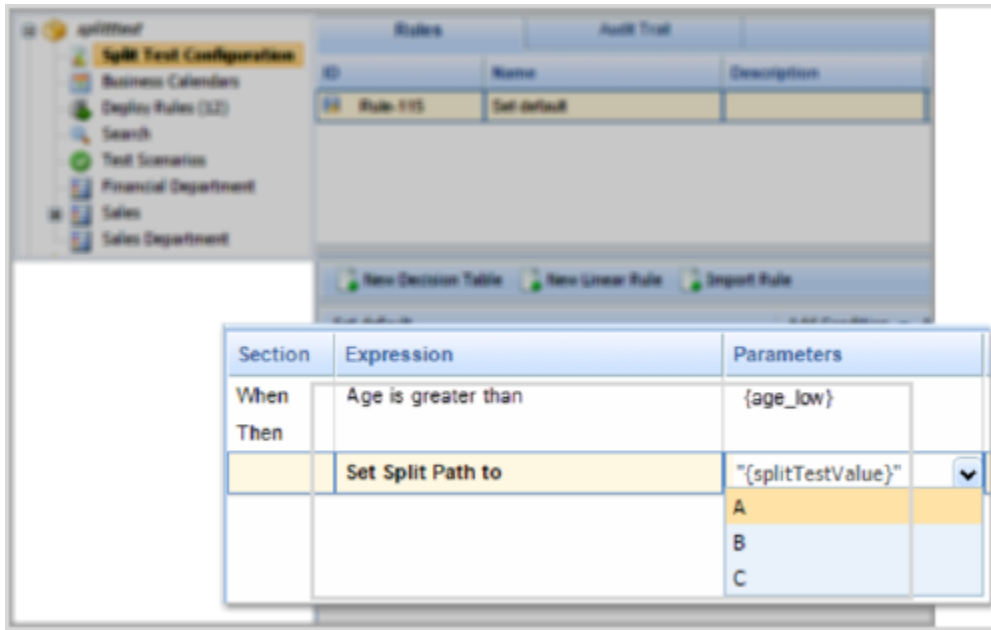
In rules created by clicking this new node, the `splitTestValue` can be selected as a Parameter for the rule. You can use this simple rule to force the Split Test path to any of the values in the template (the default shipped values in the template are A, B and C, but you can change these in the template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



[+] Tie a Split Test Rule to a Condition

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 a lower age limit:



[+] Set Up a Weighted Distribution

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 50/25/25. This should look something like this:

The screenshot shows the 'Rules' configuration window. On the left, a tree view shows 'splitttest' expanded, with 'Split Test Configuration' selected. The main area shows a table of rules. Rule-117, 'Weighted percentages', is selected. Below the table, the 'Parameters' section is visible, showing a 'Split vweight distribution' with three values: 50, 25, and 25. A handwritten note 'weighted split test in these proportions' is written over the parameters, with arrows pointing to the 50, 25, and 25 values.

ID	Name	Description	Phase	Calendar	Pending Snapshot	Start Date	End Date
Rule-117	Weighted percentages		*		☒		

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

[+] Tie a Split Test Rule to a Business Calendar

Tie Split Testing to a Business Calendar

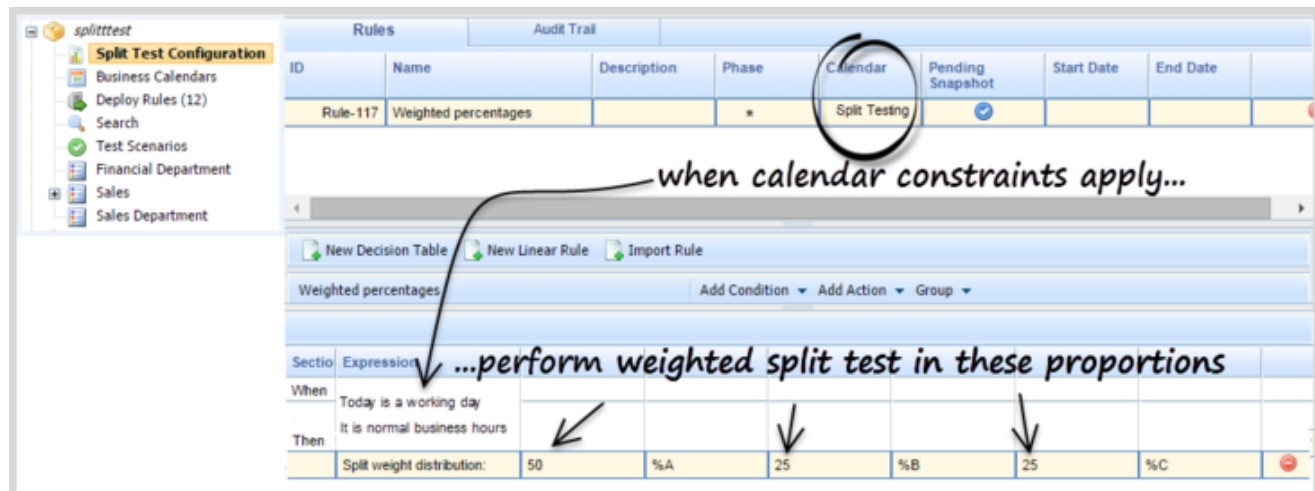
You can also tie Split Testing to a Business Calendar. For example, you might prefer to keep Split Testing outside normal business hours, or maybe run Split Testing in response to specific customer offers or other commercial events. To do this you can either use an existing Business Calendar or create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

The screenshot shows two windows. The top window, 'Business Calendars', lists a calendar named 'Split Testing' with ID 'Calendar_140'. The bottom window, 'Rules', shows the same rule 'Rule-117' with 'Weighted percentages'. The 'Calendar' dropdown menu is open, showing 'Split Testing' as the selected option. An arrow points from the 'Split Testing' calendar in the top window to the 'Split Testing' option in the dropdown menu of the bottom window.

ID	Name	Week Starts on	Week Ends on	Start Time	End Time	Time Zone
Calendar_140	Split Testing	Saturday	Saturday	8:00 AM	12:00 PM	(UTC-4) Eastern Standard T

ID	Name	Description	Phase	Calendar	Per Sn
Rule-117	Weighted percentages		First Date	Split Testin	

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



Using Test Scenarios to Check Your Split Test Logic

[+] 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.

For example, if you wanted to check the Weighted Distribution example above before deploying the rule package to a production engine, you could create a test scenario to have 10 rows of identical data that only qualify when running path **A**. Each row in the test scenario simulates a separate rule evaluation and will therefore apply the Split Test Configuration rules first. If we see a green tick, then the original **A** path was taken. If we see a red cross, then either the **B** or **C** path was taken.

Because we configured the **A** path for 50% of the time, we should see about 50% green ticks every time we hit the **Run Test Scenario** button. Here are the outcomes of running the test scenario twice:

ID	Name	Results	Income	age
TSR-127		✓	20000	20
TSR-139		✓	20000	20
TSR-138		✓	20000	20
TSR-137		✓	20000	20
TSR-136		✗	20000	20
TSR-135		✗	20000	20
TSR-134		✗	20000	20
TSR-133		✓	20000	20
TSR-132		✗	20000	20
TSR-131		✗	20000	20

ID	Name	Results	Income	age
TSR-127		✗	20000	20
TSR-139		✗	20000	20
TSR-138		✓	20000	20
TSR-137		✓	20000	20
TSR-136		✗	20000	20
TSR-135		✓	20000	20
TSR-134		✓	20000	20
TSR-133		✗	20000	20
TSR-132		✓	20000	20
TSR-131		✓	20000	20

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 50% 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

[+] 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.

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

Section	Expression	Parameters
When	Split test path is not set	
Then	Set Split Path to	A

This new 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.