Genesys Rules System Overview

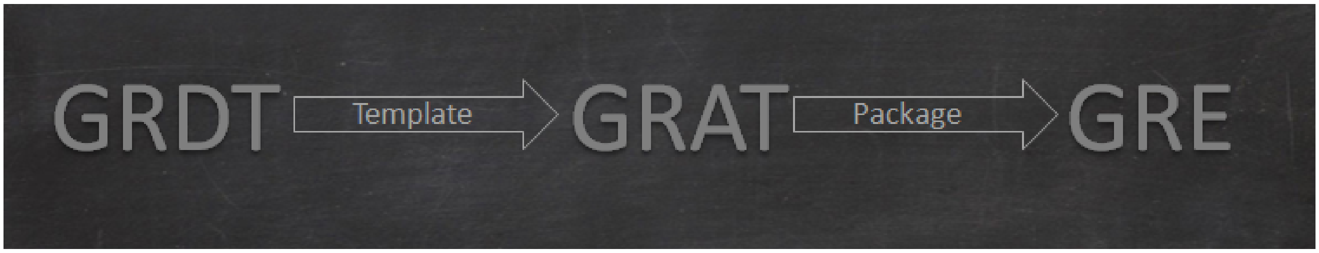
Genesys Rules System 8.5.3

12/29/2021

Table of Contents

GRS Overview

These pages give a high-level overview of how the Genesys Rules System operates within the Genesys eco-system. Start here for an introduction to basic components, elements and architectures of the GRS functionality.



GRS provides the ability to develop, author, and evaluate business rules. A business rule is a piece of logic defined by a business analyst. These rules are evaluated in a Rules Engine based upon requests received from client applications such as iWD, Conversation Manager and Genesys Web Engagement.

Components

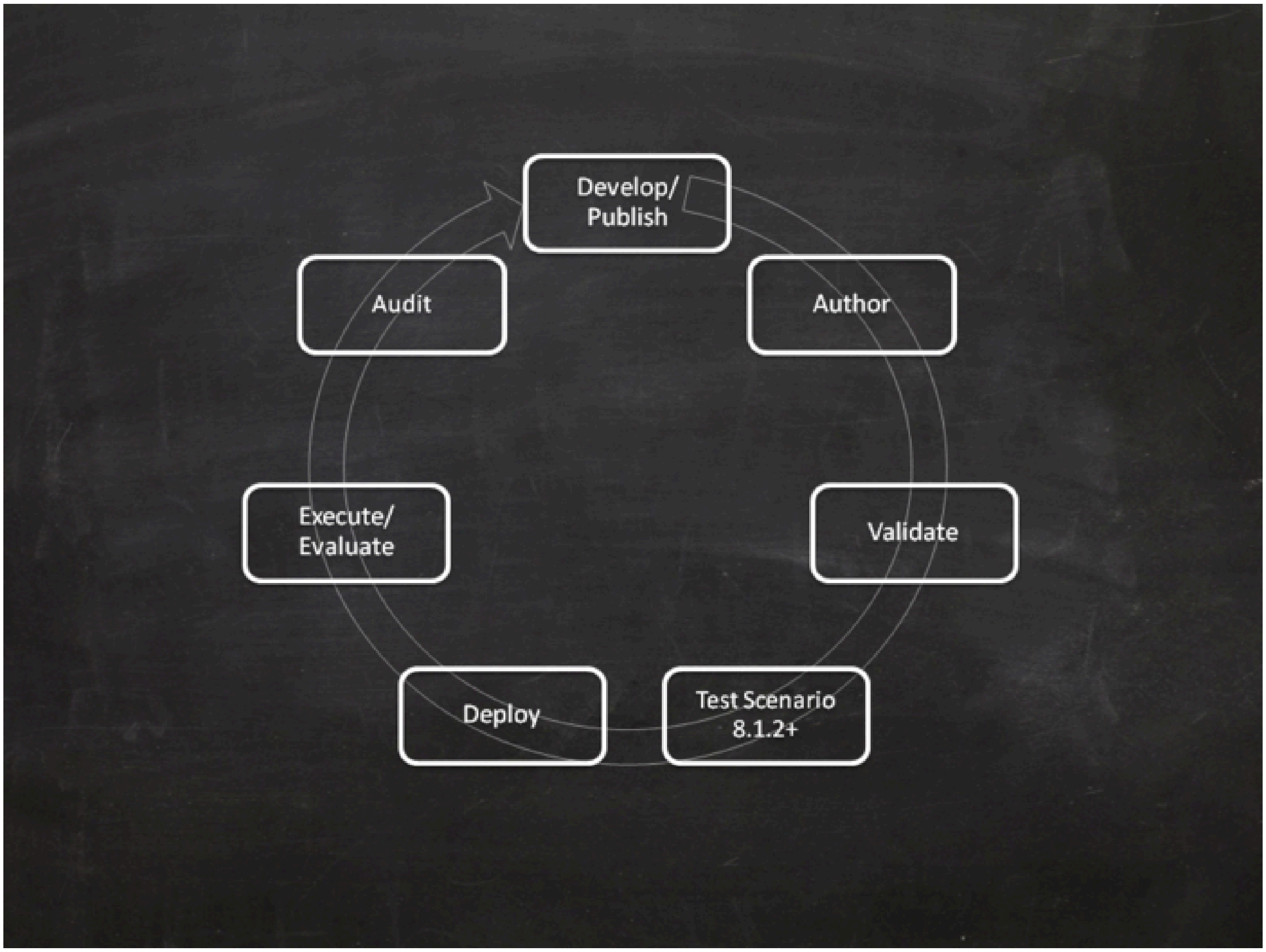
GRS consists of three software components:

Genesys Rules Development Tool (GRDT) is an Eclipse plug-in that allows advanced users (business rules developers) to create templates that define the discrete rule conditions and actions that will comprise the rules. Each rule condition and action includes the plain-language label that the business rules author will see, as well as the rule language mapping that defines how the underlying data will be retrieved or updated. For each rule condition and action, the template developer decides what kind of data the rules author will be providing. Some examples include whether the input should be an integer value, a non-integer numeric value, a string, a selection from a pre-defined list, or a selection from a dynamic list.

Genesys Rules Authoring Tool (GRAT) is a browser-based application used by business analysts to create and edit packages of business rules based on the templates created in the Genesys Rules Development Tool.

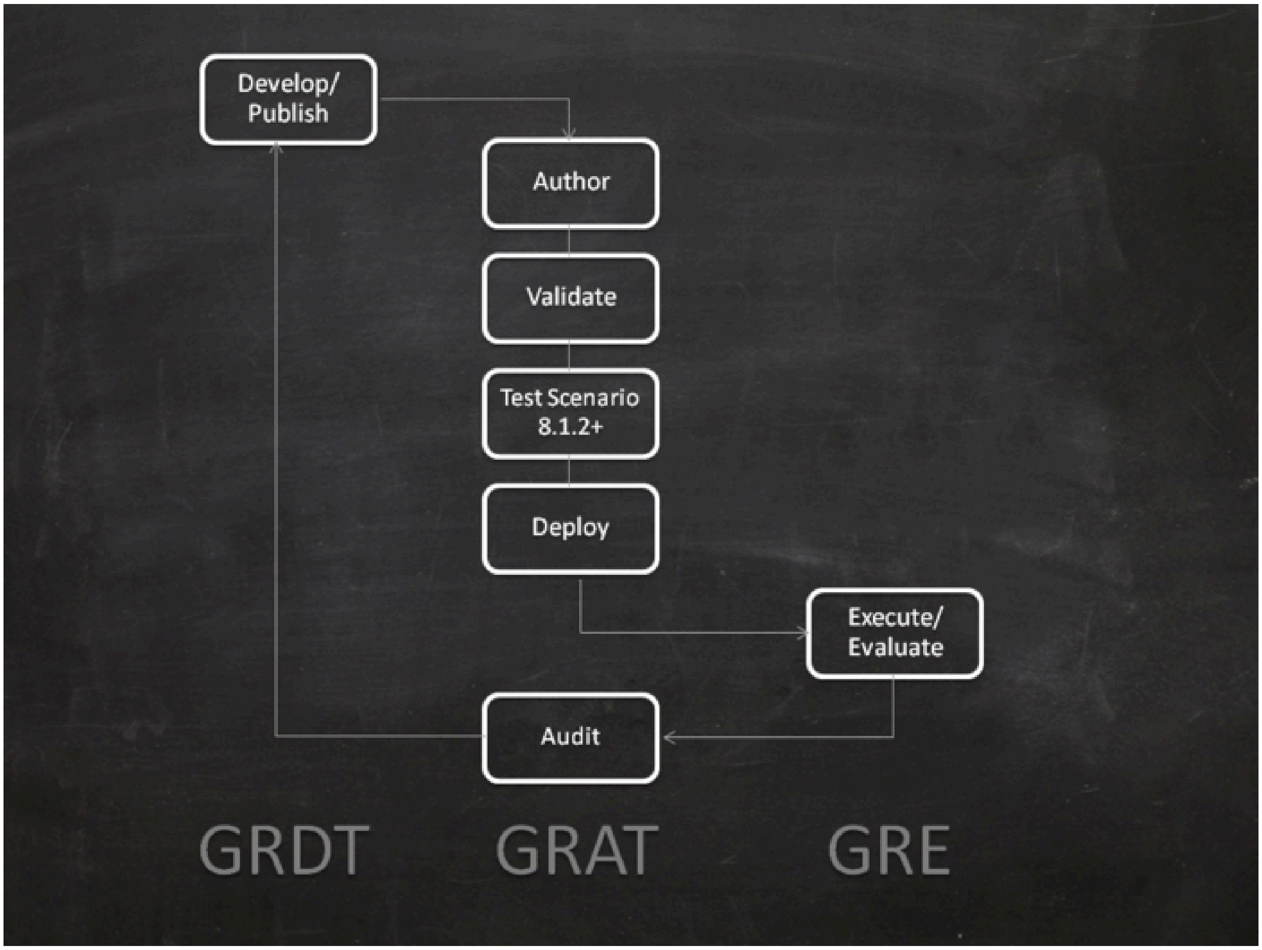
Genesys Rules Engine (GRE) evaluates the rule packages (groups of rules). Rule packages are deployed to the Rules Engine by the Rules Authoring Tool. When a rule package has been deployed, Genesys applications will be able to request the Rules Engine to evaluate the logic that is defined in this rule package.

GRS Lifecycle Overview



1. GRDT—Rule templates are DEVELOPed and the templates are PUBLISHed to the rules repository.
2. GRAT—Authors then create a rule package that incorporates one or more rule templates. GRAT is also where users:
    a. Create new rules packages that incorporate rule templates +DETAIL
    b. AUTHOR rules inside the rule package based on the rule templates +DETAIL
    c. VALIDATE the rules
    d. Run TEST SCENARIOS for the rules (from release 8.1.2 onwards) +DETAIL
    e. DEPLOY their rule package to the Genesys Rules Engine +DETAIL
    f. AUDIT the rules package +DETAIL
    g. GRE—Client applications (for example, the iWD business process (IWDBP)) then make requests to

the Genesys Rules Engine to VALIDATE and EVALUATE the rules in the rule package at various decision points in a task's lifecycle.



GRS Video Series

The following video topics show how to create a simple template from scratch in GRDT, publish it to GRAT, create a rule package and finally connect with the Composer Business Rules block to use the evaluation decision in a workflow. The use case is a simple call treatment flow which uses the customer's segment and call intention to make business decisions about how to route the contact.

You can follow the videos in sequence as they appear on these tabs.

<tabber>

Part 1 - Building a template in GRDT=

Building a template in GRDT

Name	Description	Link	Duration
Creating a template project	Setting up the Eclipse/ Composer environment, initial configuration and creating a template project	▶	7.59
Building the Fact Model	Building the Facts (data) that will be passed in by the application for evaluation by the rules engine.	▶	5.32
Creating Enumerations	Creating Enumerations (static lists) that the rule author will select from in GRAT to build a rule.	▶	2:13
Creating Parameters	Creating the Parameters that will be used in Conditions and Actions	▶	7:11
Creating Conditions	Creating the Conditions that the rule author will select from in GRAT to build a rule.	▶	5:23

Name	Description	Link	Duration
Creating Actions	Creating the Actions that the rule author will select from in GRAT to build a rule.	▶	6:18
Publishing	Publishing the template from GRDT to GRAT to enable the rule author to use it.	▶	2:11

|-| Part 2 - Building a Rule Package and Rule in GRAT =

Building a Rule Package and Rule

Name	Description	Link	Duration
Getting Started	Creating a rule package.	▶	5.32
Creating a Linear Rule	How to create a simple linear rule.	▶	6.43
Creating a Decision Table Rule	How to create a Decision Table rule.	▶	4.35
Creating Test Scenarios	Creating scenarios for testing rules.	▶	8.31
Deploying the Rule Package	How to deploy the rule package to a rules engine that will execute the rule(s).	▶	3.10

|-| Part 3 - Using Composer to execute a rule block=

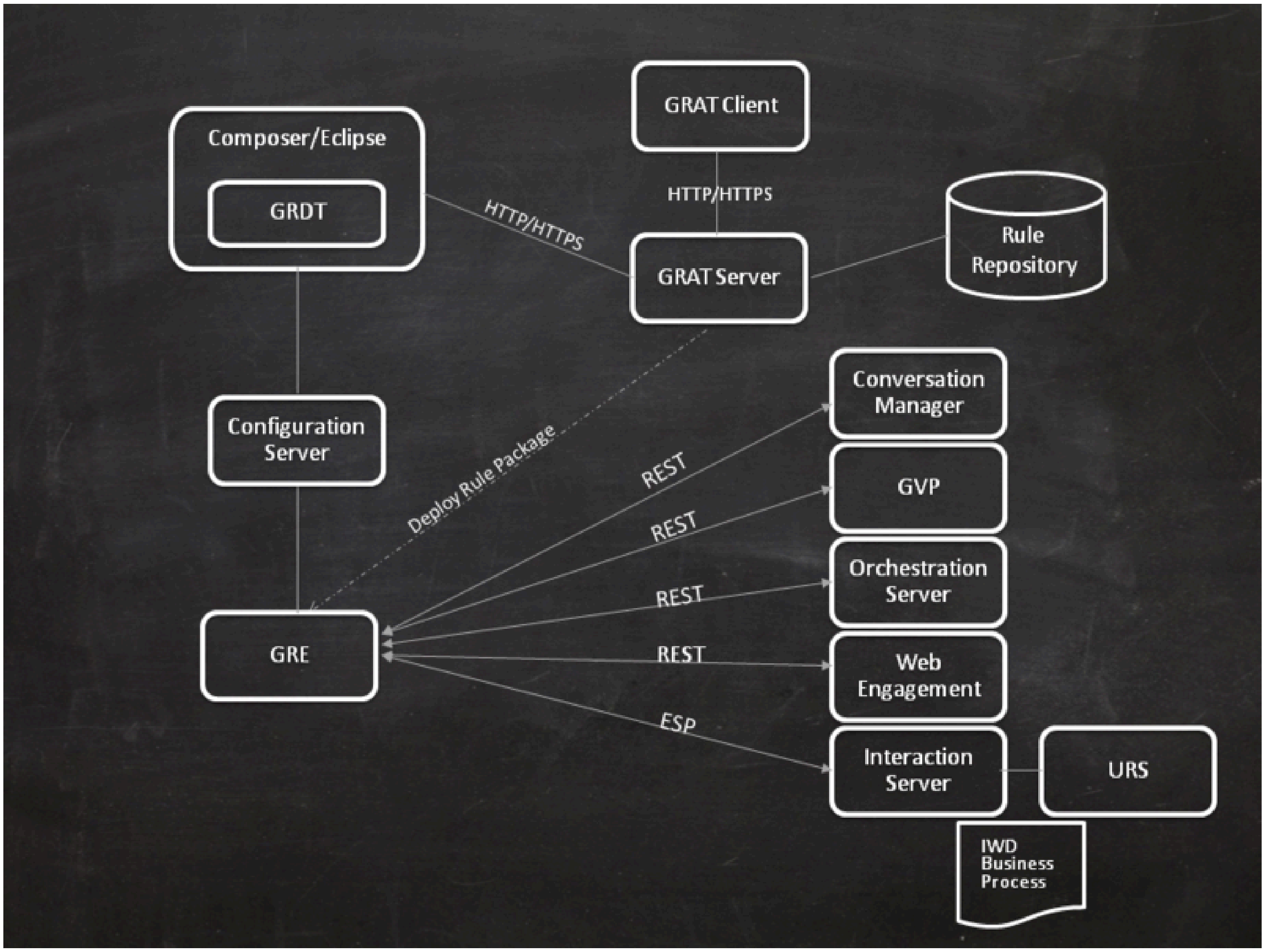
Using Composer to execute a rule block

Name	Description	Link	Duration
Getting Started	First steps in creating a Composer workflow that will use the rule.	▶	3.26
Creating a New Composer Project	Setting up the Composer workflow project.	▶	10.16
Testing the Rule BLock	Testing the new rule block	▶	2:21

GRS Architecture

Genesys Eco-System

The schematic below shows the main Genesys Rules System (GRS) building blocks, their interrelationships, and the external components that are involved.



Business Structure

The business structure is a hierarchy of business units. No business structure is created out-of-box for Genesys Rules System; the business structure must be configured in Genesys Administrator or Configuration Manager. For customers who are using the Genesys Rules System with intelligent Workload Distribution, the business structure is created in iWD Manager and then synchronized with Configuration Server, after which it becomes available for use by the Genesys Rules System.

The business structure that you configure will be visible in the Genesys Rules Authoring Tool. Each rule package will display the business structure for the Tenant. Each Tenant can contain one more Solutions as the first level of the hierarchy, and rules can be defined at each level (node) of the business structure from Solutions down.

Rules that are configured for the Solution, known as global rules, are executed first, followed by rules configured for the first node of the business structure, then rules configured for the second node, and so on. Global rules are only “global” within the defined rule package.

The business structure that you create can vary depending on a number of factors, including whether Genesys Rules System is to be used for iCFD. Sample structures are provided in the Deployment Guide. The structure can be product- or business-specific.

Object permissions are used to determine which elements of a business structure are visible to various users.

Rule Templates

Rule templates are created in GRDT, which is an Eclipse plug-in that can either be installed into a standalone Eclipse application or can be installed into Genesys Composer. Rule templates are used to define the building blocks that are used by rules authors to build rules for task classification and prioritization at the Global, Department, and Processes levels of the business structure of Genesys solution.

Rule templates are made up of several elements:

- Actions—Contain `then` expressions.
- Conditions—Contain `when` or `if` expressions.
- Enumerations—Define lists of possible choices that will be displayed to the business rule author.
- Fact models—All rule templates include a fact model with one or more facts.
- Events—From release 8.1.2, to support Complex Event Processing, template developers need to be able to designate certain facts as events.
- Functions—Sometimes used to support rule conditions and actions—for example, when parsing of timestamps is required.
- Parameters—Database, Web Service and Workforce Management parameters are used in the actions and conditions.

+ DETAIL

Support for User-Defined Template Types

In release 8.1.2, hard-coding of template types for iCFD has been removed. Users can now define template types according to their own needs (including iCFD if required). A template designer can assign a type to their templates, and then indicate whether or not that type supports events. GRAT now automatically displays the list of template types published to it, and users can select these user-defined template types or define new ones.

Rule Authoring

Rule authoring is done through GRAT. This topic describes how to log into GRAT, gives some general information about its usage, and how to use it for creating decision tables, linear rules, and business calendars.

Levels

In the Genesys Rules Authoring Tool, there are three levels at which business rules can be created:

- Rule Package (referred to as Global Rules)
- Department
- Process

Global rules enable you to specify rules that will apply to the entire solution. For example, they enable you to configure rules that classify or prioritize all tasks globally, instead of at a lower level of the business structure. Global rules are applied before any other rules.

This means that each rule phase (classification and prioritization) is triggered from within the relevant business process in the following sequence:

1. Global rules
2. Department rules
3. Process rules

When the appropriate node is selected on the rule package tree, you can then select the `Rules` tab to view or edit the rules for that level of the business structure. Rules are presented in a list, with an associated phase. The order of the rules is relevant, because they will be evaluated, within a particular phase, in the same order as they appear. You can change the order of rules by clicking the up and down buttons. The logic of a particular rule can be expressed as either a linear rule or a decision table.

Any extended or custom attribute can be read or updated by business rule conditions or actions, respectively.

Only rules on a particular node path are executed in any given rules run. The path of execution is determined by input to the Rules Engine on the execution request.

> **Important**
>
> The business structure is defined in Configuration Manager or Genesys Administrator.

Rule Packages

Rule packages provide the following capabilities:

- The ability to partition rules and facts so that they are small, well-defined, and apply only to a particular application or use. This makes them easier to debug and understand. The fact model is a description of the data. It contains field names and types which are grouped into tables/classes. Facts are input/output to rule execution and are instances of the tables/classes defined in the fact model.
- The ability to isolate rule packages from one another when executing rules. This also improves performance because the Rules Engine has fewer candidates to examine during the evaluation.
- The ability to update individual rule packages without affecting other deployed packages.
- The ability to import and export an entire rule package containing the rule definitions, business calendars, and also the templates that the rule package is dependent on.
- A rule package contains one or more rules plus the fact model that is needed to support the rules. You deploy rule packages individually to the Rules Engine.

|+DETAIL How to create a rules package|

Linear Rules

A linear rule is a business rule that has a set of conditions (when) and actions (then), and is used for a simple (linear) business case. A linear rule follows the following basic format:

```
WHEN {condition} THEN {action}
```

For example:

```
When a task is due in 1 to 8 hours, set the task's priority to 20.
```

When the condition is true, the action will occur. This form of rule is best for simple actions, such as assigning a value to return back to the application. Note, however, that linear rules can have multiple conditions and actions, or only actions with no conditions.

The conditions and actions that are available depend upon the rule templates that are included in the rule package.

|+DETAIL How to create a linear rule|

Decision Tables

Decision tables allow you to join a number of Linear Rules with the same set of conditions (when) and actions (then) to be used for a complex (structured) business case. Use decision tables to avoid dozens of linear rules with identical structure in the system.

Important

Choices in decision tables need to be mutually exclusive to avoid ambiguity. This ensures that there is only one outcome per evaluation. If the choices are not mutually exclusive, multiple rows can be executed, in no guaranteed order. The last row executed will determine the final result.

After you have created a decision table, you can create additional decision tables or linear rules, or deploy your rule package.

Defining a decision table is similar to defining a linear rule.

|+DETAIL How to create a decision table|

Business Calendars

A business calendar is a set of rules that define working days and hours, and holidays that are applicable for the business. Business calendars can be used in rules to perform date and time calculations, taking into account the working schedule of the business. Business calendars can be assigned once (for example, at the Global Rule level), or can be assigned dynamically in a rule when needed.

Business calendars consist of a set of standard mandatory attributes and optional business calendar rules. Business calendars must be assigned to a task before any business calendar-related calculations can be performed on task values. A Calendar can be assigned at the parent rule level, or to an individual rule. Only one calendar can be assigned to a task at a time, so a calendar can be assigned by one rule and then, overwritten by a later rule. Rule packages can contain one or more Business Calendars. Business calendars define the working days and hours of the organization. They can also be associated with any rule in the package.

Calendars are out-of-the-box classes available in the Fact Model that can be used by Rules. A calendar contains:

- Name
- Time zone
- Week start day and time
- Week end day and time
- Holidays (one or more)
- Time Change (one or more)

A holiday is fixed, relative, or annual.

- A fixed holiday contains the date of the holiday, including day, month, and year, such as 1/1/2010.
- A relative holiday contains the month and weekday of the holiday and whether it is on the first, second, third, fourth, or last day of that month, such as the third Thursday of November.
- An annual holiday contains the month and day of the holiday, such as July 4.

A time change indicates how the work hours can be adjusted on particular days; for example, defining a half day on a particular day of the work week. Like a holiday, a time change is fixed, relative, or annual and contains the same date definition as the corresponding holiday definition. In addition, the time change contains the start and end time for the defined date.

Business calendars are needed for defining rules based on work hours. For instance:

```
WHEN Task is idle for more than 3 Working Days THEN increase Priority by 20
```

```
WHEN Today is a holiday AND Task is urgent THEN Agent Group is “Urgent Care”
```

The italicized portions of the above examples use business calendar information.

|+DETAIL How to create a business calendar|

Rule Deployment

Once you have created all the necessary rules in a rule package, you can deploy the rule package (+DETAIL) to either a single GRE or to a cluster of GREs. Once the package is deployed, it can be invoked by a client.

Rule package deployment is done using GRAT, and is independent of the individual solutions deployment features. So, you can deploy a new version of a rule package for a Genesys solution without having to redeploy the entire solution.

When a rule is created or edited and it has not been deployed, there is a checkmark in the **Pending Deployment** column of the rule. In addition, when a rule package requires deployment or redeployment, there will be a visual indication next to the **Deploy Rules** node in the GRAT navigation tree, under the rule package itself.

The deployment process (whether you choose to deploy immediately or to schedule the deployment for later) attempts to compile the rule package and informs you of the result via the **Deployment Pending** pop-up message. You can check on the status of your deployment by looking at the **Deployment History** tab, which will now show a new status of `Pending`. When deployment is in pending status, you will not be able to cancel or undo it.

This process enables you to correct any errors before deployment. In addition, if you attempt a deployment that would duplicate either;

- An already scheduled deployment or;
- An attribute of an already scheduled deployment, such as;
  - The same rule package
  - For the same snapshot
  - For the same destination server/cluster

an appropriate message is displayed. You can then either change the attributes of your deployment, or go to **Deployment History** and change/delete the scheduled deployment.

To use the deployment screen, you must have deploy permissions set up in Genesys Administrator.

|+DETAIL Deploying Rules Packages|

From Release 8.5.2

From release 8.5.2, partial deployments to GRE clusters are allowed, Previously, any deployment to a cluster in which one node failed to update would result in the deployment being rolled back. From 8.5.2, partial deployments are allowed, and the cluster nodes can be configured to auto-synchronize so that temporary disconnects or additions to the cluster can be handled without interruption to deployments. **+ DETAILS**

From Release 8.5.3

From release 8.5.3, GRATs can also be configured in clusters to provide greater resilience. Deployments can be amended and rescheduled by other GRAT users in the cluster, which was not possible previously. **+ DETAILS**

From Release 8.5.303

From release 8.5.303 you can also undeploy rules packages from single GREs or clusters. **+ DETAILS**

Rule Evaluation

After a rule package has been deployed to GRE or to a cluster of GRE engines, it can be invoked by a client.

Rule Evaluation Order

Within all the rules that are configured for a particular phase, GRE will evaluate rules in this order:

- Global rules (package-level)
- Department rules
- Process rules

To ensure that GRE performs its evaluation in this order, the `sequential-mode` option (in the `settings` section in the GRE `Application`) must be set to `false`. This is the default setting.

To ensure that the GRE evaluates all rules within a particular phase, within a particular node of the business structure hierarchy, the `group-by-level` option (in the `settings` section in the GRAT server `Application`) must be set to `true`. This is the default setting.

Within the rules that are defined at a particular node of the business structure hierarchy (for example, `Department`), GRE will follow the salience (order) that is defined in the rule package. This is controlled by the rules author, who can move rules up and down in the GRAT using the arrow button on the right side of the rule.

Controlling Rule Evaluation Order

Within a set of rules for a particular phase, at a particular node in the business structure hierarchy, an action of one rule can influence a condition in a subsequent rule. To do this, it is necessary to include the `update($data)` parameter; in the `Rule Language Mapping` of the rule action or as a separate rule action. See the following example.

Example: Setting Rules at a Specific Phase

If you had one classification rule at the `Department` level that set the priority of a task to `100` and a second classification rule at the `Department` level that will increase the priority by `90`, but only if the priority has already been set to `100`:

The first classification rule would look like this:

```
WHEN

   Priority is not equal to 100

THEN
```

```
         Set Priority to 100

         Update
```

The second classification rule would look like this:

```
      WHEN

         Priority is equal to 100

      THEN

         Increase Priority 90
```

In the first classification rule, it is important to specify that the priority be set to `100` only when it is not already equal to `100`. This is to avoid a problem where GRE might get into a loop during rule evaluation. Also in the second classification rule, there is a rule action called `Update`. The Rule Language Mapping for this rule action is: `update($data);`

You could also modify the `Rule Language Mapping` of the `Set Priority` rule action, to add `update($data);` to the end of the `Rule Language Mapping`. For example:

```
setIntValue("Priority", {priority}, $data); update ($data);
```

In the latter case, however, you would want to create a new version of the `Set Priority` rule action and give it a different Language Expression to help differentiate it, such as `Set extended Priority`.

Controlling Rule Evaluation Order in Decision Tables from 8.5.0

By default, up to release 8.5.0, rules were executed from the bottom up. In release 8.5.0, your system administrators can configure rule execution to be "bottom-up" or "top-down". The Rule Evaluation Order indicator at the bottom of the screen shows you which of these is selected, and a ToolTip is available when you hover your cursor over this indicator. Any changes made to this configuration will apply dynamically, but only take effect after a restart or a browser refresh.

For more detailed information, see the working example in the **Deployment Guide** (new document).

Rule Testing

Before deploying a rules package to the Genesys Rules Engine, subject to having the relevant permissions in Genesys Administrator, you can :

- Create, modify and run one or more test scenarios for each rule package
- Add input data, business context and phase data and expected results
- Review the test outcomes in plain language
- Import and export the test scenarios in the same way as with rules packages

These test features allow rules authors to test any changes made to existing rules packages before deploying them, in order to ensure that no errors are introduced.

The rule testing functionality is available via a node in the navigation tree called **Test Scenarios**. Click this node to use the rule testing features.

DETAILS

A/B/C Split Testing

From release 8.5.3, A/B/C Split Testing is supported.

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.

DETAILS

Role-Based Access Control

Genesys Rules System role-based access control utilizes Configuration Server-defined access groups and roles to control visibility and access to rule packages, rule templates, rules, and business calendars. Because these objects are not stored in the Configuration Server database they will not have security permissions associated with them, as Configuration Server objects do. The GRAT server will utilize the access permissions for the container object, and the Genesys Rules System objects will inherit these access permissions.

Role-based access control requires Configuration Server 8.0.2 or higher and Genesys Administrator 8.0.2 or higher.

Rule packages and business calendars inherit their access permissions from the Tenant object with which they are associated and the Business Structure folder access permissions. Business rules are associated with a specific node in the business structure. Their access permissions are inherited from the Configuration Server–defined node with which they are associated (the business structure nodes are created by using Configuration Manager or Genesys Administrator). Rule templates have Script objects created in Configuration Server that are used to hold the individual access permissions of the rule template. Additionally, rule templates inherit the access permissions from the business structure node with which they are associated.

For more detail about Role-Based Access Control and GRS, go to the **Deployment Guide** (new document).

For a full discussion of Genesys Role-Based Access Control, please refer to the **Genesys Security Deployment Guide** (new document).

Using GRS with Other Genesys Solutions

Supported Solutions

GRS is used by Genesys solutions, including:

- intelligent Workload Distribution
- Conversation Manager
- Genesys Web Engagement

It is also used by:

- VXML applications executed by the Genesys Voice Platform
- SCXML applications executed by the Genesys Orchestration Server

Each of these solutions also has its own GRS configuration requirements—for example;

- Configuring the proper rule template type when creating a new template
- Configuring the proper rule package type when creating a rule package
- Proper use of rule phases

Support for intelligent Workload Distribution (iWD)

GRS provides all the business rules functionality for the Genesys intelligent Workload Distribution (iWD) solution, a business application for dynamically prioritizing the distribution of work tasks to the people who are best suited to handle them. GRS enables business users to define priorities, Service Level Agreements, and other attributes of tasks.

iWD uses GRS to provide the rules engine service, and rules development and authoring user interfaces. iWD provides a Standard Rules Template for use with GRS, and the Genesys Rules Authoring Tool (GRAT) can be launched from iWD Manager without the need for separate user authentication.

Read about IWD and the Genesys Rules System (new document)

Support for Conversation Manager

Conversation Manager is a contact center solution that creates coherent customer communication in real-time customer engagement applications that span one or more channels such as web, mobile, chat, IVR and voice.
In a nutshell, Conversation Manager helps you recognize moments when you can take action to improve the customer experience. Within Conversation Manager, **Context Services** helps you to recognize the moment and the business rules help you to take action.

GRS ships with a **special template for conversation rules** that enables much closer integration between GRS and Context Services blocks without having to build a new template from scratch in Composer (create a Fact model, actions and conditions, and pass complex data structures between the Composer Business Rules block and the rules engine and re-evaluate the returned data, and so on).

Read about Conversation Manager (new document).

Support for Web Engagement

Genesys Web Engagement uses a special template type—Complex Event Processing (or CEP). This template type enables rule developers to build templates that rule authors then use to create rules and packages that use event fact types. Choosing this template type determines:

- How the Drools Rule Language (DRL) is eventually generated by the GRAT.
- Which applications the rule package can be deployed to.

Read about Genesys Proactive/Web Engagement (new document)

GRS and Genesys Composer

You can use Genesys Composer to create applications for rules evaluation requests. Composer 8.1.0 is required, because that is the version in which the Business Rule block was introduced. Refer to the **Composer documentation** (new document) for more information about how to use the Business Rule block.

This function block is available on both the callflow and the workflow diagram palettes.