



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.

Quick Start Guide

Genesys Web Engagement 8.5.0

Table of Contents

Genesys Web Engagement Quick Start Guide	3
Getting Started with the Playground Application	4
Working with the Playground Application	16
Testing the Playground Application	32

Genesys Web Engagement Quick Start Guide

Welcome to the *Genesys Web Engagement 8.5 Quick Start Guide*. In this guide you'll find steps to get Genesys Web Engagement up and running quickly by installing and working with the Playground application.

Get started!

Getting Started with the Playground Application

The fastest way to get started with Genesys Web Engagement is to use the Playground application, which is included in the Genesys Web Engagement installation package; after you install GWE, you can find the application in the **GWE_installation_directory\apps\playground** folder. Then you just build and deploy it and you are ready to work with the features of Genesys Web Engagement.

Read below to learn more about the application or continue on to "Installing Genesys Web Engagement" for installation details and running the Playground application.

- **Read more — Playground application**

The Playground application not only includes a Web Engagement application, but also a functioning sample website you can use to test and demonstrate Web Engagement's functional capabilities. This allows you to see Genesys Web Engagement at work on a website that is designed to highlight the GWE features and show you how to use them.

English | Francais

PlayGround

Start page

Change instrumentation script ?≡

Widgets customisation ?≡

Reactive engagement

Get pacing state for chat

Chat with pacing

true

Chat without pacing

System events

Page Entered

Page Exited

Proactive engagement

Singleton	Engage!
Timeout 30	Engage!
Sequence	Click here first → Click here to
Sequence Ads	Click here first → Click here to
Set	Click ↔
Counter (3)	Count to Engage
Simple search	<div>proactive engagement</div>
Search with category	<div>What is</div>

The Playground application website.

Browser Support

The Playground website was created using Bootstrap and works on all web browsers that support Bootstrap. See the Bootstrap documentation for details: <http://getbootstrap.com/getting-started/#support>

- **Read more — GWE components**

Genesys Web Engagement Components

You should be familiar with the following Genesys Web Engagement components before you dive into the Playground application:

- [DSL File](#)
- [Categories](#)
- [Instrumentation Script](#)
- [Rules Files](#)
- [Strategies](#)
- [Widgets](#)
- [Configuration Options](#)
- [Monitoring Agent API](#)

DSL File

The DSL file is used to manage and customize business events. When you [create an application](#), a set of Domain Specific Language (DSL) files that are used by your application is also created. These files are defined in the **apps\Your application name\resources\dsl** directory. You can use the DSL to define Business events (read about the structure of these events [here](#)) that are specific to your solution needs.

Default domain-model.xml

The **domain-model.xml** is the main default DSL file for your application:

```
<?xml version="1.0" encoding="utf-8" ?>
<properties>
  <events>
    <!-- Add your code here
    <event id="" name="">
    </event>
    -->

    <!-- This is template for your search event -->
    <!--
    <event id="SearchEvent" name="Search">
      <trigger name="SearchTrigger" element="" action="click" url="" count="1" />
      <val name="searchString" value="" />
    </event>
    -->
    <event id="TimeoutEvent30" name="Timeout-30" condition=""
postcondition="document.hasFocus() === true">
      <trigger name="TimeoutTrigger" element="" action="timer:30000"
type="timeout" url="" count="1" />
    </event>

  </events>
</properties>
```

By using the **<event>** element, you can create as many business events as you need. These events can be tied to the HTML components of your page and can have the same name, as long as they have different identifiers (these identifiers must be unique across the DSL file, to make a distinction between the events sent by the browser). It can be useful to associate several HTML components with the same event if these HTML components have the same function. For instance, you can define several events associated with a search feature and give all these events the same name: "Search".

For each event, you can define triggers which describe the condition to match in order to submit the event:

- Triggers can implement timeouts.
- Triggers can be associated with DOM events.
- You can define several triggers for the same event (see [<trigger>](#) for further details).

Each trigger should have an `element` attribute that specifies the document's DOM element to attach the trigger to, and the `action` attribute, which species the DOM event to track.

You can specify standard DOM events for the action:

- Browser Events
- Document Loading
- Keyboard Events
- Mouse Events
- Form Events

In addition to the standard DOM events, the DSL supports the following two values: `timer` and `enterpress`.

The following example generates a "Search" event if the visitor does a site search. The "searchString" value is the string entered in the "INPUT.search-submit" form.

```
<event id="SearchEventClick" name="Search">
  <trigger name="SearchTrigger" element="INPUT.search-submit" action="click"
url="" count="1" />
  <val name="searchString" value="INPUT.search-submit" />
</event>
```

If the DSL uses the optional `condition` attribute, the event's triggers are installed on the page if the condition evaluates to true. The following example creates a Business event with a time that can be triggered only if the text inside the `<h1>` tag is "Compare":

```
<event id="InactivityTimeout4CompareProductsEvent"
name="InactivityTimeout4CompareProducts" condition="$('h1').text() == 'Compare'">
  <trigger name="InactivityTimeout4CompareProductsTrigger" element=""
action="timer:10000" type="timeout" url="http://www.MySite.com/site/olspage.jsp"
count="1"/>
</event>
```

If the DSL uses an optional `postcondition` attribute, this can manage how an event is generated by checking a condition after the actions are completed. The following example creates a Business event timeout by timer if a page is in focus. In this case, the event does not generate if the page is opened in the background:

```
<event id="TimeoutEvent10" name="Timeout-10" condition=""
postcondition="document.hasFocus() === true">
  <trigger name="TimeoutTrigger" element="" action="timer:10000" type="timeout"
url="" count="1" />
</event>
```

A DSL trigger can use the `type` attribute. This can have a value of either `timeout` or `nomove`, which specifies how the timer action works. If the type is `timeout`, then the timer interval begins after the page is loaded. If the type is `nomove`, then the timer resets each time the user moves the mouse.

You can also apply the optional `url` attribute. This attribute defines the URL of the specific page that raises the Business event. The Business event is not submitted if the current document's URL does not match the URL parameter.

Finally, you can apply the optional `count` attribute. This attribute specifies how many times the trigger needs to be matched before the event is generated and sent to the Web Engagement Server.

For more information about the DSL elements, see the [Business Events DSL](#).

Categories

You can use the GWE Plug-in for Genesys Administrator Extension to define, in a few clicks, Web Engagement categories that contain business information related to URL or web page titles. These categories are used in the CEP rule templates, which provide rules that define when to submit actionable events to Web Engagement — this is what starts the engagement process.

For example, let's look at Solutions on the Genesys website. In this scenario, you can define a Solution category associated with the <http://www.genesys.com/solutions> page and several or all solution sub-pages, such as <http://www.genesys.com/solutions/cloud> or <http://www.genesys.com/solutions/enterprise-workload-management>.

- To associate the category with all the pages containing the "solutions" string in the URL, you can create the "solutions" tag. This tag defines the "solutions" string as a plain text expression to search in the events triggered by the visitor browsers.
- To set up a specific list of sub-pages for the Solutions category, you can create a tag for each sub-page:
 - The "cloud" tag, which defines the "cloud" string as the plain text expression to search in the events triggered by the visitor browsers.
 - The "enterprise-workload" tag, which defines the "enterprise-workload-management" string as the plain text expression to search in the events triggered by the visitor browsers.

Now your rules can use this category to match solution-related pages.

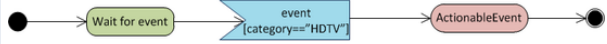
Instrumentation Script



The Tracker Application instrumentation script is a small piece of JavaScript code that you paste into your website to enable Web Engagement functionality. You create this script by using the Script Generator in the Genesys Web Engagement Plug-in for Genesys Administrator Extension.


Rules Files

Genesys Rules System (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 Genesys Web Engagement. GRS implements the CEP (Complex Event Processing) template for GPE. This template type enables rule developers to build templates that rule authors then use to create rules and packages. These rules use customized event types and rule conditions and actions. 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.

The Playground application has a pre-built rule file, called **rule.com.playground.drl**, located at **GWE_installation_directory\apps\playground\resources\drl**. The rules included in this file demonstrate the basic concepts of how Genesys rule files are used.

Singleton	
"Rule-101 singleton"	<p>The rule receives each single event as a formal parameter. If the event's value matches the right category, then the actionable event is sent to the Web Engagement Server.</p> 
Expression Example	<p>When page transition event occurs that belongs to category \$category</p> <p>Then generate actionable event</p>

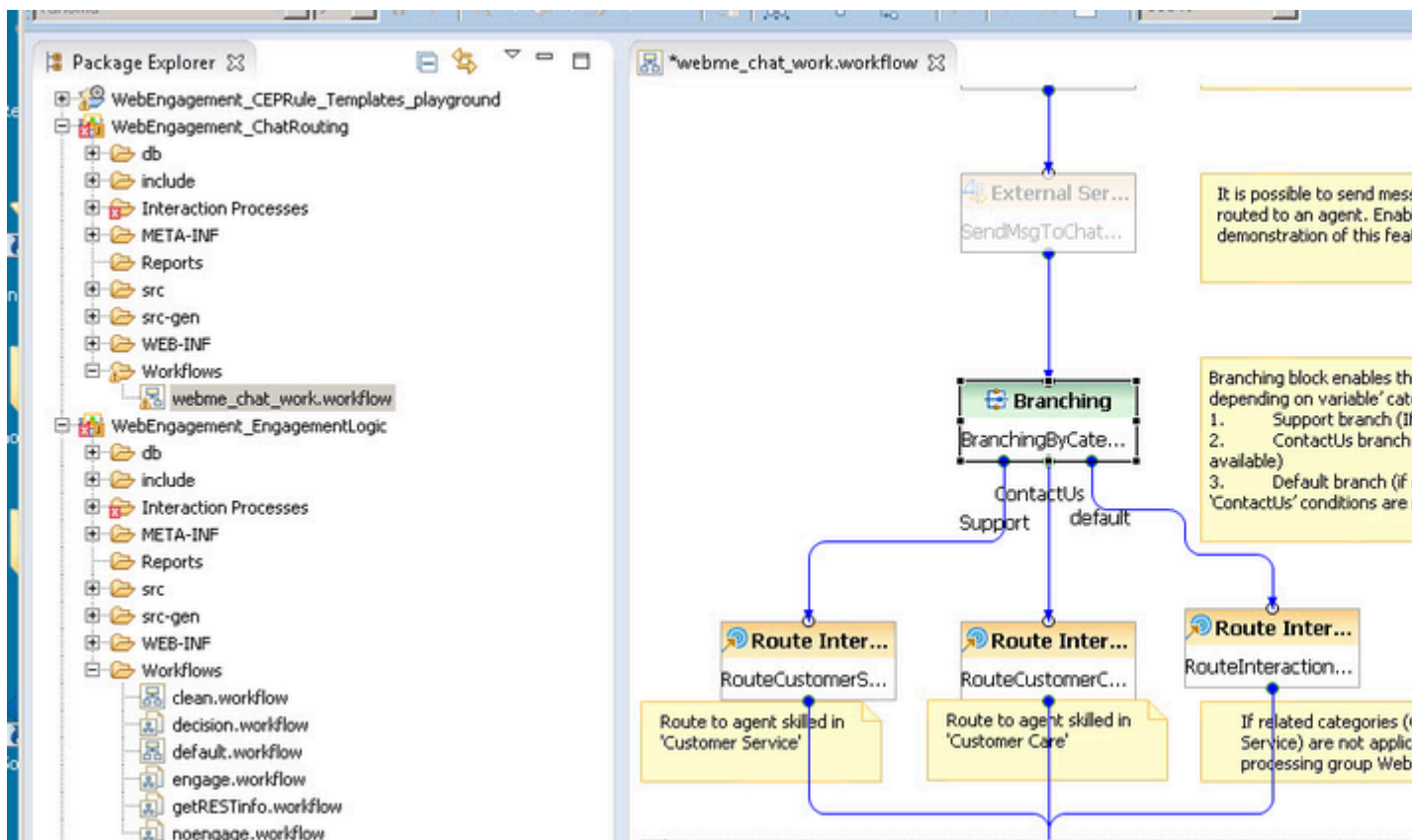
Sequence	
"Rule-108 sequence"	<p>This rule analyses the event stream received from the categorization engine and builds the sequence of events by category values. As soon as the event sequence is completed, the actionable event is submitted. Note that the event sequence must follow a specific order.</p> 
Expression Example	<p>When page transition event occurs that belongs to category \$category1 save as \$event1</p> <p>and event following \$event1 with category \$category2 save as \$event2</p> <p>(...)</p> <p>and event following \$eventⁿ⁻¹ with category \$categoryⁿ save as \$eventⁿ</p> <p>Then generate actionable event based on \$eventⁿ</p>
Set	
"Rule-136 Set"	<p>This rule analyses the event stream received from the categorization engine and collects the events by category values. As soon as the event set is completed, the actionable event is submitted. If you use this rule, the event order is not taken into account.</p> 
Expressions	<p>When page transition event occurs that belongs to category \$category1</p> <p>or page transition event occurs that belongs to category \$category1</p> <p>(...)</p> <p>or page transition event occurs that belongs to category \$categoryⁿ</p> <p>Then generate actionable event</p>
Counter	

<p>"Rule-122 counter"</p>	<p>This rule analyses the event stream received from the categorization engine and counts events which occur for a given category. As soon as the counter is reached, the actionable event is submitted.</p> 
<p>Expressions</p>	<p>When Category \$category counts \$count times</p> <p>Then generate actionable event</p>

Strategies

When you create your application, Genesys Web Engagement also creates default chat routing and engagement logic strategies in the `\apps\application_name\resources\composer-projects\` folder. Orchestration Server (ORS) uses these strategies to decide whether and when to make a proactive offer and which channels to offer (chat or web callback). You can modify these strategies by importing them into Composer.

The following shows the Chat Routing workflow, where interactions are routed to agents with "Customer Service" or "Customer Care" skills:



A Chat Routing workflow example.

When you alter the strategies, you must save your changes, generate the code, redeploy, and restart your Genesys Web

Engagement application to apply those changes.

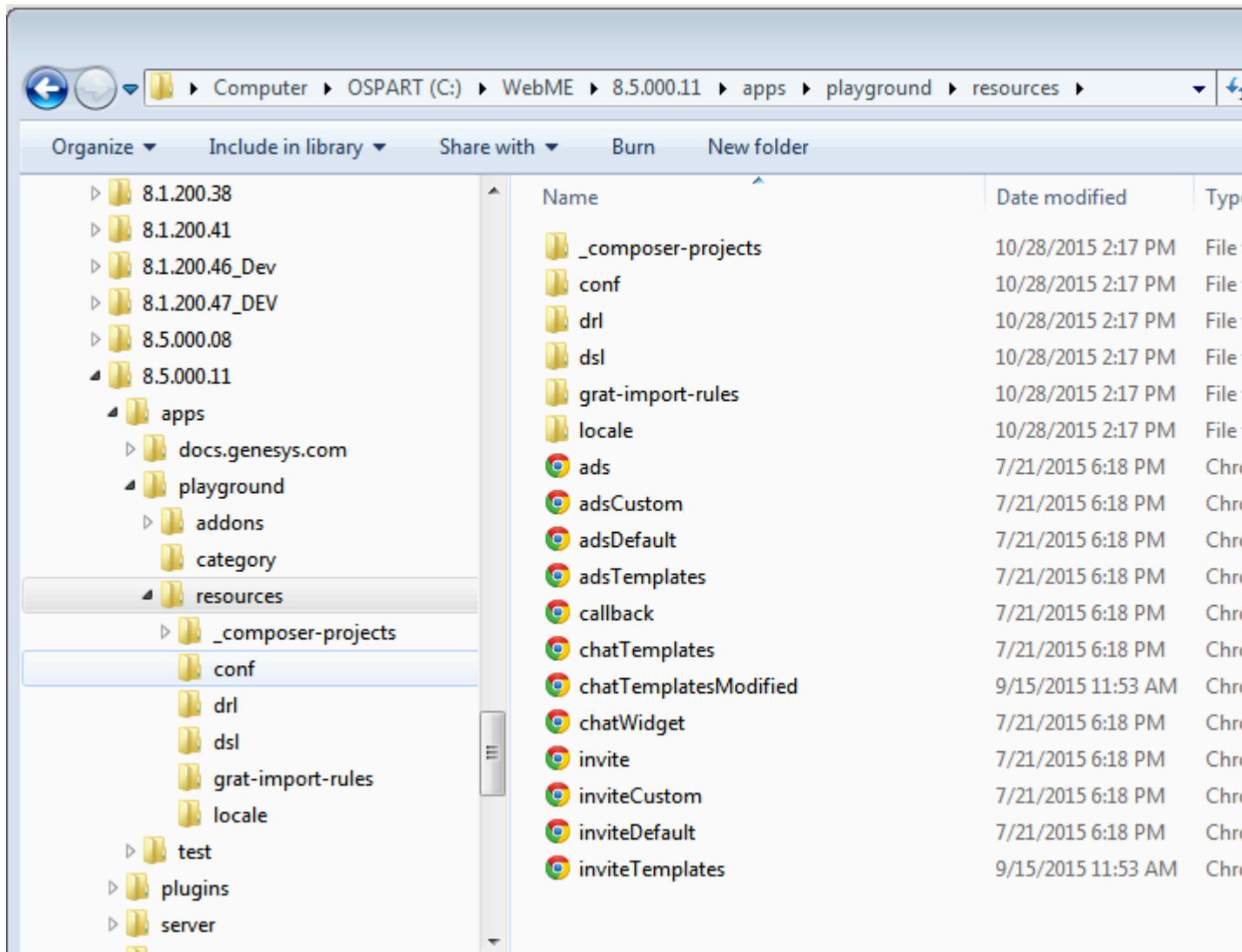
Widgets

Genesys Web Engagement includes three pre-integrated Browser Tier widgets:

- Invitation Widget
- Chat Widget
- Callback Widget

These widgets are based on HTML, CSS, and Javascript, and can be customized to suit the look and feel of your website. When you create an application, Genesys Web Engagement creates a Composer project that has a directory dedicated to the widgets:

`GWE_installation_directory\apps\application_name\resources`.



The Browser Tier Widgets.

You can customize the widgets using the HTML files in this directory, CSS, or the [Notification Service API](#). You can also build your own version of a particular widget with the help of the [Web Engagement APIs](#).

Configuration Options

For a full list of configuration options for the Web Engagement servers, see [Configuration Options](#).

Monitoring Agent API

The Monitoring Agent API is available through the JavaScript libraries provided by the Web Engagement Server at runtime. By using this API in your web pages, you can submit events to the Genesys Web Engagement Server, independently from the set of events and conditions defined in the DSL files loaded by the browser's Monitoring Agents. For more information, see the [Monitoring Agent API documentation](#).

Installing Genesys Web Engagement

In order to use the Playground application, you must first install and configure Genesys Web Engagement and its related components, and then deploy the Playground Application.

For details, see the steps below:

1. Review the [prerequisites](#) and make sure your Genesys environment meets the requirements.
2. [Install the Genesys Web Engagement Servers](#). These procedures describe how to install and configure the Web Engagement Server cluster and nodes.
3. Make sure you have properly configured agents and add agent shortcuts to the Web Engagement Chat and Web Engagement Voice agent groups: In Genesys Administrator, navigate to **Provisioning > Accounts > Agent Group**, and select Web Engagement Chat. Go to the agent tab and click **Add** to add the agent to the group. Repeat these steps for the Web Engagement Voice agent group.
4. [Install the Plug-in for Workspace Desktop Edition](#). You can install this plug-in if you want to enable chat and web callback engagement features so you can see Web Engagement from the agent's perspective when you test the Playground Application.
5. [Install the Plug-in for Genesys Administrator Extension](#). You can install this plug-in if you want to create or modify categories when you test the Playground Application.
6. Go to the **apps\playground\addons\demo\war\html** folder and open the **instrumentation.html** file in an editor. Modify the following lines to replace `demosrv.genesyslab.com` with the fully qualified domain name of your host:
 - `var serverDefault = 'http://demosrv.genesyslab.com:9081';`
 - `var serverSecure = 'http://demosrv.genesyslab.com:9081';`
 - `domainName: 'demosrv.genesyslab.com',`Save your changes and close the file.
7. [Deploy an Application](#). To complete these procedures, make sure you use the application name **playground**.
8. Make sure the components in your Genesys environment are running:
 - [Configuration Server](#)
 - [Universal Routing Server](#)
 - [Interaction Server](#)

- **Orchestration Server**
- **Stat Server**
- **Chat Server**
- **Interaction Workspace**

9. **Start the Web Engagement Servers.**

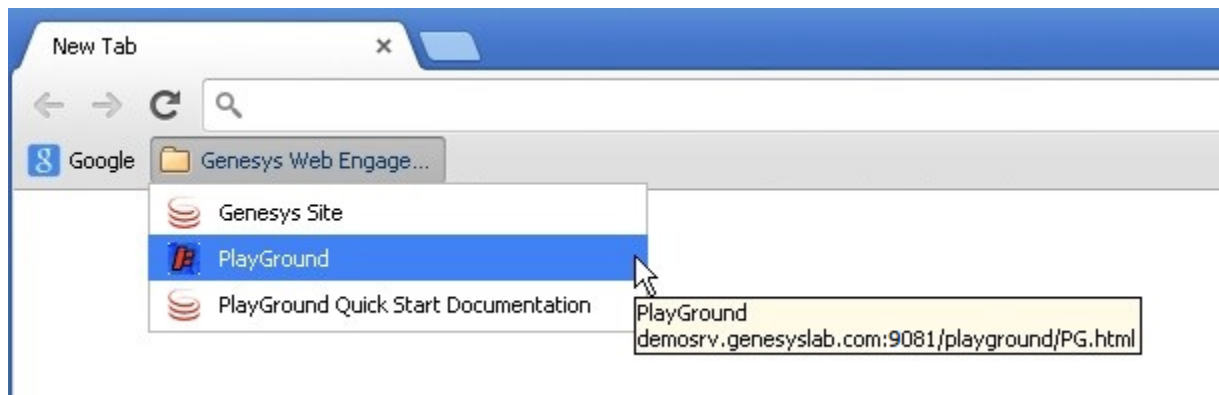
10. Login any agents from the Web Engagement Chat or Web Engagement Voice agent groups.

Congratulations! You are ready to run the Playground application.

Running the Playground Application

You can now run the application in any web browser or the **InTools** application — a GWE tool that allows you to work with the DSL and view events.

To launch InTools, navigate to **GWE_installation_directory/tools/intools/**. Take the **intools.crx** file and drop it into the Extensions page of your Google Chrome or Chromium browser. Click the **Genesys Web Engagement** bookmark in the toolbar and select **Playground** to launch the Playground application.



You can access the Playground application directly from InTools.

To open the Playground application from a web browser, navigate to `http://gwe_server_host_name:gwe_server_port/playground/PG.html`.

- *gwe_server_host_name* — The name or IP address of the host for the Web Engagement Server.
- *gwe_server_port* — The default port for the Web Engagement Server.

For example, `http://127.0.0.1:9081/playground/PG.html`

Example: Web Chat End-to-End

Complete the steps below for an example of an end-to-end scenario for proactive chat.

1. Make sure the Playground application is open and you have a logged in and ready agent in either the Web Engagement Chat or Web Engagement Voice agent groups.
2. Click **Engage!** next to the Singleton business rule.


Proactive engagement

Singleton	Engage!
Timeout 30	Engage!
Sequence	Click here first → Click here to Engage
Sequence Ads	Click here first → Click here to Ads
Set	Click ↔ Click
Counter (3)	Count to Engage
Simple search	<input type="text" value="proactive engagement"/> <input type="button" value="Search"/>
Search with category	<input type="text" value="What is"/> <input type="button" value="Search"/>

Click **Engage!**

3. The chat pop-up appears. Click **Chat**.

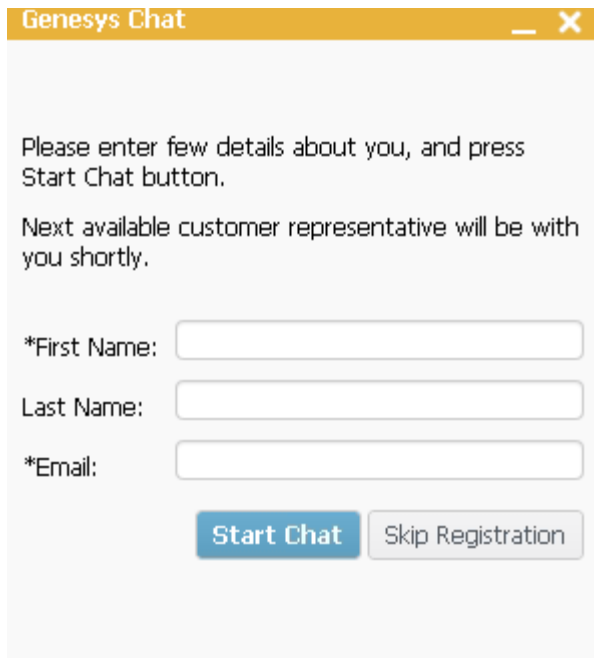
Chat



Good afternoon! Would you like some help with the selection? Our technical experts are available to answer questions

Click **Chat** in the chat pop-up.

4. Fill out the registration form and click **Start Chat**.



The screenshot shows a window titled "Genesys Chat" with a yellow header bar containing a minus sign and a close button. The main content area is light gray and contains the following text: "Please enter few details about you, and press Start Chat button." and "Next available customer representative will be with you shortly." Below this text are three input fields: "*First Name:", "Last Name:", and "*Email:". At the bottom of the form are two buttons: "Start Chat" (blue) and "Skip Registration" (gray).

Click **Start Chat** after filling out the registration form.

5. On the agent side, accept the chat. You just walked through your first proactive engagement!

Next Steps

- [Working with the Playground Application](#)

Working with the Playground Application

The Playground application website features a header block and four other functional blocks that can all be collapsed with a click. You can use the links and buttons in these blocks to trigger chat, business rules, and business events.

[English](#) | [Francais](#)

PlayGround

Start page

[Change instrumentation script ?](#) 

[Widgets customisation ?](#) 

Reactive engagement

[Get pacing state for chat](#) 

[Chat with pacing](#) true 

[Chat without pacing](#) 

System events

[Page Entered](#) 

[Page Exited](#) 

Proactive engagement

[Singleton](#) [Engage!](#)

[Timeout 30](#) [Engage!](#)

[Sequence](#) [Click here first](#) → [Click here to Engage](#)

[Sequence Ads](#) [Click here first](#) → [Click here to Ad](#)

[Set](#) [Click](#) ↔ [Click](#)

[Counter \(3\)](#) [Count to Engage](#)

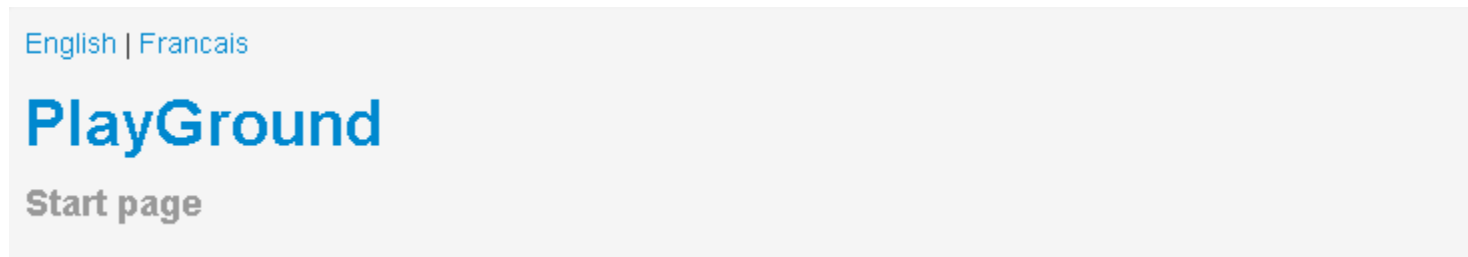
[Simple search](#)

[Search with category](#)

The Playground application.

PlayGround

The **PlayGround** header block displays the current page name (**Start page** in the image above) and links to the Read Me file for the application.



The **PlayGround** header block.

Change instrumentation script

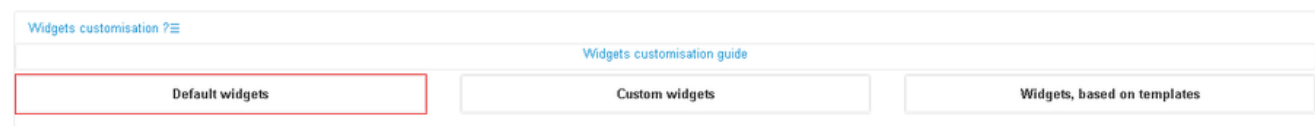
The **Change instrumentation script** block allows you to configure the instrumentation script on the fly, so that changes are applied without restarting the Web Engagement servers. For details about configuring the script, see [Configuring the Instrumentation Script](#).



The **Change instrumentation script** block.

Customizing Your Widgets

The **Widgets customisation** block allows you to switch between preconfigured widgets on the fly, so that changes can be applied without restarting the Web Engagement Server.



- **Default widgets**—Click this link to use the default GWE widgets.
- **Custom widgets**—Click this link to use widgets with custom CSS.
- **Widgets, based on templates**—Click this link to use a custom template file for your widgets.

You can also get more information about how to customize your widgets by viewing the customization guide that is available from the **Widgets customisation guide** link in this block.

As you can see here, the customization guide contains a brief description of the major GWE functionality and information on how to customize your widgets:

Guide

General principles

Instrumentation

Invitation

Widgets

Default templates

Change CSS on customer page

Change CSS in widgets

Additional content

Modification logo

Agent avatars

Text messages & localization

Authorisation

Back to top

Change CSS on customer page

You can modify the style of any element in the widget. In this example all modified CSS classes contains on your page.
In this example, Google Chrome browser is used (similar developer features also available in any other modern browser).

Start of Procedure

- Launch chat widget
- Right-click on any element in the widget and click "Inspect Element" to cast the Developer Tools.
- You can see that element consists of three elements, each having its own dedicated CSS class. You can create the new styling for these elements.
- You can copy CSS to this textfield (Click on text field for edit):

```
div.gwc-chat-embedded-window { border: 1px solid #606968 !important; border-radius: 8px 8px 8px 8px !important;} div.gwc-chat-head {background: linear-gradient(to top, #606968, #d0d0d0) !important;border-radius: 8px 8px 0px 0px !important;}
```

Notice: if you modify default CSS from widget - add **!important** after class declaration (as in example).

- Launch chat widget for view changes.

Notice: The HTML structure and CSS classes are subject to change. Genesys do not guarantee backwards compatibility of them in the future versions. This means that you may need to update your customizations when updating the chat widget.

Additional example | Link to documentation with details

Change CSS in widgets

You can modify the style of any element in the widget. In this example all modified CSS classes contains in widget template.

- Create widget prototype with [Change CSS on customer page](#)
- Modify [default template](#).
- You can modify default or add new CSS classes

default:

```
<div class="gwc-chat">
```

modified:

Reactive engagement

The **Reactive engagement** block allows you to initiate chat interactions with an agent simply by clicking a link.

Reactive engagement

Get pacing state for chat

?

Chat with pacing

true

?


Chat without pacing

?

The **Reactive engagement** block.

The block contains three links that demonstrate reactive chat, with and without the pacing algorithm.

- Get pacing state for chat — Displays `true` if the pacing algorithm allows reactive chat and `false` if the pacing algorithm doesn't allow reactive chat. See [Configuring the Pacing Algorithm](#) for details.
- Chat with pacing — Opens a registration form for a chat if the pacing algorithm allows reactive chat.
- Chat without pacing — Opens a registration form for a chat, regardless of whether the pacing algorithm allows reactive chat.

You can modify the code for each of the actions by clicking the related view code button , making your changes, and then clicking **Save**.

Proactive engagement

Proactive engagement

Singleton	Engage!		
Timeout 30	Engage!		
Sequence	Click here first → Click here to Engage		
Sequence Ads	Click here first → Click here to Ads		
Set	Click	↔	Click
Counter (3)	Count to Engage		
Simple search	<input type="text" value="proactive engagement"/>	<input type="button" value="Search"/>	
Search with category	<input type="text" value="What is"/>	<input type="button" value="Search"/>	

The **Proactive engagement** block.

The **Proactive engagement** block contains a set of links that trigger business rules. Clicking one of these links changes the address of the page, the page title, and the name of the page in the **PlayGround** header block. In this way, the application simulates a functioning multi-page website.

To trigger business rules, the following conditions must be true:

- All system components must work. See Step 8 in [Installing Genesys Web Engagement](#) for component information.
- The website must have a properly configured instrumentation script in the **Change instrumentation script** block.
- All resources from the instrumentation script should be available.
- There are available agents for the selected media channel.

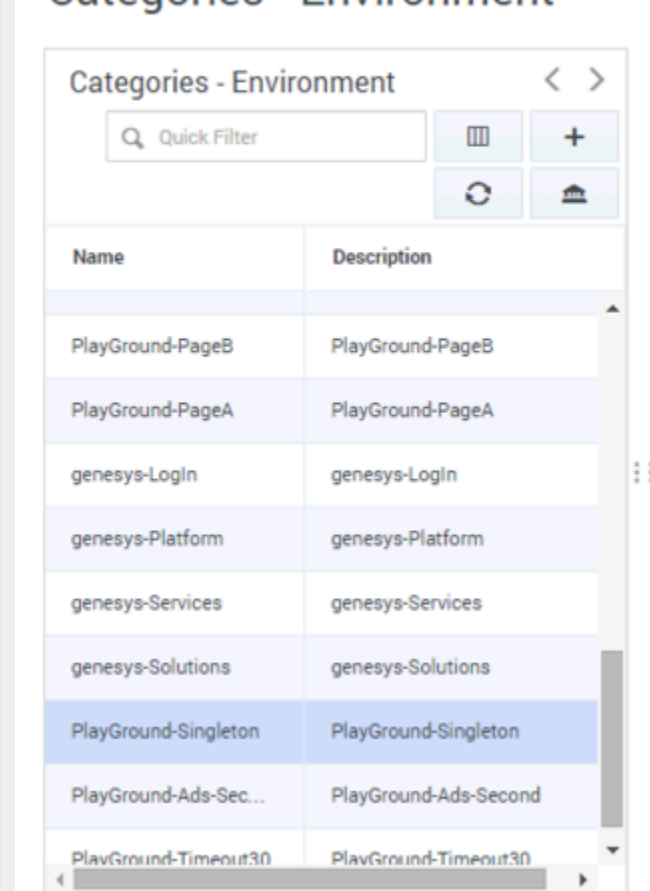
- The invitation limit (3 by default) should not be exhausted. The invitation limit is set in the Engagement strategy for the application. See [Engagement Policy \(Decision Workflow\)](#) for details.

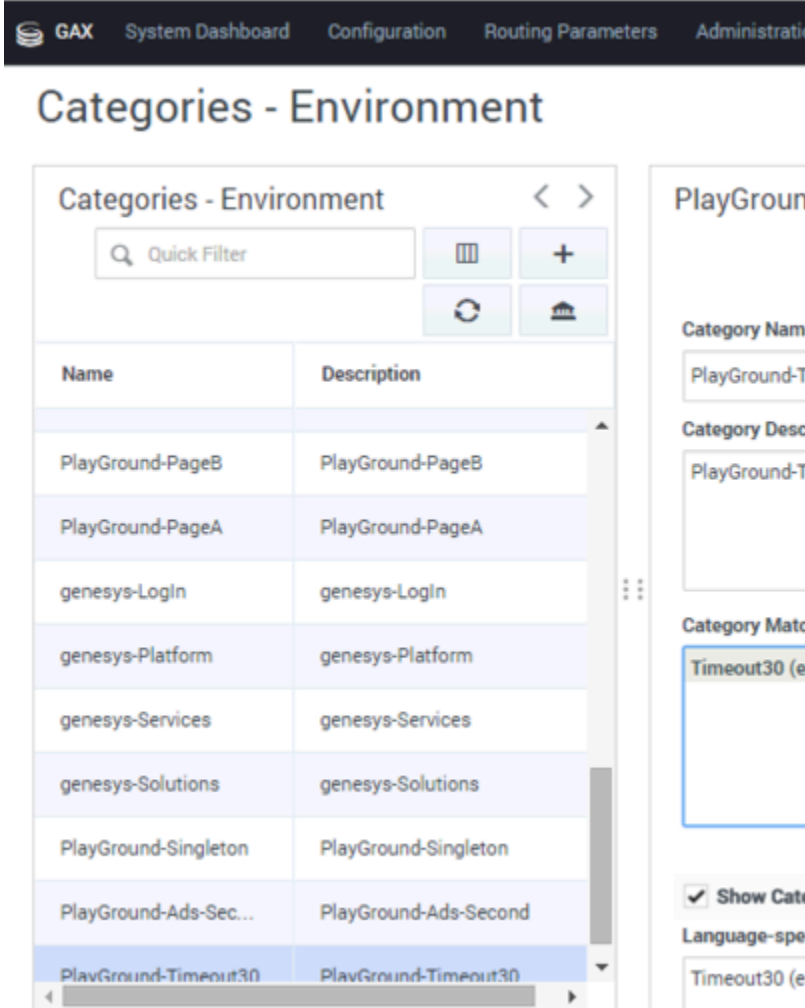
Here is a summary of events when you trigger a business rule:

- The application displays the engagement invitation. You can accept, reject, or ignore the invite.
- If you ignore, the invitation disappears. If you reject, the invitation closes.
- If you accept and are logged in (see the **User Identification** block for details), the application does not display the registration form and you begin the chat session.
- If you accept and are not authorized, the application displays the registration form. If you complete the registration form, the information is passed on to the agent in Interaction Workspace. If you do not complete the registration form, you communicate with the agent as an anonymous user.

The Proactive block lets you try out different business rules based on the default rule templates included with Genesys Web Engagement. See [Creating Business Information](#) for details about rule templates and [Deploying a Rules Package](#) for more information about rules.

Business Rule	Description
Singleton	Entrance to the page is specified in the PlayGround-Singleton category. This example demonstrates the interaction of a specific page with the business logic. Category: PlayGround-Singleton

Business Rule	Description
	 <p>The Playground-Singleton category.</p> <p>Entrance to the page is specified in the Playground-Timeout30 category. A business rule is triggered after a 30-second delay. This example demonstrates the use of the delay on the page.</p> <p>Category: Playground-Timeout30</p>
Timeout 30	

Business Rule	Description
	<div></div> <p>The PlayGround-Timeout30 category.</p>
Sequence	<p>Serial input to the first page First, then on page Second. The sequence may be any length. For example, this could be extended to check if all the necessary pages of a Wizard have been successfully completed.</p> <p>Categories: PlayGround-Seq-First and PlayGround-Seq-Second</p>

Business Rule

Description

GAX

System Dashboard

Configuration

Routing Parameters

Administration

Categories - Environment

Quick Filter

+

↺

↻

Name	Description
PlayGround-Seq-First	PlayGround-Seq-First
genesys-SearchCate...	genesys-SearchCategory
PlayGround-Counter	PlayGround-Counter
PlayGround-Seq-Sec...	PlayGround-Seq-Second
PlayGround-Search	PlayGround-Search
genesys-Contact	genesys-Contact
genesys-About	genesys-About
PlayGround-Ads-First	PlayGround-Ads-First
PlayGround-PageB	PlayGround-PageB

PlayGround-Seq-First

Category Name

PlayGround-S

Category Desc

PlayGround-S

Category Match

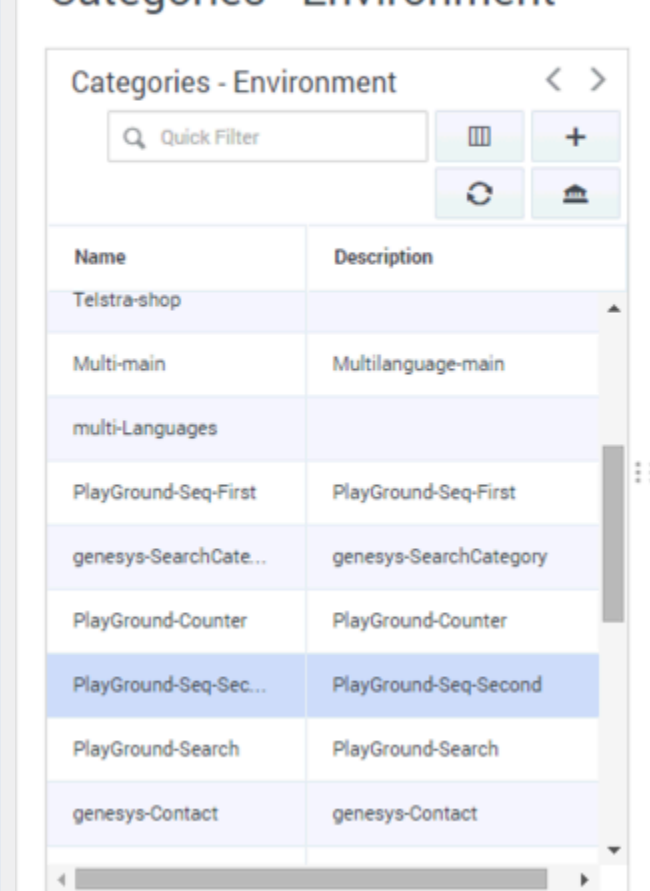
Seq-First (en

✓ Show Cate

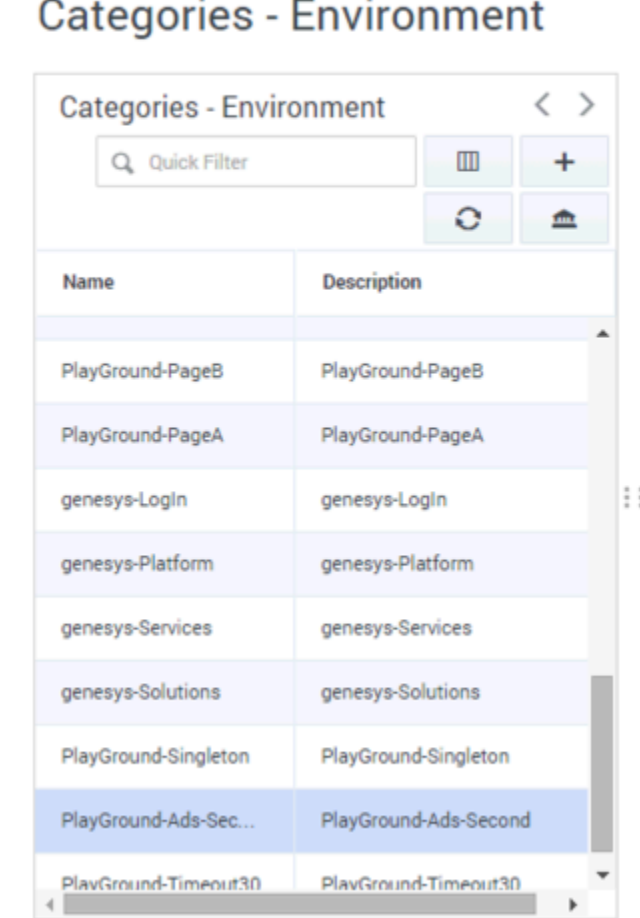
Language-spe

Seq-First (en

The PlayGround-Seq-First category.

Business Rule	Description
	 <p>The PlayGround-Seq-Second category.</p>
	<p>Serial input to the first page AdvertisementOne, then on page AdvertisementTwo. The sequence may be any length. For example, this could be used to display advertisements.</p> <p>Categories: PlayGround-Ads-First and PlayGround-Ads-Second</p>

Business Rule	Description																				
	<div data-bbox="824 289 1624 1291"> <div data-bbox="836 304 1624 346"> GAX System Dashboard Configuration Routing Parameters Administration </div> <div data-bbox="852 367 1372 430"> <h2>Categories - Environment</h2> </div> <div data-bbox="852 451 1437 1291"> <div data-bbox="868 462 1421 504"> Categories - Environment </div> <div data-bbox="917 514 1421 619"> <input type="text" value="Quick Filter"/> <div> <div>+</div> <div>↺</div> <div>↻</div> <div>🏠</div> </div> </div> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>PlayGround-Seq-Sec...</td> <td>PlayGround-Seq-Second</td> </tr> <tr> <td>PlayGround-Search</td> <td>PlayGround-Search</td> </tr> <tr> <td>genesys-Contact</td> <td>genesys-Contact</td> </tr> <tr> <td>genesys-About</td> <td>genesys-About</td> </tr> <tr> <td>PlayGround-Ads-First</td> <td>PlayGround-Ads-First</td> </tr> <tr> <td>PlayGround-PageB</td> <td>PlayGround-PageB</td> </tr> <tr> <td>PlayGround-PageA</td> <td>PlayGround-PageA</td> </tr> <tr> <td>genesys-Login</td> <td>genesys-Login</td> </tr> <tr> <td>genesys-Platform</td> <td>genesys-Platform</td> </tr> </tbody> </table> </div> <div data-bbox="1469 451 1624 1291"> <div data-bbox="1485 462 1624 504"> PlayGround </div> <div data-bbox="1502 598 1624 640"> Category Name </div> <div data-bbox="1502 640 1624 682"> <input type="text" value="PlayGround-A"/> </div> <div data-bbox="1502 693 1624 735"> Category Desc </div> <div data-bbox="1502 735 1624 777"> <input type="text" value="PlayGround-A"/> </div> <div data-bbox="1502 892 1624 934"> Category Matc </div> <div data-bbox="1502 934 1624 976"> <input type="text" value="Ads-First (en"/> </div> <div data-bbox="1502 1155 1624 1197"> <input checked="" type="checkbox"/> Show Cat </div> <div data-bbox="1502 1197 1624 1239"> Language-spe </div> <div data-bbox="1502 1239 1624 1281"> <input type="text" value="Ads-First (en"/> </div> </div> </div>	Name	Description	PlayGround-Seq-Sec...	PlayGround-Seq-Second	PlayGround-Search	PlayGround-Search	genesys-Contact	genesys-Contact	genesys-About	genesys-About	PlayGround-Ads-First	PlayGround-Ads-First	PlayGround-PageB	PlayGround-PageB	PlayGround-PageA	PlayGround-PageA	genesys-Login	genesys-Login	genesys-Platform	genesys-Platform
Name	Description																				
PlayGround-Seq-Sec...	PlayGround-Seq-Second																				
PlayGround-Search	PlayGround-Search																				
genesys-Contact	genesys-Contact																				
genesys-About	genesys-About																				
PlayGround-Ads-First	PlayGround-Ads-First																				
PlayGround-PageB	PlayGround-PageB																				
PlayGround-PageA	PlayGround-PageA																				
genesys-Login	genesys-Login																				
genesys-Platform	genesys-Platform																				

Business Rule	Description
	 <p>The PlayGround-Ads-Second category.</p>
Set	<p>Bypassing pages set in any order. The user can go to PageA, then on to PageB, or vice versa. This example could be used to demonstrate triggering business rules when filling in all the fields in a form.</p> <p>Categories: Playground-PageA and Playground-PageB</p>

Business Rule

GAX

System Dashboard

Configuration

Routing Parameters

Administration

Categories - Environment

Categories - Environment

Quick Filter

Name	Description
PlayGround-PageB	PlayGround-PageB
PlayGround-PageA	PlayGround-PageA
genesys-Login	genesys-Login
genesys-Platform	genesys-Platform
genesys-Services	genesys-Services
genesys-Solutions	genesys-Solutions
PlayGround-Singleton	PlayGround-Singleton
PlayGround-Ads-Sec...	PlayGround-Ads-Second
PlayGround-Timeout30	PlayGround-Timeout30

PlayGround

Category Name

PlayGround-P

Category Desc

PlayGround-P

Category Matc

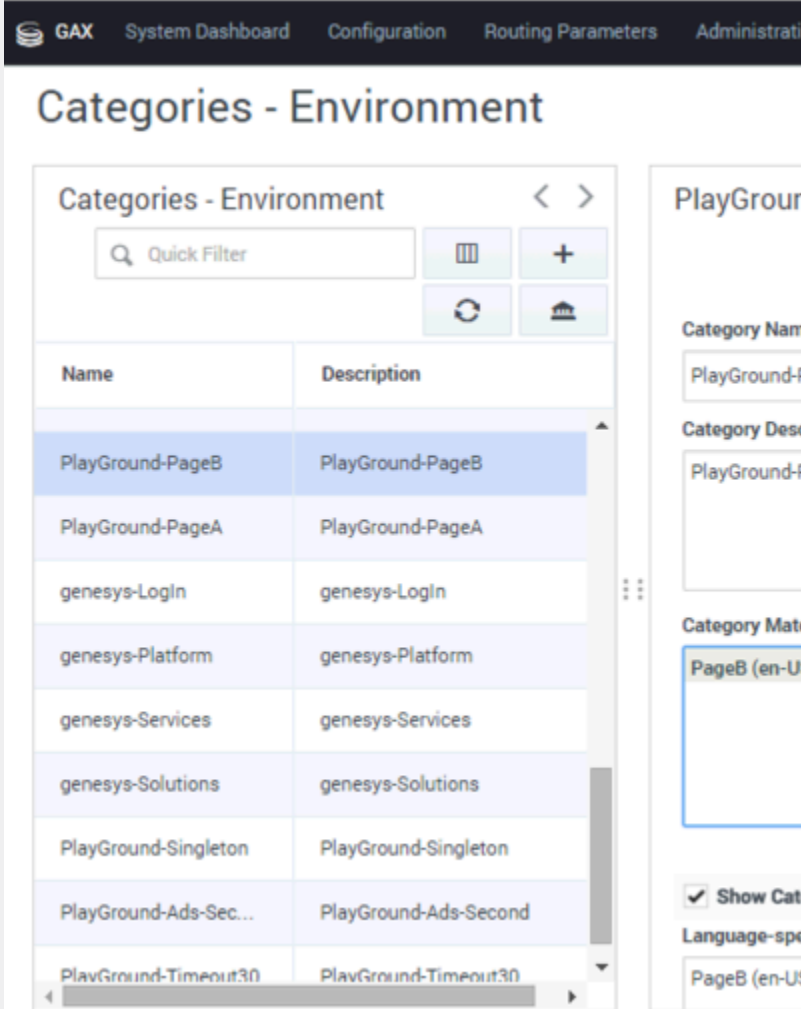
PageA (en-US

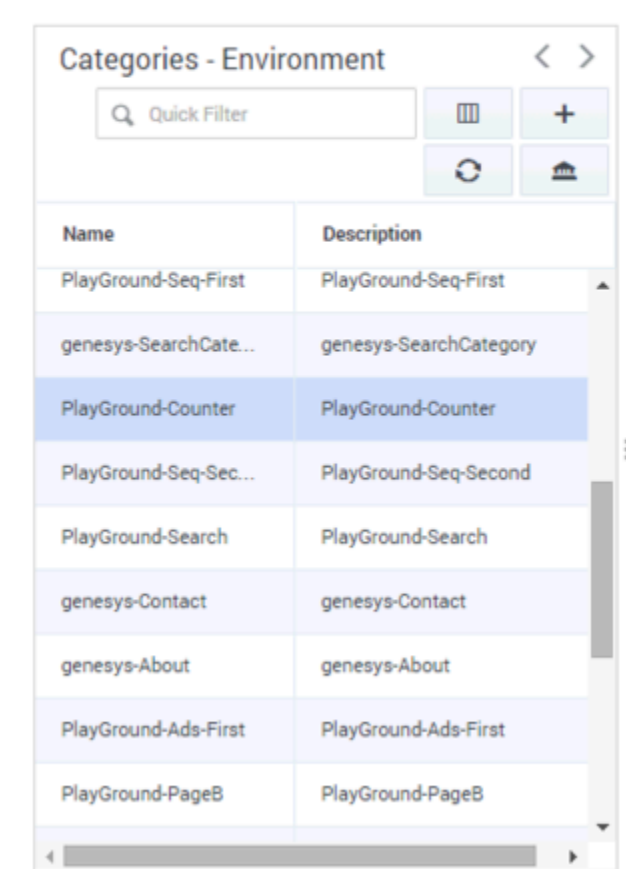
☒ Show Cate

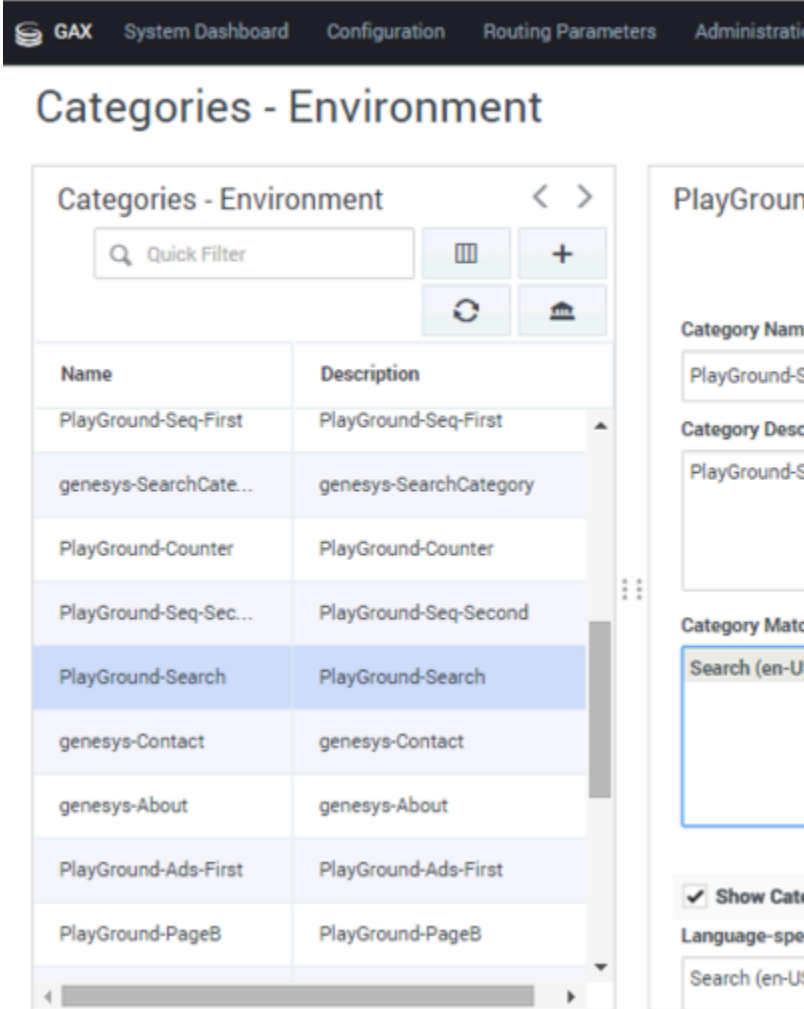
Language-spe

PageA (en-US


The Playground-PageA category.

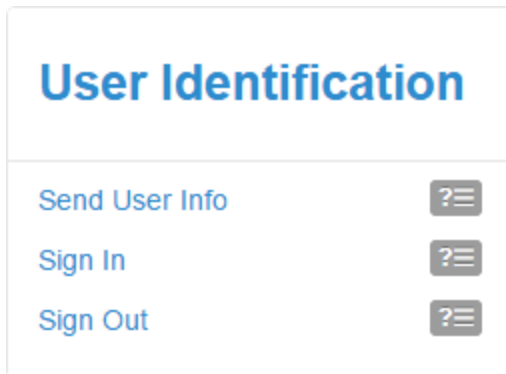
Business Rule	Description
	 <p>The Playground-PageB category.</p>
Counter (3)	<p>Counts multiple entries on the same page.</p> <p>Category: Playground-Counter</p>

Business Rule	Description
	 <p>The Playground-Counter category.</p>
Simple search	Search for a specific keyword. The keyword is hard-coded in the business rule.
Search with category	<p>Search for a specific keyword. The keyword is defined by the Playground-Search category. A regular expression can be used to describe the keyword.</p> <p>Category: Playground-Search</p>

Business Rule	Description
	 <p>The PlayGround-Search category.</p>

User Identification

The **User Identification** block contains links to send events that are generated by the **Monitoring API**. All events in this block can be modified by clicking the related view code button  and then **Save**. You can use the user information to manage registration (from the registration form for the authorized user) and send the necessary information to the agent.




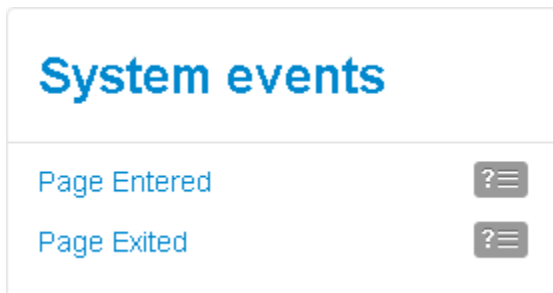
The **User Identification** block.

This block demonstrates three generated events:

- **Send User Info**—Send events with user information. You can use this event to identify an authorized user who has previously visited the site.
- **Sign In**—Send the authorized user login event.
- **Sign Out**—Send the authorized user log out event.

System events

The **System events** block contains links that demonstrate the **PageEntered** and **PageExited** system events that are part of the **Monitoring JS API**. You can modify the code for each of the actions by clicking the related view code button , making your changes, and then clicking **Save**.



The **System events** block.

Important: The **PageEntered** event updates the current **pageID**. Make sure that a **PageExited** event is sent before sending a **PageEntered** event, as it is important for correctly modeling the actions of a Single Page Application.

Testing the Playground Application

To give you a better idea of how to use the Playground application, let's walk through an example of building a "keyword search" scenario.

Test Scenario Steps

Here are the steps in our scenario:

1. An anonymous user visits the site.
2. The user enters a search term, "contract", in the search box.
3. The user browses the search results.
4. A chat invitation window appears.
5. The user accepts the invitation.
6. The user fills out the registration form and submits it.
7. The user chats with an agent.
8. The agent sees pages the user visited.
9. The agent sends the user a link to a page of interest.
10. The user exits the chat session.

Test Scenario Preparation

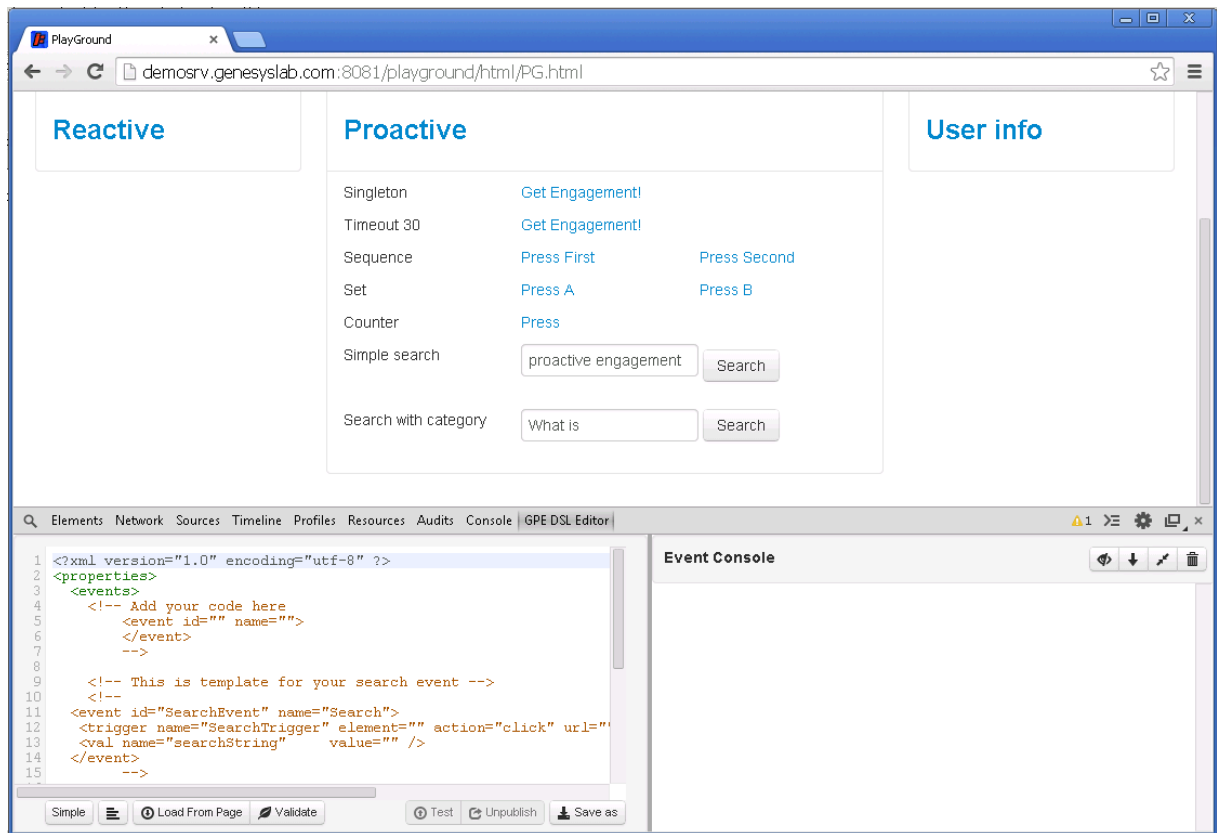
Here are the steps to complete before we can walk through our test scenario. Note that for steps 1 through 6, you do not need to take any action if you are using the Playground application, as these items are covered out of the box.

1. Customize the DSL file with the InTool application.

Complete the steps below to add the events to the DSL file for our test scenario.

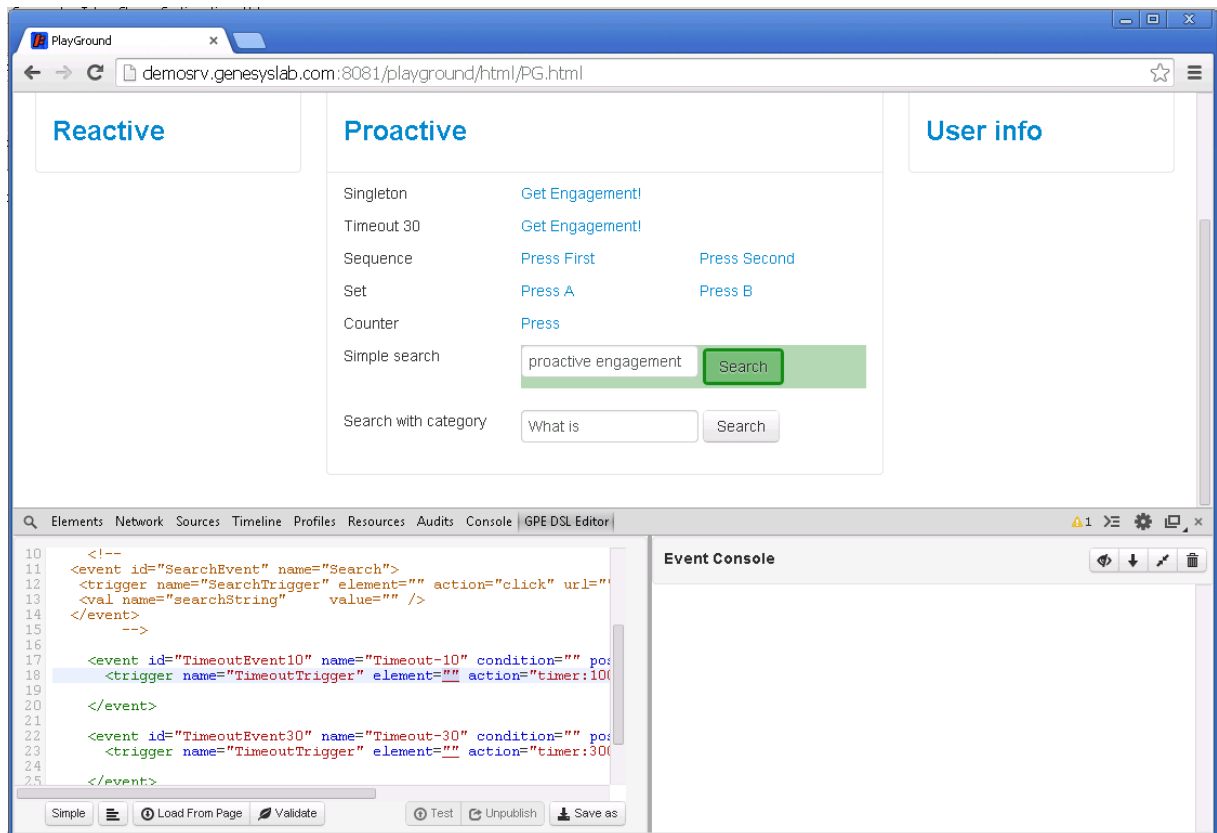
Start

1. Open the Playground application in InTools. You can launch InTools by navigating to **GWE_installation_directory\tools\intools**. Take the **intools.crx** file and drop it into the Extensions page of your Google Chrome or Chromium browser.
2. Press F12 to open the developer tools and select **GPE DSL Editor**.

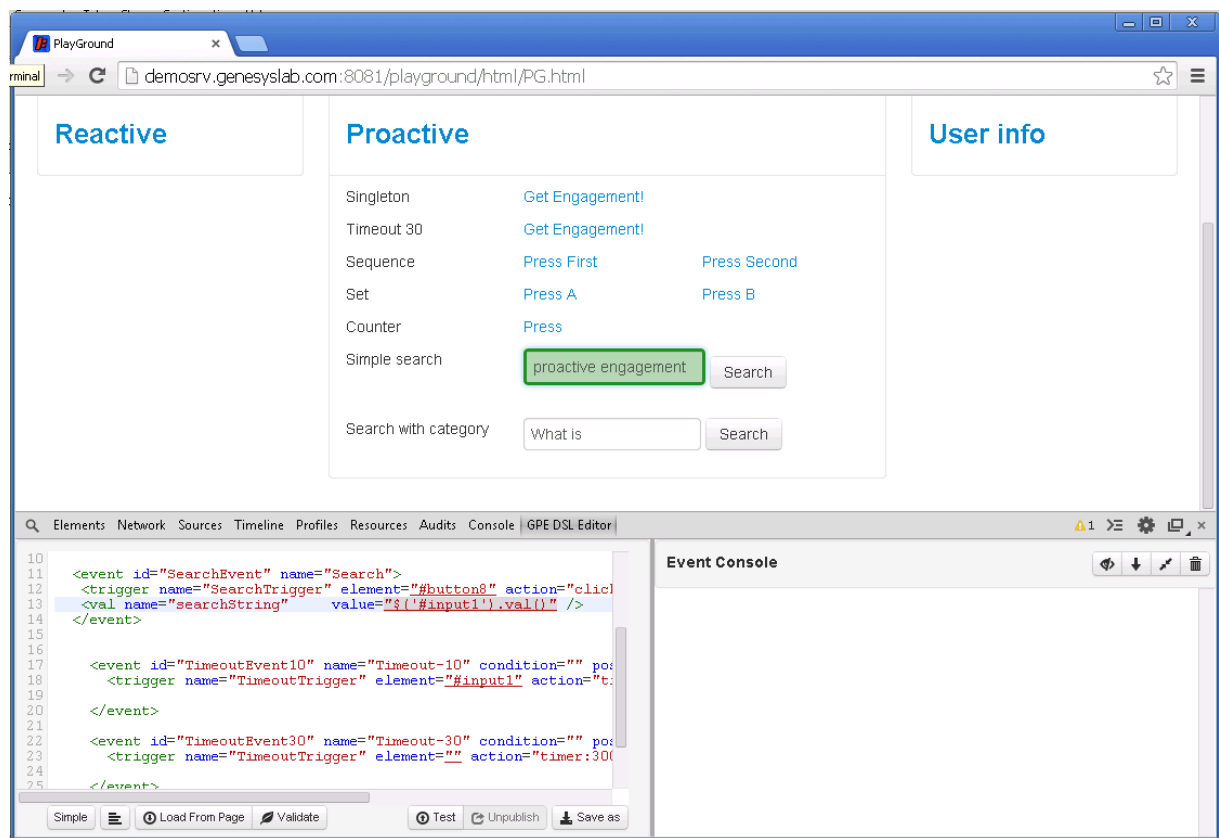


The left window shows the DSL for Playground application, while the right window shows the events on the page.

- Find the "TimeoutEvent10" **<event>** in the DSL and select the "element" attribute on its **<trigger>**. InTool goes into search mode.
- Select **Search**. The selector for **Search** replaces the "element" attribute value. We now have a trigger for the event in Step 5 of our test scenario: 10 seconds after loading the search results appears chat invitation window.



5. If you select a "value" attribute in the DSL, InTool goes into search mode. Now you can select an item on the page and the selector for that item will replace the "value" attribute in the DSL.



6. Click **Test** to confirm your event works.

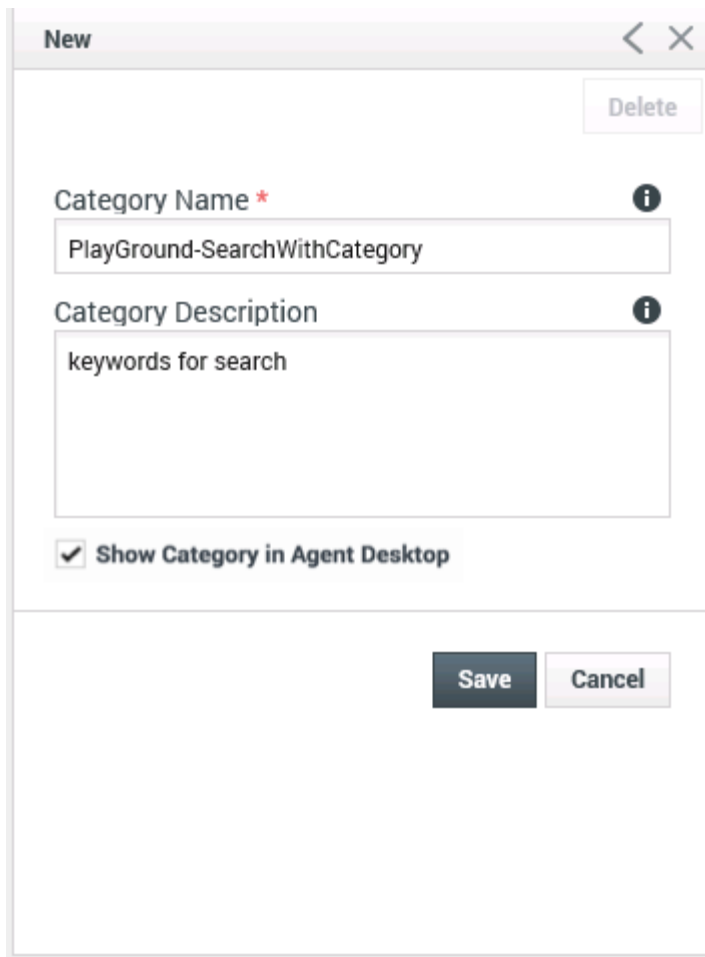
End

2. Create categories with the GAX plug-in.

In the steps below we create a category that matches the "contract" search term, Step 2 in our Test Scenario.

Start

1. In Genesys Administrator Extension, navigate to **Configuration > Web Engagement > Categories**.
2. Click **+** to create a new category.
3. Enter the following information:
 - **Category Name** — Playground-SearchWithCategory
 - **Category Description** — Keywords for search
 - Check **Show category in Interaction Workspace** to make the category available for agents.



The screenshot shows a 'New' category creation form. At the top right, there are navigation icons (back and close) and a 'Delete' button. The form contains two main input fields: 'Category Name' with a red asterisk and an information icon, containing the text 'PlayGround-SearchWithCategory'; and 'Category Description' with an information icon, containing the text 'keywords for search'. Below these fields is a checkbox labeled 'Show Category in Agent Desktop' which is checked. At the bottom right, there are 'Save' and 'Cancel' buttons.

- Click **Save**. The new category is added to the list of categories
4. Select the **Playground-SearchWithCategory** category.
 5. In the Category Matching Tags section, click +.
 6. Enter the following information:
 - **Name** — PlayGround-SearchWithCategory
 - **Type** — Plain Text
 - **Expression** — contact
 - **Language** — Select a language from the list.

New < X

Name *
PlayGround-SearchWithCategory

Type *
Plain Text ▼

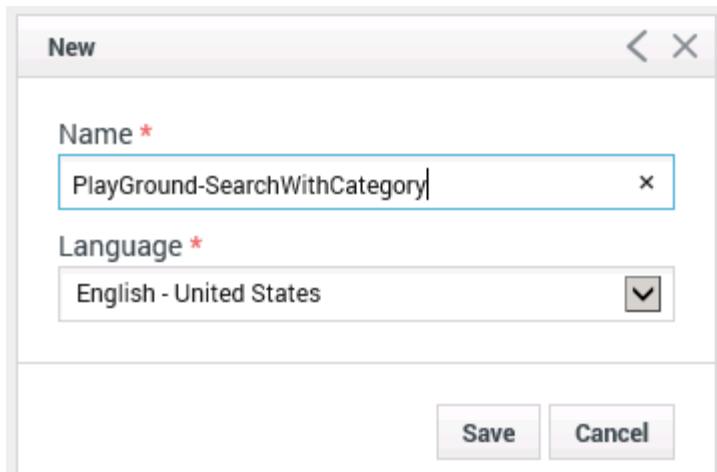
Expression *
contract x

☐ Case-sensitive

Language *
English - United States ▼

Save Cancel

- Click **Save**. The tag is added to the Category Matching Tags section.
7. In the Language section, click +.
 8. Enter the following information:
 - **Name** — PlayGround-SearchWithCategory
 - **Language** — Select a language from the list.



- Click **Save**. The language is added to the Language section.

End

3. Configure Genesys Rules System

Complete the following procedures Genesys Rules System to work with Web Engagement.

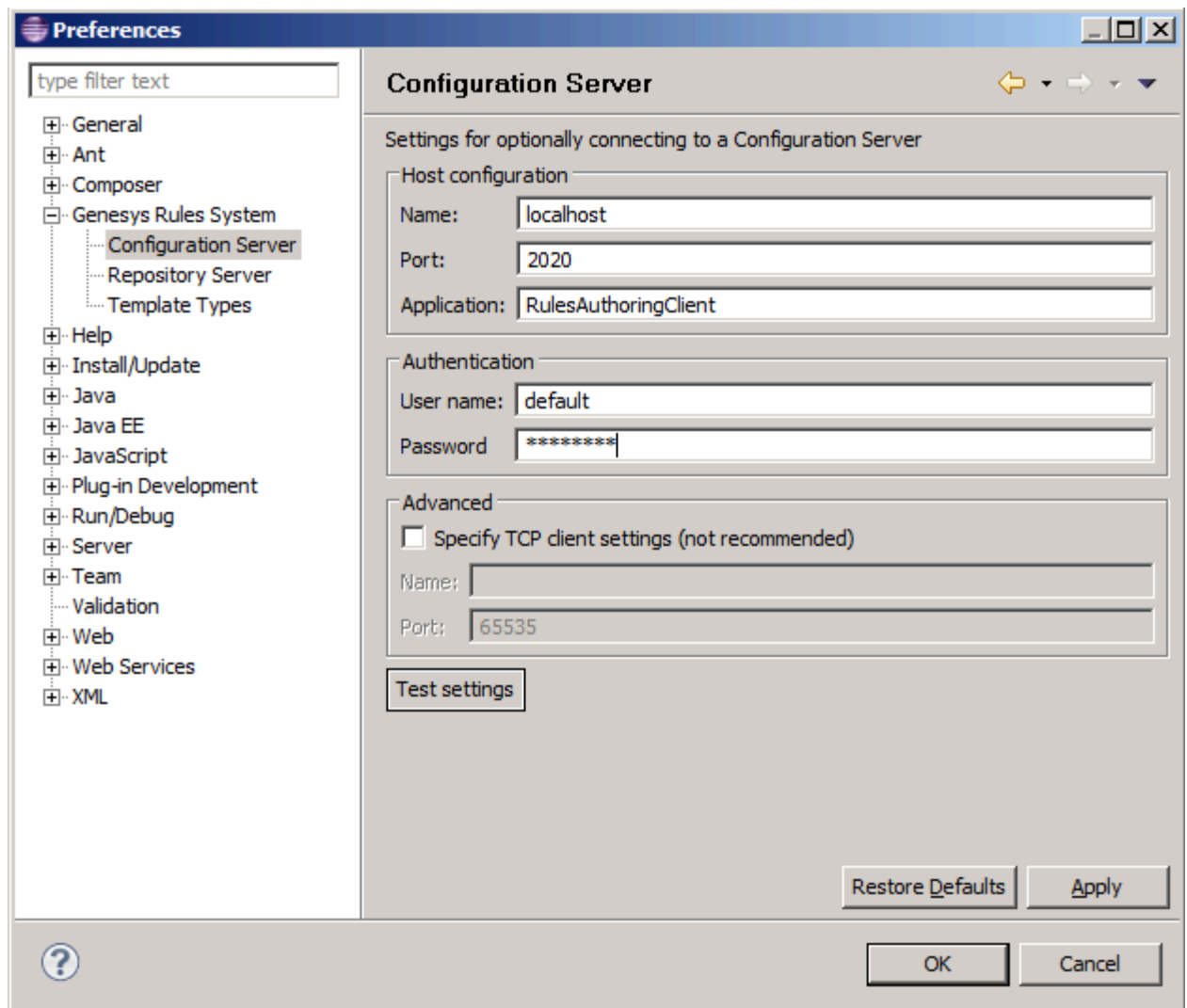
Configuring Genesys Rules Development Tool

Prerequisites

- Your environment includes Genesys Rules Development Tool (GRDT), deployed as a Composer plug-in. See [Genesys environment prerequisites](#) for compliant versions. For more information about installing GRDT, refer to [Installing the GRDT Component](#) in the the Genesys Rules System Deployment Guide.
- You enabled the Galileo update site in GRDT, as described in [Installing the GRDT Component](#).

Start

1. Open Genesys Rules Development Tool by starting Composer.
2. Navigate to **Window > Preferences**. The **Preferences** window opens.
3. Navigate to **Genesys Rules System > Configuration Server**. Edit the settings.
 - Enter the Configuration Server host name. For instance, localhost.
 - Enter the Configuration Server port. For instance, 2020.
 - Enter the application name for the Rules Authoring Client application. For instance, RulesAuthoringClient.
 - In the Authentication section, enter the name and password for a user who can connect to Configuration Server and click **Apply**.

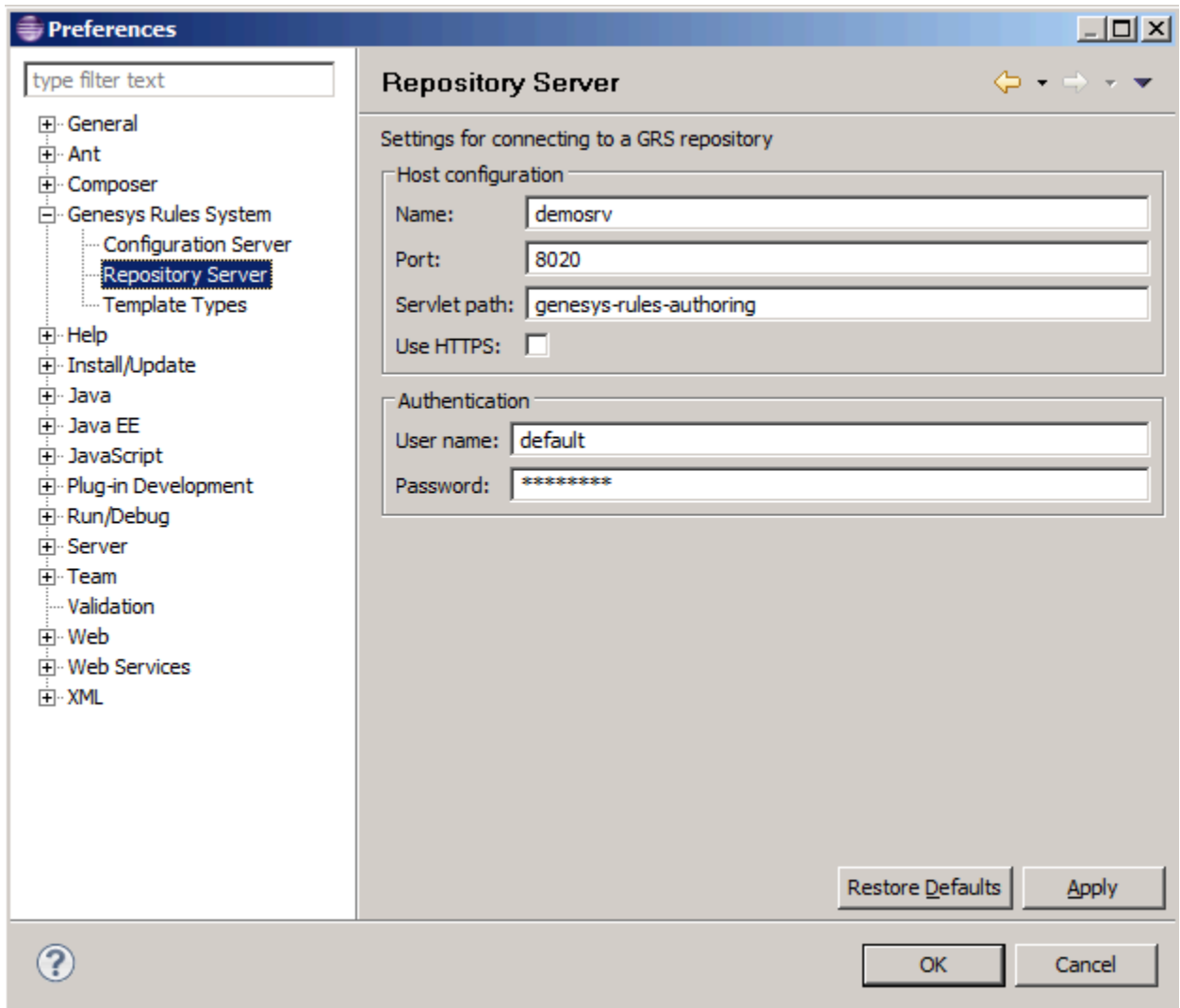


GRDT settings for Configuration Server.

4. Navigate to **Genesys Rules System > Repository Server**. Edit the settings.

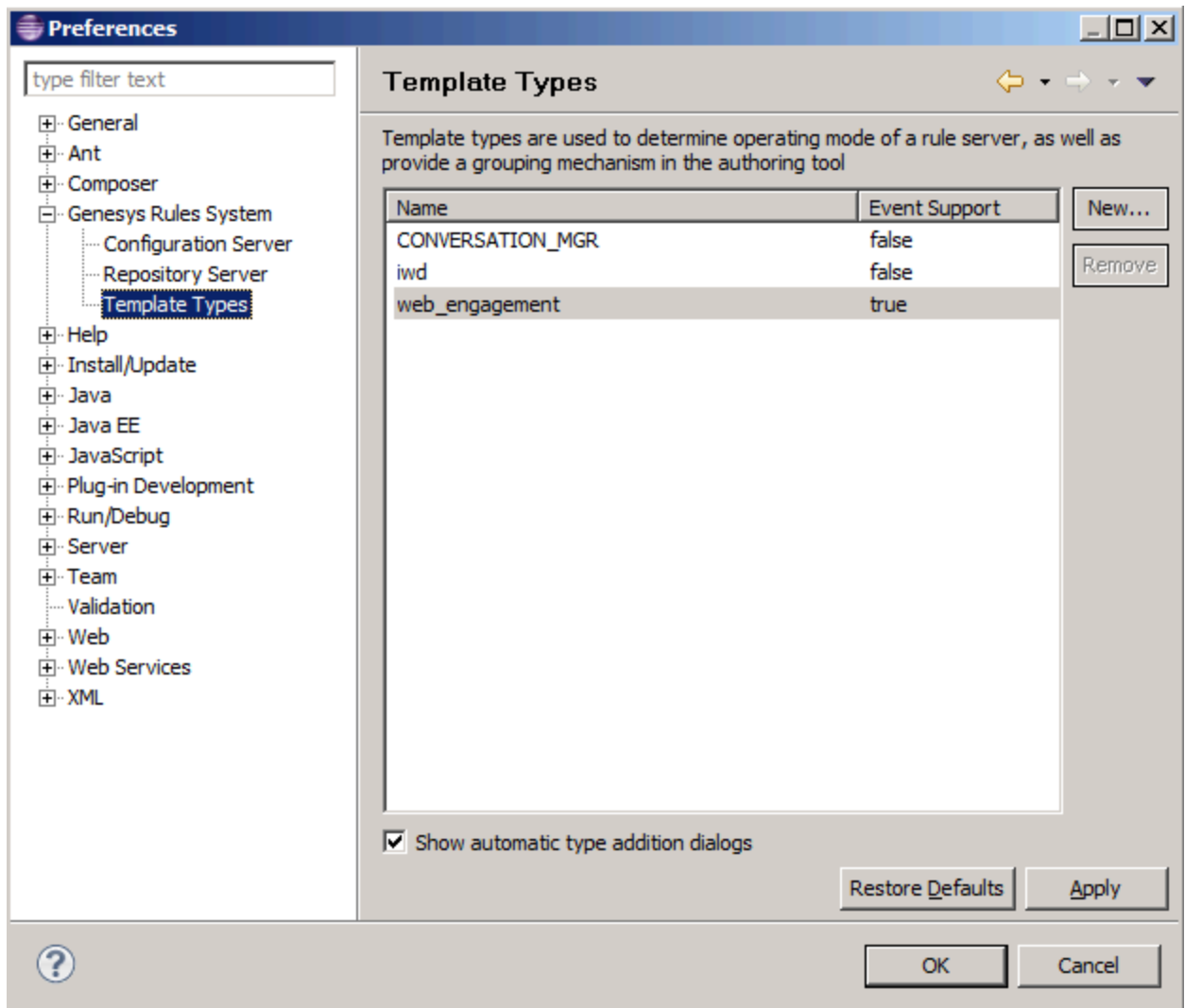
- Enter the Repository Server host name. This is the name of the host specified for the GRS application (application with type Business Rules Execution Server) in Genesys CME. For instance, localhost.
 - Enter the Repository Server port. This is the port with the ID genesys - rules - engine that is specified for the GRS application (with type Business Rules Execution Server) in Genesys CME. For instance, 8020.
 - Enter the Servlet path as genesys - rules - authoring.
 - In the Authentication section, enter a name and password for a user who:
 - Has Read and Execute permissions for the Genesys Rules Authoring client application (set up in Configuration Server); this user must have explicit Read and Execute permissions or must belong to an access group with those permissions.
 - Belongs to a Role with the following privileges: Template - Create, Template - Modify, Template - Delete.
-

5. Click **Apply**.



GRDT settings for the Repository Server.

6. Navigate to **Genesys Rules System > Template Types**. If it is not present, add the **web_engagement** template type and set **Event Support** to true. Click **Apply**.



Template types in Composer

7. Click **OK**.

End

Configuring Genesys Rules Authoring Tool

Prerequisites

- Your environment includes Genesys Rules Authoring Tool (GRAT). See [Genesys environment prerequisites](#) for compliant versions. For more information about installing GRAT, refer to the [Genesys Rules System Deployment Guide](#).

Start

- In Genesys Administrator, navigate to **Provisioning > Environment > Applications**, select the GRAT application (the application with type Business Rules Execution Server), and click **Edit**.

2. In the Connections section, click **Add...**
3. Select the Genesys Web Engagement Cluster and click **OK**.
4. Select the Web Engagement Cluster in the list of connections and click **Edit**.
5. In the **Connection Info** window, select default for the ID field. Click **OK**.

The 'Connection Info' dialog box is shown with the 'General' tab selected. The fields are as follows:

- * Server: GWE_85_Cluster
- * ID: default (9081)
- Connection Protocol: (empty)
- Local Timeout: 0
- Remote Timeout: 0
- Trace Mode: [Unknown Trace Mode]
- Connection Type: Unsecured

At the bottom, there are 'OK' and 'Cancel' buttons. A status bar at the very bottom shows 'GWE_85_Cluster' and 'Message_Server' with a value of 0.

ID is set to the default port.

6. Select the **Options** tab.
7. In the **[settings]** section, set **verify-deploy-address** to false. You must set this option because in Genesys Web Engagement, rule packages are deployed to the Web Engagement Server, not to Genesys Rules Engine. When set to false, this option prevents GRS from trying to verify that the server deploying the rules is Genesys Rules Engine.

ConfigurationOptionsPermissionsDependenciesAlarmsLogs

NewDeleteExportImport

View: Advanced View (Options)

Name	Section	Option	Value
Filter	Filter	Filter	Filter

settings (6 Items)

settings/group-by-level	settings	group-by-level	false
settings/max-connections	settings	max-connections	99
settings/session-timeout	settings	session-timeout	30
settings/session-timeout-alert-interval	settings	session-timeout-aler...	1
settings/strict-mode	settings	strict-mode	true
settings/verify-deploy-address	settings	verify-deploy-address	false

verify-deploy-address is set to false.

8. In the **Security** tab, set a user who has Read, Create, Change rights for the Scripts folder in **Log On As**. This user should also have: Read access to all tenants which are supposed to be used; Role with

sufficient permissions (as detailed in [Genesys Rules System Deployment Guide](#)); Read access to Business Structure folder and associated nodes that are supposed to be used; Read access to Scripts folder and Scripts objects (which are representations of the rule templates).

9. Click **Save & Close**.

End

(Optional) Configuring Roles Settings for Rules Management

You should complete this procedure if you need to import permissions to enable a user to create rules for Genesys Web Engagement. Once roles are imported, you can assign them to the user who publishes the rule templates and creates rules for GWE.

You should not complete this procedure if you have already created a user who has permissions to create rules packages in Genesys Rules Authoring (as described in the "Role-based Access Control" chapter of the [Genesys Rules System Deployment Guide](#)).

Start

1. In Genesys Administrator, navigate to **Provisioning > Accounts > Roles** and click **New...**
2. Enter a name for the new role. For example, GWE_Rules_Administrator.
3. In the Members section, you can specify who should have this role. Click **Add** to add as many access groups or users as you need.

GWE_Rules_Administrator - \Roles

Cancel Save & Close Save Save & New Reload Validate Permissions

Configuration Role Privileges Permissions

*** General**

* Name: GWE_Rules_Administrator

Description: in GRDT - import or create, modify and publish templates, in GRAT - create, modify, delete and

Tenant: Environment

State: ☒ Enabled

Members

Users:

User Name	Agent	Last Name	First Name
No objects to display			

Access Groups:

Name	Type
No objects to display	

New Role

4. Select the **Role Privileges** tab and select Genesys Rules Authoring Tool. The privileges for GRAT are added to the GWE_Rules_Administrator role.

GWE_Rules_Administrator - \Roles

Cancel Save & Close Save Save & New Reload Validate Permissions

Configuration **Role Privileges** Permissions

Allow All Export Import

Add/Remove Products

☐ Interaction Workspace

☒ Genesys Rule Authoring Tool

☐ Genesys Administrator

Name	Value
Filter	Filter
Rule Authoring (4 Items)	
Create Rule	
Delete Rule	
Modify Rule	
View Rule	
Rule Packages (4 Items)	
Create Package	
Delete Package	
Deploy Package	
Modify Package	
Rule Templates (3 Items)	
Create Template	
Delete Template	
Modify Template	
Business Calendars (4 Items)	
Create Calendar	
Delete Calendar	
Modify Calendar	
View Calendar	

Imported Role Privileges

5. Click **Save & Close**.

End

Members of the GWE_Rules_Administrator role can now do the following:

- Create, modify, and delete rules
- Create, modify, delete, and deploy rule packages
- Create, modify, and publish CEP rule templates

4. Prepare the CEP Rule Templates.

Complete the following procedures to import, configure, and publish the CEP Rule Templates.

Importing the CEP Rule Templates in Genesys Rules Development Tool

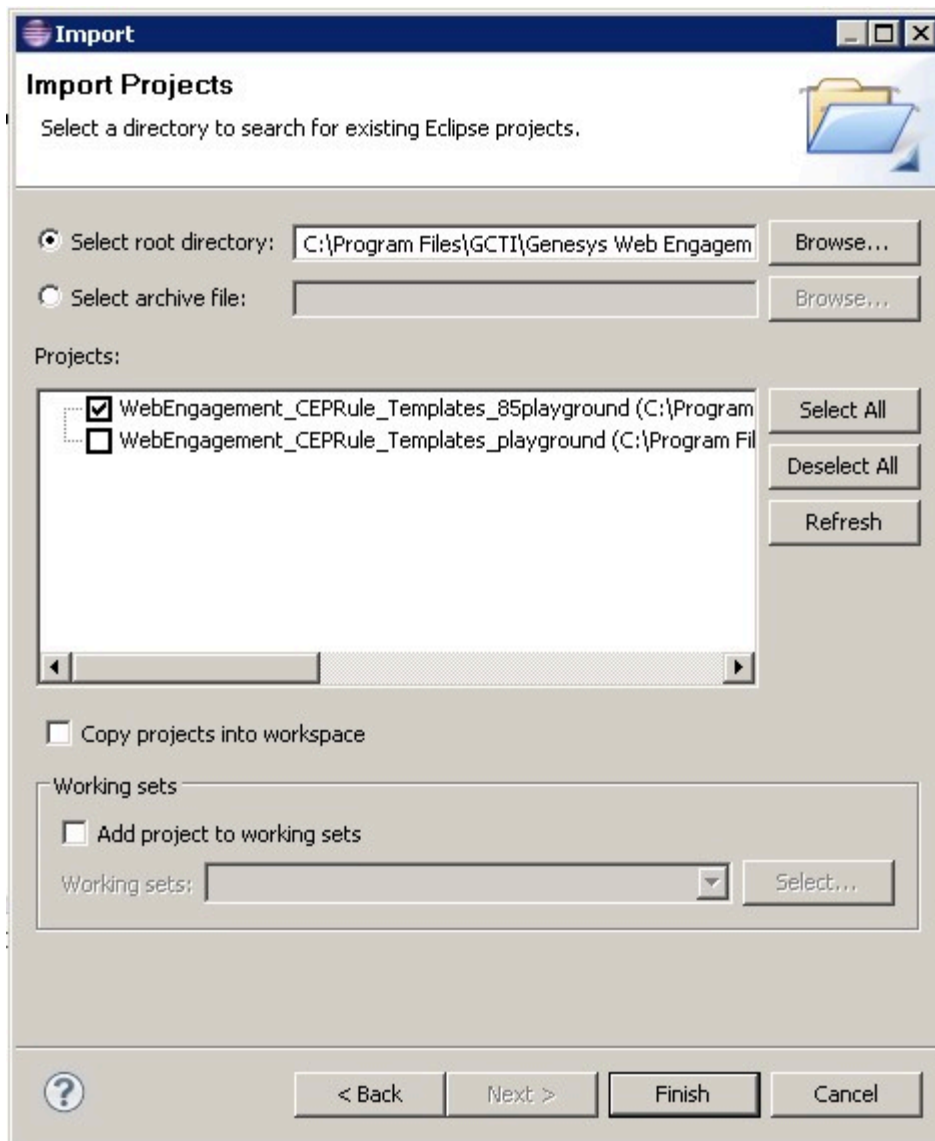
Complete this procedure to import the CEP rule templates in the Genesys Rules Development Tool. Even if you do not plan to customize the templates, your rule template must be published in the Rules System Repository before you try to create rules.

Prerequisites

- The Genesys Rules Development Tool is installed, configured, and opened in Composer.

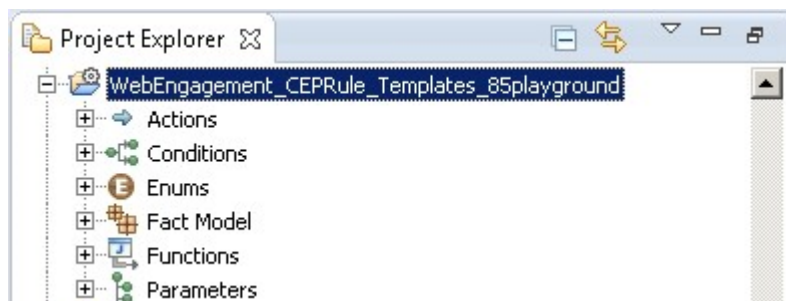
Start

1. Navigate to **Window > Open Perspective > Other > Template Development** to switch to the Template Development perspective of the Genesys Rules Development Tool.
2. Select **File > Import....**
3. In the **Import** dialog window, navigate to **General > Existing Projects into Workspace**. Click **Next**.
4. Select **Select Root Directory:**, then click **Browse**.
5. Import your project, located in the folder at **Web Engagement installation directory\apps\playground\resources_composer-projects\WebEngagement_CEPRule_Templates**. Select the rules template project to import:
 - Browse to the **\apps\playground\resources_composer-projects** folder in the Genesys Web Engagement installation directory and select a project.
 - Click **OK**. **WebEngagement_CEPRule_Templates_playground** is added to the **Projects** list.
 - Select the **WebEngagement_CEPRule_Templates_playground** project.
 - Warning: Do **not** enable the option **Copy projects into workspace**.



Import the default templates by clicking **Finish**.

- Click **Finish** to import the project. **WebEngagement_CEP_Rule_Templates_playground** is added to the **Project Explorer**.



WebEngagement_CEP_Rule_Templates_playground is added to the Project Explorer.

End

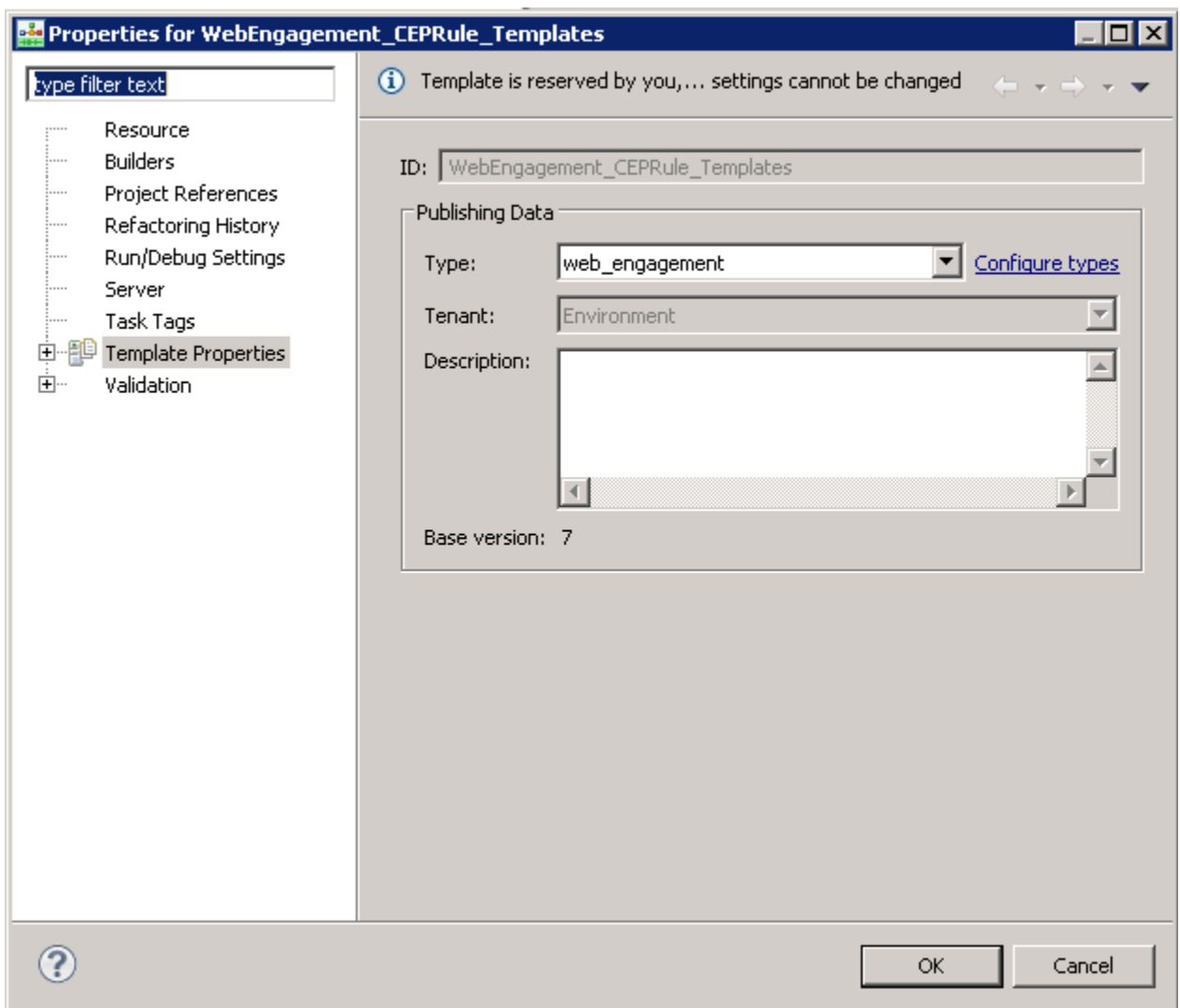
Configuring the CEP Rule Templates

Prerequisites

- The **Web Engagement Categories** business attribute is defined in Genesys Administrator.

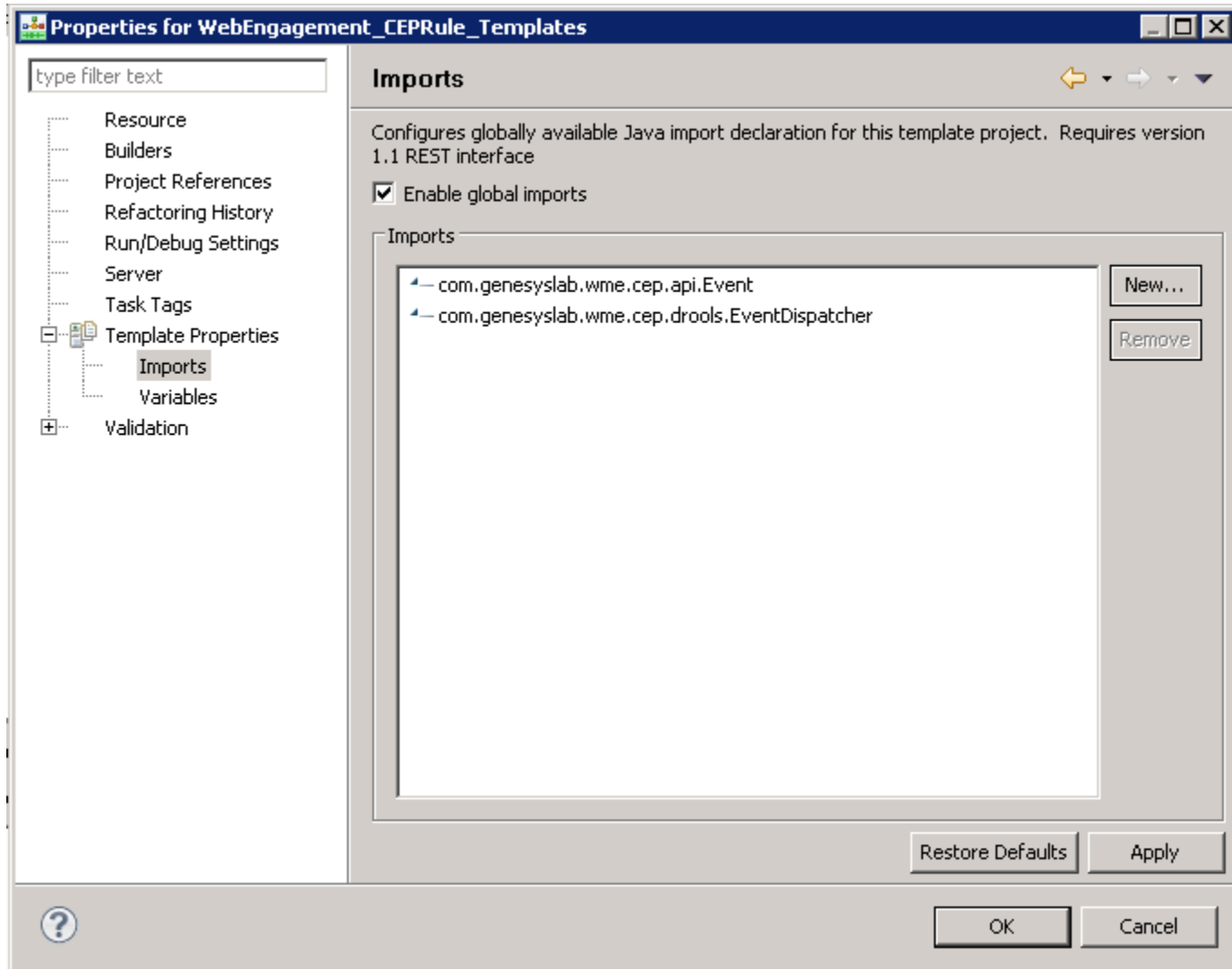
Start

1. In the GRDT **Project Explorer**, right-click on the **WebEngagement_CEPRule_Templates** project. Click **Properties**.
2. In the **Properties** dialog window, navigate to **Template Properties**. In **Publishing Data**, set **Type** to **web_engagement**.



Set the **type** to **web_engagement**.

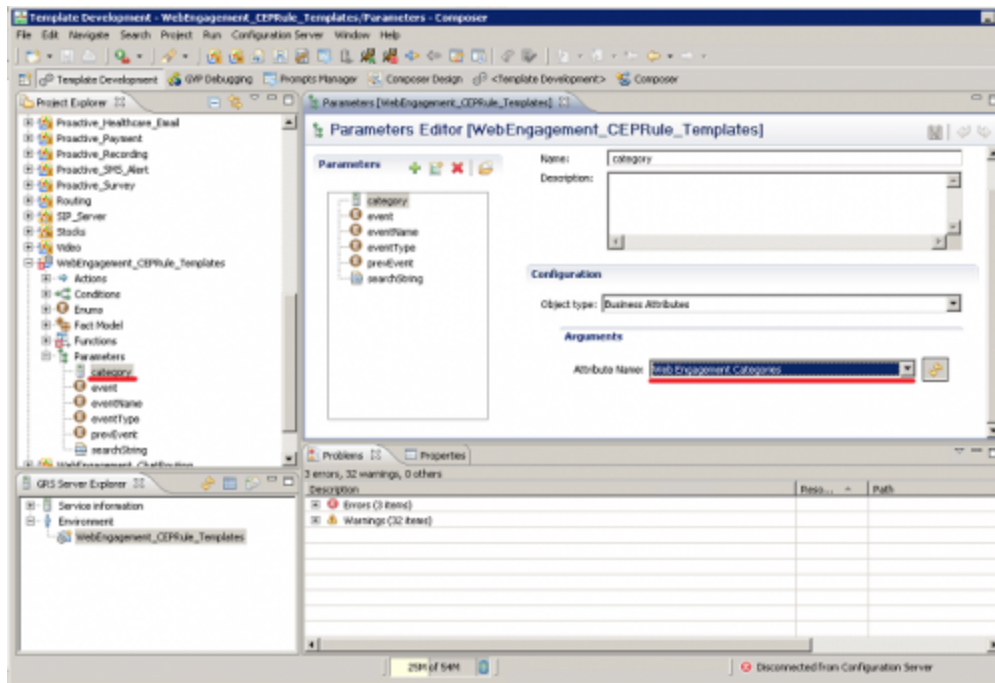
3. Navigate to **Template Properties > Imports**. The **Imports** panel opens.
4. Select the **Enable global imports** option.



Enabling global imports.

Note: The **com.genesyslab.wme.cep.api.Event** and **com.genesyslab.wme.cep.drools.EventDispatcher** packages must be present.

5. Click **OK**.
6. In the **Project Explorer**, navigate to **WebEngagement_CEPRule_Templates > Parameters > category**.
7. In the **Parameters Editor** panel, set **Attribute Name** to Web Engagement Categories.



8. Click **Save**.

End

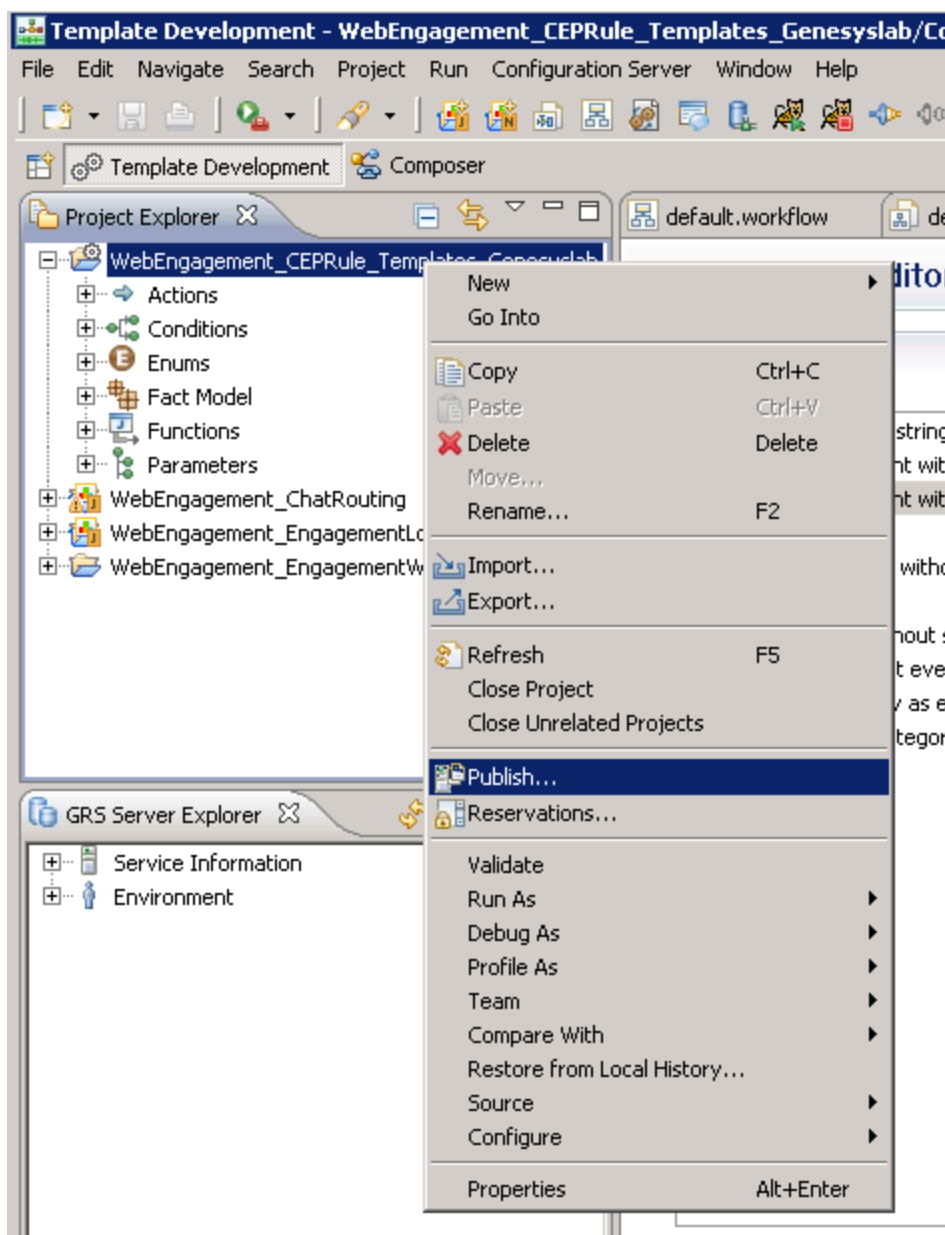
Publishing the CEP Rule Templates in the Rules Repository

Prerequisites

- Your user has the correct permissions to manage rules in GRAT, as detailed in the [Genesys Rules System Deployment Guide](#).
- You configured GRDT to enable a connection to Configuration Server and Rules Repository Server.

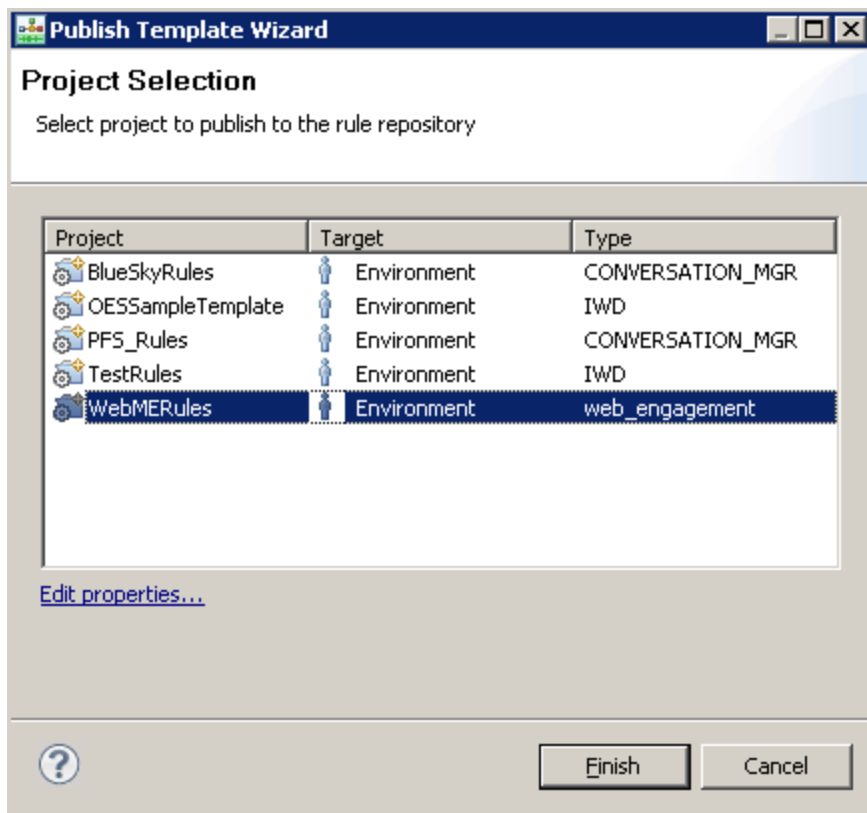
Start

1. In **Project Explorer**, right click **WebEngagement_CEPRule_Templates**.
2. Select **Publish**. The **Publish Template Wizard** opens.



The Publish Template Wizard.

3. Select **WebEngagement_CEP_Rule_Templates**.



Select **WebEngagement_CEP_Rule_Templates**.

4. Click **Finish**.

End

5. Create business rules in Genesys Rules Authoring Tool.

Complete the steps below to create business rules in GRAT to use in our Test Scenario.

Start

1. In GRAT, navigate to your site and select New Rule Package

The screenshot shows the Genesys Rules Authoring application. The left sidebar displays a tree view of environments: Environment, ACME Solution, BlueSky, Demo Solution, Playground (highlighted with a 'New Rule Package' button), Premier Financial Services, and Social. The main area is titled 'GENESYS RULES AUTHORIZING' and shows the 'General' tab for a new rule package. The fields are filled with: Package Name: com.playground, Business Name: Playground Rules, Package Type: web_engagement, and Description: Rules for the Playground application test scenario. There are 'Import Rule Package' and 'Export Rule Package' buttons. Below the description is a table of templates.

Selected	Name	Version	Version Comment	Modified By	Date Modified
<input checked="" type="checkbox"/>	WebEngagement_CEP1	1		demo	Jun 27, 2014 4:35 PM
<input type="checkbox"/>	WebEngagement_CEP1	1		demo	Nov 7, 2013 7:49 PM
<input type="checkbox"/>	WebEngagement_CEP1	4		demo	Nov 6, 2013 3:19 PM
<input type="checkbox"/>	WebEngagement_CEP1	3		demo	Oct 29, 2013 11:44 PM
<input type="checkbox"/>	WebEngagement_CEP1	2		demo	Sep 24, 2013 2:48 PM

At the bottom right are 'Save', 'Cancel', and 'Delete' buttons.

2. Enter the following information:
 - **Package Name** — com.playground
 - **Business Name** — Playground Rules
 - **Package Type** — Select web_engagement
 - **Description** — Rules for the Playground application test scenario.
 - **Template** — Select the Web Engagement CEP template from the list.
 - Click **Save**.
3. Select the package you just created and select the **Rules** tab.

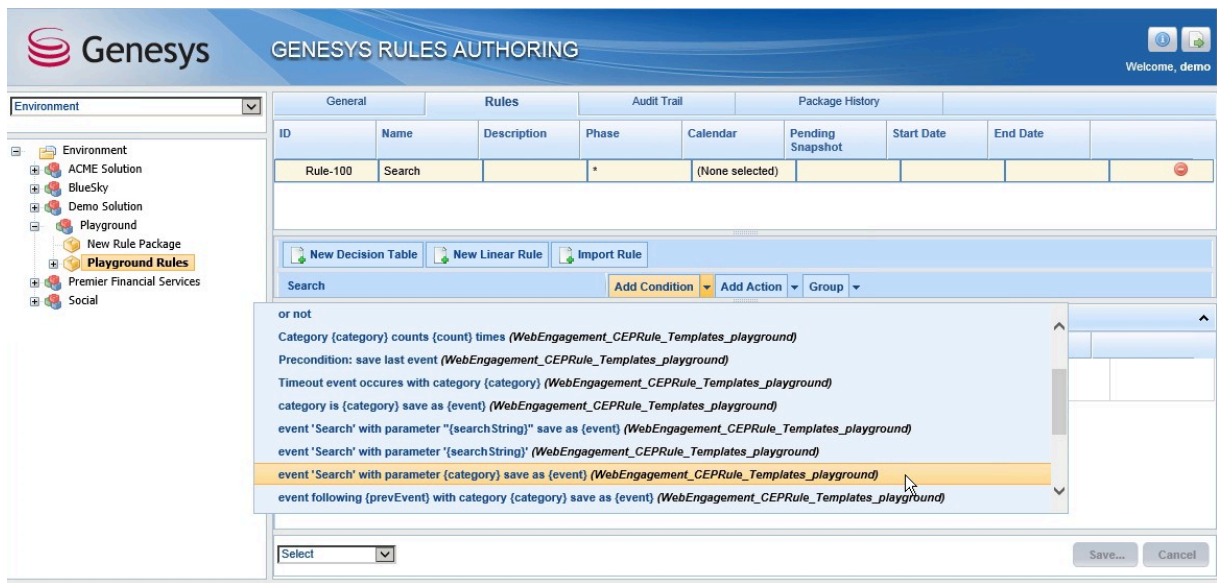


4. Click **New Linear Rule** and enter the following information:

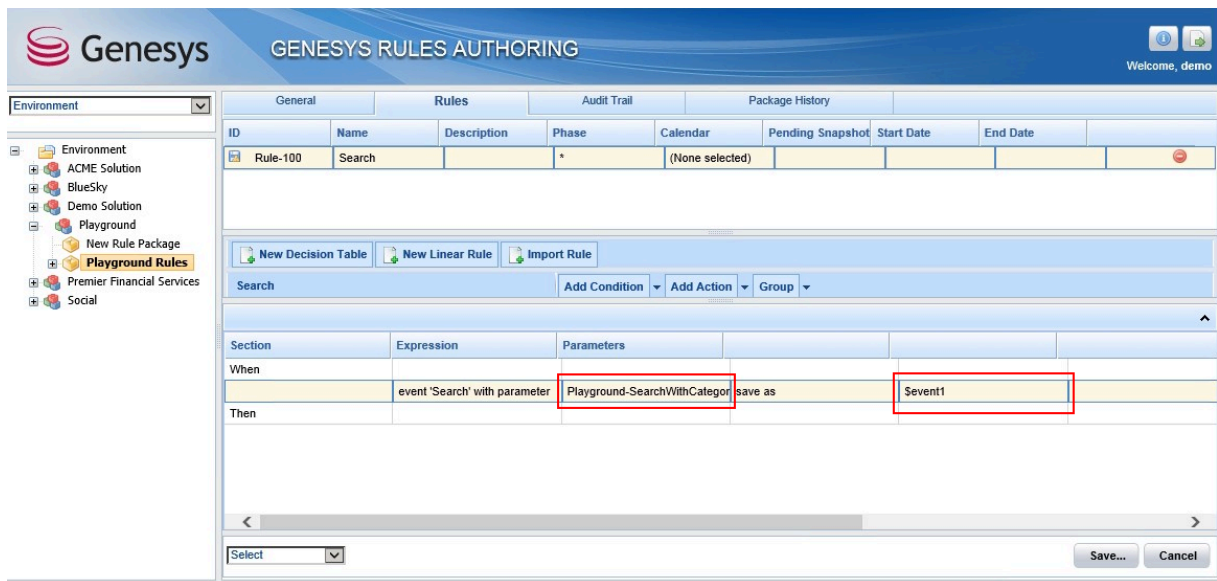
- **Name** — Search
- **Phase** — *



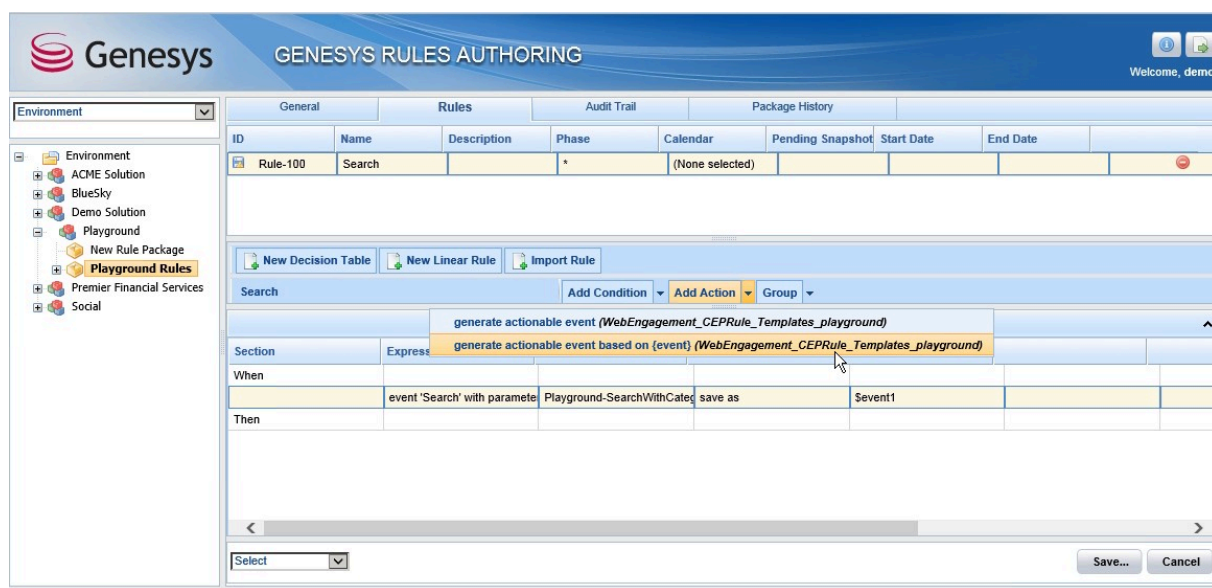
5. Click **Add Condition** and select "event 'Search' with parameter {category} save as {event}".



6. Select "PlayGround-SearchWithCategory" and "\$event1".



7. Click **Add Action** and select "generate actionable event based on {event}".



8. Select "\$event1" from the list.

9. Click **Save**. We now have a business rule for our search with category scenario.

End

6. Deploy rules to the Genesys Web Engagement Environment.

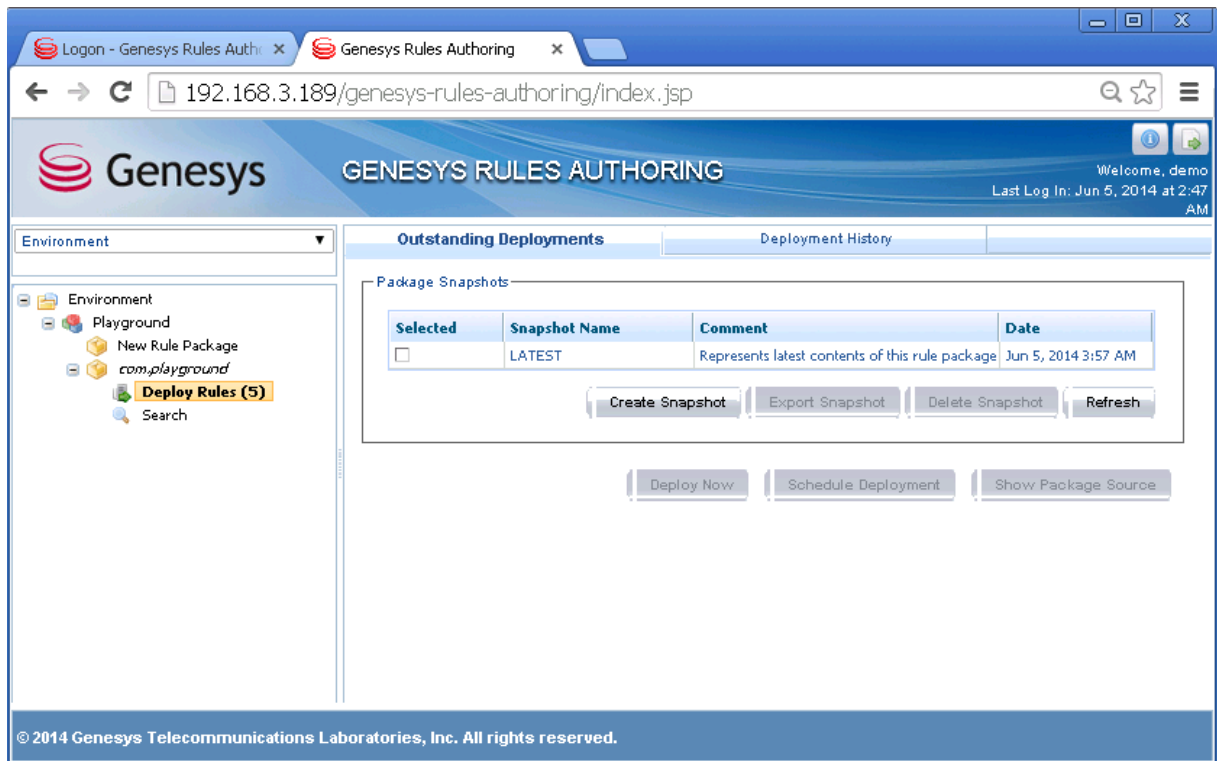
Complete the steps below to deploy your rules to the GWE environment.

Prerequisites

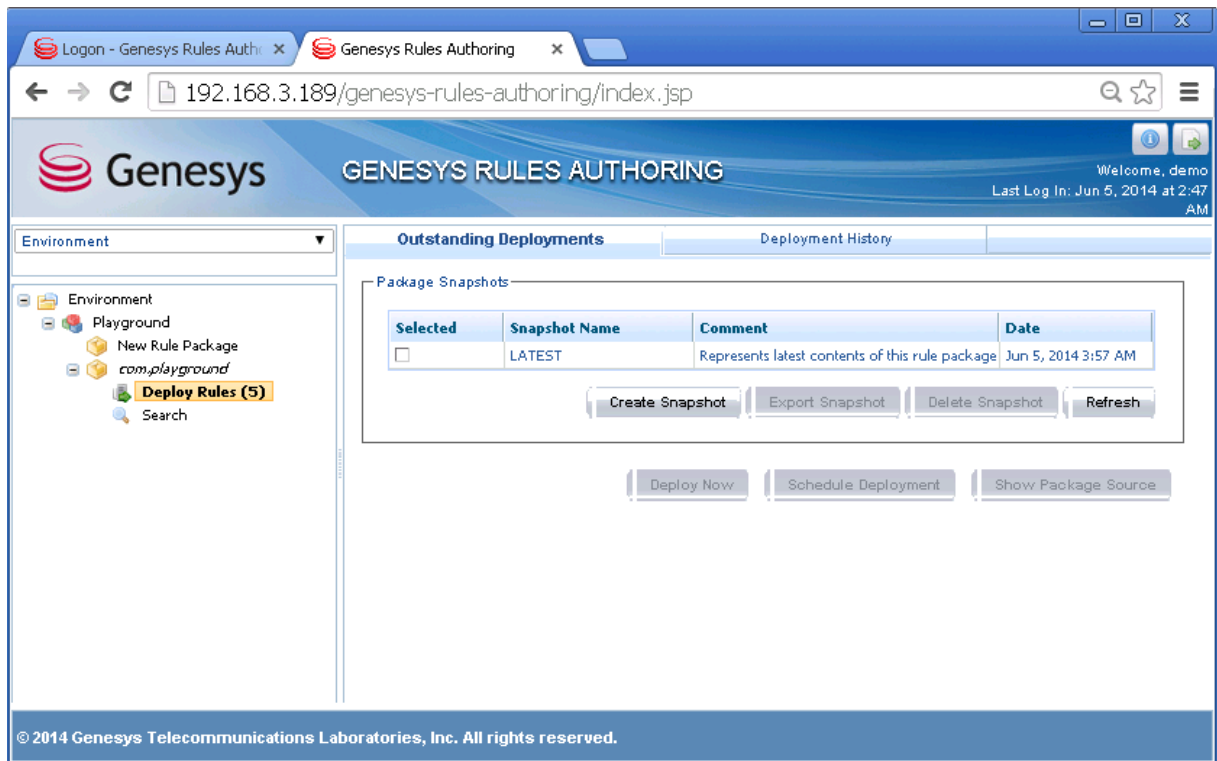
- You started the Web Engagement servers.

Start

1. In Genesys Rules Authoring Tool, navigate to **Solution > your rules package > Deploy Rules**.



2. Select the checkbox next to your rules package in the Package Snapshots section.
3. Click **Deploy Now**. The **Deploy** window opens.
4. Select your Genesys Web Engagement Server for the **Location**.



5. Click **Deploy**. The rules package is deployed to the Web Engagement system.

End

7. Deploy the Playground application.

Warning

Make sure to **stop** your GWE servers before deploying an application.

Deploying your Application

Start

1. Navigate to the installation directory for Genesys Web Engagement and open a new console window.
2. Type:

```
deploy playground
```

End

8. Make sure the components in your Genesys environment are running.

- Configuration Server
- Universal Routing Server

- [Interaction Server](#)
- [Orchestration Server](#)
- [Stat Server](#)
- [Chat Server](#)
- [Interaction Workspace](#)

9. Start the Web Engagement servers.

Start

To start your servers from Genesys Administrator:

1. Navigate to **Provisioning > Environment > Applications**.
2. Select the Web Engagement Server.
3. Click **Start applications** in the **Runtime** panel.

To start your servers using the provided **start** script (**start.bat** on Windows and **start.sh** on Linux):

1. Navigate to the Web Engagement installation directory and launch a console window.
 - For Windows, type: `start.bat`
 - For Linux, type: `start.sh`

End

The Web Engagement Server is started.

10. Confirm that the test scenario is a success by walking through the [Test Scenario Steps](#).

Other Test Scenarios

You can use the Playground application to complete the following test scenarios:

- Entrance to a specific page (Singleton)
- Inactivity timeout for the user (Singleton)
- Select an element on the page (Singleton)
- Search
- Get information about user
- Successful completion of the wizard (Sequence)
- Go to the previous page of the wizard (Sequence)
- Repeated input on the same page (Counter)
- Send custom event
- Changing the type of invitation depending on the status of the user (gold, platinum)
- Filling out the form (Sequence)
- Display advertisement (Sequence)